

Exploring the Role of Matrices in Science, Technology, and Medicine field: A Comprehensive Review

D P Singh

Amity University Uttar Pradesh Greater Noida Campus

Abstract— Matrices play a fundamental role in modern science, technology, and medicine, serving as powerful tools for modeling, analysis, and problem-solving across diverse disciplines. This comprehensive review explores the wide-ranging applications of matrices, highlighting their significance in areas such as physics, engineering, computer science, data analysis, and medical sciences. The paper examines how matrix theory supports advancements in image processing, machine learning, network analysis, quantum mechanics, and biomedical signal processing. Special emphasis is given to their role in medical imaging techniques such as MRI and CT scans, where matrices enable efficient data representation and reconstruction.

Furthermore, the review discusses recent developments in computational methods and numerical techniques that enhance the efficiency and applicability of matrix-based solutions. By synthesizing insights from various domains, this study demonstrates how matrices serve as a unifying mathematical framework that facilitates innovation and interdisciplinary research. The paper concludes by identifying current challenges and future directions, emphasizing the growing importance of matrix-based approaches in solving complex real-world problems.

Index Terms—Matrices, Linear Algebra, Scientific Computing, Engineering Applications, Machine Learning, Data Analysis, Medical Imaging, Biomedical Signal Processing, Numerical Methods, Interdisciplinary Research.

I. INTRODUCTION

A matrix is a core concept in mathematics, particularly in the field of Linear Algebra. It serves as a systematic method for arranging, representing, and processing data.

A matrix is a rectangular arrangement of numbers, symbols, or algebraic expressions organized into rows and columns. It is commonly represented in bracket form as: $A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}$

The horizontal lines are called rows and the vertical lines are called columns. Each element is represented as a_{ij} , where: i = row number, j = column number. The order of a matrix is written as: Order=Number of rows \times Number of columns [14].

Various categories of matrices, including row, column, square, diagonal, identity, symmetric, and skew-symmetric matrices, exhibit distinct structural characteristics that enable their use in different mathematical computations and practical applications. Matrix operations such as addition, subtraction, multiplication, transpose, adjoint, and inverse are widely applied in solving linear equations, data processing, computer graphics, network analysis, and scientific modeling. These matrix types and operations are summarized in Table 1 and Table 2 [14].

Types of Matrices (Table1):		
Type	Mathematical Model	Key Property
Row Matrix	$(1 \times n)$	One row
Column Matrix	$(n \times 1)$	One column
Rectangular Matrix	$(m \times n)$	(m rows and n columns)
Square Matrix	$(n \times n)$	Equal rows & columns
Zero or null Matrix	$[0]$	All elements zero
Identity Matrix	I_n	Diagonal = 1
Diagonal Matrix	d_i	Non-zero diagonal only
Scalar Matrix	(kl)	Equal diagonal
Minor of a Matrix	M_{ij}	Determinant after removing row i , column j
Cofactor Matrix	$C_{ij} = (-1)^{i+j}M_{ij}$	Signed minor
Symmetric Matrix	$(A^T = A)$	Mirror elements
Skew-Symmetric Matrix	$(A^T = -A)$	Opposite elements

Basic operation of a Matrix (Table2):			
Operation	Mathematical Model	Condition	Result
Addition/Subtraction.	$(a_{ij} \pm b_{ij})$	Same order	Same size matrix
Multiplication	$C = A \cdot B$ $= \sum_{k=1}^n a_{ik} b_{kj}$	Columns of pre matrix = Rows of post matrix	New matrix
Transpose	$A^T = [a_{ji}]$	Rows \leftrightarrow Columns	Order changes
Adjoint of a Matrix	$\text{adj}(A) = (C_{ij})^T$	Transpose of Cofactor Matrix	Cofactors directly build the adjoint
Inverse	$A^{-1} = \frac{\text{Adj}(A)}{ A }, A \neq 0$	$AA^{-1} = I$	Inverse of A

Matrices are among the most fundamental mathematical instruments driving contemporary scientific and technological progress. Emerging from the field of linear algebra, they offer an organized and concise framework for representing, analyzing, and manipulating data, thereby supporting efficient solutions to complex problems in diverse domains. In recent decades, the scope of matrix applications has grown remarkably, establishing their importance in areas such as physics, engineering, computer science, and medical sciences [15].

Within science and engineering, matrices play a crucial role in modeling physical phenomena, solving systems of linear equations, and describing transformations in multidimensional spaces. In physics, for example, matrices are central to quantum mechanics, where they are used to represent operators and quantum states [21]. Likewise, in engineering, matrices are extensively applied in structural analysis, signal processing, and control system design, allowing researchers and practitioners to study complex and large-scale dynamic systems effectively [7].

The significance of matrices has increased substantially with the advancement of computational technologies. In the domains of computer science and data analytics, matrices form the foundation of numerous algorithms used in machine learning, artificial intelligence, and network analysis. Methods such as matrix factorization, eigenvalue decomposition, and singular value decomposition

(SVD) are extensively utilized in applications including recommendation systems, image processing, and pattern recognition [12]. Moreover, matrices provide efficient mechanisms for storing, organizing, and processing large-scale datasets, making them indispensable in the age of big data. In the healthcare and medical sectors, matrix-oriented techniques have significantly transformed diagnostic and analytical procedures. Imaging technologies such as Magnetic Resonance Imaging (MRI) and Computed Tomography (CT) depend heavily on matrix representations for reconstructing images from raw signals and data [9]. Furthermore, matrices play a vital role in biomedical signal processing, particularly in the analysis of electrocardiogram (ECG) and electroencephalogram (EEG) data, thereby supporting precise diagnosis and continuous disease monitoring. These diverse applications demonstrate the essential contribution of matrices to advancements in healthcare and medical research.

Recent advances in numerical methods and high-performance computing have significantly expanded the use of matrices in addressing complex real-world problems. Modern computational approaches enable quicker data processing and improved accuracy, thereby increasing the importance of matrix-based techniques in interdisciplinary research [1]. As scientific and technological problems grow more sophisticated, matrices continue to serve as a fundamental framework connecting theoretical principles with practical applications.

This paper seeks to examine the wide-ranging applications of matrices in the fields of science, technology, and medicine. By integrating insights from multiple disciplines, it demonstrates the role of matrix theory in fostering innovation and supporting effective problem-solving. The review further discusses existing challenges and emerging developments, highlighting the increasing relevance of matrix-oriented methodologies in solving contemporary scientific and technological issues.

II. LITRATURE REVIEW

Matrices have long served as a core mathematical framework supporting numerous applications in science, technology, and medicine. Their capability to represent complex relationships in an organized and computationally efficient form has made them

essential in both theoretical investigations and practical implementations. Over time, research has consistently demonstrated the broad applicability of matrix theory in addressing real-world challenges across multiple disciplines.

In mathematics and physics, matrices constitute the foundation of linear algebra, which is widely used for modeling physical phenomena and solving systems of equations. As discussed in the book *Linear Algebra and Its Applications*, matrix techniques provide powerful tools for dealing with transformations, eigenvalue analysis, and dynamic systems. These principles are especially significant in quantum mechanics, where matrices are employed to represent quantum operators and states [21]. Likewise, engineering disciplines extensively utilize matrix methods in areas such as structural analysis, control engineering, and signal processing [7].

Within computer science and modern technology, matrices are indispensable for data organization, machine learning, and artificial intelligence. Contemporary machine learning models depend heavily on matrix operations to process large datasets and optimize computational performance. Goodfellow et al. [12] emphasize that neural networks fundamentally operate through matrix multiplications and transformations during both training and prediction phases. Furthermore, matrices are integral to digital image processing, where images are represented as arrays of pixel values and transformed using linear algebraic techniques [20]. Applications including computer vision, pattern recognition, and network analytics further highlight the growing significance of matrix-based approaches in advancing modern technology.

Matrices play a crucial role in statistics and data analysis. Advanced techniques such as Principal Component Analysis (PCA), Singular Value Decomposition (SVD), and regression modeling are fundamentally based on matrix operations, allowing researchers to derive valuable information from large and complex datasets [13]. These approaches are particularly important in big data analytics, where they support dimensionality reduction and the discovery of hidden patterns. In addition, matrices are extensively applied in graph theory and network analysis, where

adjacency matrices are used to describe connections among nodes within complex networks [22].

In the healthcare and biomedical domains, matrix applications have significantly enhanced modern diagnostic and analytical systems. Imaging technologies such as Magnetic Resonance Imaging (MRI) and Computed Tomography (CT) employ matrix-driven algorithms for efficient image reconstruction and processing. Kak and Slaney [2] explained that tomographic reconstruction methods use matrices to transform raw projection information into precise cross-sectional medical images. Likewise, biomedical signal processing techniques, including Electrocardiogram (ECG) and Electroencephalogram (EEG) analysis, depend on matrix-based methods for noise removal, filtering, and feature extraction [5].

Recent progress in computational techniques has considerably improved the effectiveness and usability of matrix-oriented methods. Advanced numerical approaches, including iterative algorithms, sparse matrix computations, and parallel processing, have made it possible to solve large-scale problems that were once beyond practical computational limits [4]. These innovations have played a vital role in domains such as scientific computing, artificial intelligence, and medical diagnostics, where the analysis of high-dimensional datasets is crucial.

Although matrix applications are extensively utilized across various disciplines, several challenges still persist in optimizing computations for highly complex systems. Factors such as high computational expense, numerical instability, and scalability limitations remain important areas of ongoing investigation. However, continuous advancements in sophisticated algorithms and high-performance computing infrastructures are expected to overcome these difficulties and broaden the range of matrix-based applications.

In summary, existing literature highlights matrices as a fundamental mathematical framework that connects multiple disciplines. Their capability to represent, analyze, and resolve complex problems makes them essential to contemporary research and technological innovation. With the continued growth of interdisciplinary research, the significance of matrices is likely to increase further, contributing to future developments in science, technology, and healthcare.

III. METHODOLOGY

Matrix models play a significant role in modern science, engineering, technology, and medical research because they provide a compact and efficient way to represent complex systems and large datasets. In science, matrices are widely used in physics, chemistry, and environmental studies for solving systems of equations, analyzing experimental data, and modeling dynamic processes. In engineering and technology, matrix methods are essential in structural analysis, electrical circuit design, computer graphics, and artificial intelligence, robotics, and communication systems. They help engineers perform simulations, optimize designs, and process signals efficiently. In the medical field, matrix models are increasingly applied in medical imaging, bioinformatics, disease prediction, and healthcare analytics. Techniques such as MRI and CT scan image reconstruction rely heavily on matrix computations. In this paper, we have discussed various matrix models that are widely used in the fields of Science, Engineering & Technology, and Medicine. Furthermore, we have explained the applications of some of the models and highlighted their significance in solving practical problems across these disciplines

3.1. Hamiltonian Matrix Model (Quantum Mechanics):

The Hamiltonian matrix model is one of the most fundamental mathematical models in quantum mechanics. It describes the total energy (kinetic + potential) of a system and determines how the system evolves over time [15].

Mathematical Model: The Hamiltonian operator is represented in matrix form as: $H\psi = E\psi$

Where: H = Hamiltonian matrix (energy operator), ψ = State vector (wave function), E = Energy eigenvalue.

Matrix Form Representation: For a discrete quantum system, Hamiltonian is written as:

$$H = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix}$$

Diagonal elements → energy of states. Off-diagonal elements → interaction between states.

3.2 Stoichiometric Matrix (Chemical Reaction Networks):

A Stoichiometric Matrix is a mathematical representation that describes the participation of chemical species in chemical reactions. In this matrix,

rows represent chemical species, columns represent reactions, and the entries correspond to stoichiometric coefficients, where negative values indicate reactants and positive values indicate products [23].

Mathematical Model for a system with: m species, r reactions. The stoichiometric matrix is: $S = [s_{ij}]_{m \times r}$, where: $s_{ij} < 0 \rightarrow$ species consumed, $s_{ij} > 0 \rightarrow$ species produced. The dynamic model: $\frac{dx}{dt} = S \cdot v(X)$, where: X = concentration vector, $v(X)$ = reaction rate vector.

3.3. Adjacency Matrix:

An adjacency matrix represents the connections between nodes in a network [3]. In this matrix, both rows and columns correspond to nodes such as species, organisms, or network points. The matrix entry is assigned a value of 1 if a connection exists between two nodes, and 0 if no connection exists.

Mathematical Model: For a network with n nodes:

$$A = [a_{ij}]_{n \times n}$$

where: $a_{ij} = \begin{cases} 1, & \text{If node } i \text{ is connected to } j \\ 0, & \text{otherwise} \end{cases}$

3.4 Transformation Matrix (Concept):

A Transformation Matrix is a mathematical tool used to modify the position, size, and orientation of an object in space. It is extensively applied in computer graphics, including animation, gaming, and image processing, as well as in robotics for robotic arm movement and coordinate transformations [11].

Mathematical Model: A) Rotation Matrix (2D): $\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$

Where: (x,y) = original point, (x',y') = rotated point, θ = angle of rotation.

B) Scaling Matrix (2D): $\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$

Where: S_x = scaling factor in x-direction, S_y = scaling factor in y-direction.

3.5 Jacobian Matrix:

The Jacobian matrix is a fundamental concept in multivariable calculus that describes how a system of functions changes with respect to its variables. It is extensively used in robotics, kinematics, and optimization for analyzing motion, coordinate transformations, and system behavior [19].

Mathematical Model of Jacobian Matrix: Suppose we have a vector function: $F(x_1, x_2, x_3 \dots \dots x_n) =$

$$\begin{bmatrix} f_1(x_1, x_2, x_3 \dots \dots x_n) \\ f_2(x_1, x_2, x_3 \dots \dots x_n) \\ \vdots \\ f_m(x_1, x_2, x_3 \dots \dots x_n) \end{bmatrix}$$

Then the Jacobian Matrix (J) is defined as:

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} & \dots & \frac{\partial f_2}{\partial x_n} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} & \dots & \frac{\partial f_3}{\partial x_n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \frac{\partial f_m}{\partial x_3} & \dots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

It contains all first-order partial derivatives of the system. In robotics, the Jacobian relates: Joint velocities → End-effector velocity

$$v = J(q) \dot{q}$$

Where: v = velocity of robot end-effector, J(q) = Jacobian matrix, \dot{q} = joint velocities.

3.6 Incidence Matrix Model (Network Representation):

An Incidence Matrix is a mathematical representation of a network (graph) that describes the relationship between nodes (vertices) and edges (branches). It is widely used in electrical circuit analysis, network theory, and graph theory [17].

Mathematical Model of Incidence Matrix

Let: n = number of nodes, m = number of edges.

Then the Incidence Matrix A is an n×m matrix: $A = [a_{ij}]_{n \times m}$

3.7 Elements of the Matrix:

Each element a_{ij} is defined as: $a_{ij} =$

$$\begin{cases} +1 & \text{If edge } j \text{ leaves node } i \\ -1 & \text{If edge } j \text{ enters node } i \\ 0 & \text{If edge } j \text{ is not connected to node } i \end{cases}$$

This representation is known as the oriented incidence matrix, which is widely used in electrical circuit analysis and graph theory [15].

3.8 Mass Matrix Model (Mechanical Systems):

The Mass Matrix Model is used in mechanical and structural systems to describe the distribution of mass in systems having multiple degrees of freedom. It establishes the relationship between applied forces and resulting accelerations. The model is widely applied in

vibration analysis, the Finite Element Method (FEM), robotics, and multibody dynamic systems. In simple terms, the mass matrix represents how the mass of a system resists motion in different directions [18].

Mathematical Model: For a dynamic mechanical system: $[M]\{\ddot{x}\} + [C]\{\dot{x}\} + [K]\{x\} = \{F\}$ Where: [M] = Mass Matrix, [C] = Damping Matrix, [K] = Stiffness Matrix,

$\{x\}$ = Displacement vector, $\{\dot{x}\}$ = Velocity vector, $\{\ddot{x}\}$ = Acceleration vector, $\{F\}$ = External force vector.

3.9 Confusion Matrix: A Confusion Matrix is a matrix-based model used to evaluate the performance of a classification system (in machine learning or medical diagnosis). It compares actual outcomes with predicted outcomes [26,27,28,29,30].

Mathematical Structure of Confusion Matrix:

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

Important Performance Metrics (Mathematical Model):

From the confusion matrix, we define:

$$\text{Accuracy: } Accuracy = \frac{\{TP + TN\}}{\{TP + TN + FP + FN\}}$$

$$\text{Precision: } Precision = \frac{\{TP\}}{\{TP + FP\}}$$

$$\text{Recall (Sensitivity): } Recall = \frac{\{TP\}}{\{TP + FN\}}$$

$$\text{Specificity: } Specificity = \frac{\{TN\}}{\{TN + FP\}}$$

$$F1 \text{ Score: } F1 = \frac{2 \cdot (Precision \cdot Recall)}{Precision + Recall}$$

3.10 Transition Matrix (Epidemiology Model):

A transition matrix describes how a population shifts between different disease states over time. Common epidemiological states include Susceptible (S), Infected (I), and Recovered (R). This matrix-based approach is widely used in epidemiology for modeling and analyzing the spread of infectious diseases [10].

Mathematical Model: Let the state vector be: $X(t) =$

$$\begin{bmatrix} S(t) \\ I(t) \\ R(t) \end{bmatrix}$$

$$\text{A transition matrix P is: } P = \begin{bmatrix} p_{SS} & p_{SI} & p_{SR} \\ p_{IS} & p_{II} & p_{IR} \\ p_{RS} & p_{RI} & p_{RR} \end{bmatrix}$$

Where: p_{ij} = probability of moving from state $i \rightarrow j$
 The model evolves as: $(t + 1) = P \cdot X(t)$.

3.11. Dose–Response Matrix (Pharmacology Model):

A dose–response matrix examines the relationship between varying doses of a drug and the corresponding biological responses produced in an organism [8]. It is widely used in drug testing, toxicology, and clinical trials to evaluate the effectiveness and safety of pharmaceutical compounds.

Mathematical Model: Let: D = dose levels, R = response levels

A general nonlinear model: $R(D) = \frac{E_{max} \cdot D}{EC_{50} + D}$

Where: E_{max} = maximum effect, EC_{50} = dose giving 50% response

Matrix representation: $M = \begin{bmatrix} R_1(D_1) & R_2(D_1) & \dots \\ R_1(D_2) & R_2(D_2) & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}$

3.12 Image Matrix (Pixel Matrix):

An Image Matrix is a mathematical representation of an image in which each matrix element corresponds to the intensity value of a pixel. A grayscale image is typically represented as a two-dimensional (2D) matrix, whereas a color image is represented as a three-dimensional (3D) matrix consisting of Red (R), Green (G), and Blue (B) channels [20].

Mathematical Model: A grayscale image can be represented as:

$$I = \begin{bmatrix} i_{11} & i_{12} & i_{13} & \dots & i_{1n} \\ i_{21} & i_{22} & i_{23} & \dots & i_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ i_{m1} & i_{m2} & i_{m3} & \dots & i_{mn} \end{bmatrix}$$

Where: i_{ij} = intensity of pixel at row i , column j . Values range from 0 (black) to 255 (white).

3.13 Genetic Expression Matrix:

A Genetic Expression Matrix is a mathematical representation used to describe the expression levels of genes across multiple biological samples or experimental conditions. In this matrix, rows correspond to genes, while columns represent samples such as patients, tissues, or experimental conditions. Each matrix element indicates the expression level of a particular gene in a specific sample [23].

Mathematical Model: $G = \begin{bmatrix} g_{11} & g_{12} & g_{13} & \dots & g_{1n} \\ g_{21} & g_{22} & g_{23} & \dots & g_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_{m1} & g_{m2} & g_{m3} & \dots & g_{mn} \end{bmatrix}$; g_{ij} = expression level of gene i in sample j .

3.14 Kernel Matrix (Gram Matrix):

The Kernel Matrix, also known as the Gram Matrix, is widely used in machine learning and pattern recognition to transform data into a higher-dimensional feature space where complex patterns become linearly separable using a hyperplane. Instead of explicitly computing coordinates in the transformed space, kernel methods use a kernel function to measure similarity between data points [6].

The kernel function is defined as: $K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$

Mathematical Model: The Kernel Matrix K is: $K_{ij} = K(x_i, x_j)$

Linear Kernel: $K(x_i, x_j) = x_i^T \cdot x_j$

Polynomial Kernel: $K(x_i, x_j) = (x_i^T x_j + c)^d$

3.15 RBF (Gaussian) Kernel:

The Radial Basis Function (RBF) or Gaussian Kernel is one of the most commonly used kernels in machine learning because it can efficiently model nonlinear relationships between data points [12].

$K(x_i, x_j) = \exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})$, where σ controls the spread of the Gaussian function.

3.16 Weight Matrix (Neural Networks):

In deep learning, a weight matrix represents the connection strengths between neurons across different layers of a neural network. Each input feature is multiplied by corresponding weights, summed together, and then processed through an activation function to generate the output [25].

Mathematical Model: For a single layer neural network: $y = f(Wx + b)$

Where: x = input vector, W = weight matrix, b = bias vector, $f(\cdot)$ = activation function (ReLU, sigmoid, etc.), y = output. Expanded Form: $y_i = f(\sum_{j=1}^n w_{ij}x_j + b_i)$

3.17 Covariance Matrix:

A covariance matrix describes the relationship between variables by measuring how they vary together. A positive covariance indicates that variables increase together, whereas a negative covariance indicates that one variable increase while the other decreases. A covariance value of zero suggests no linear relationship between the variables [6].

Mathematical Model: For dataset X with mean μ : $\Sigma = \frac{1}{n-1}(X - \mu)^T(X - \mu)$

Element-wise Form: $Cov(X_i, X_j) = \sum_{k=1}^n (x_{ki} - \mu_i)(x_{kj} - \mu_j)$.

3.18 Feature Matrix:

A Feature Matrix, also known as a Data Matrix, is a fundamental structure in machine learning and data analysis in which rows represent observations or samples, while columns correspond to features or variables used for model training and analysis [6].

It is usually denoted as: $X = [X_{ij}]_{m \times n}$

where: $i=1, 2, m \rightarrow$ number of samples, $j=1, 2, n \rightarrow$ number of features.

3.19 Graph Laplacian Matrix

The Graph Laplacian Matrix represents the structural properties of a graph and is extensively used in spectral clustering, network connectivity analysis, image segmentation, and graph-based machine learning models. It is derived using the adjacency matrix and degree matrix of a graph [15].

Mathematical Model: Let: A = Adjacency Matrix, D = Degree Matrix

Then Laplacian: $L=D-A$, Where: D_{ii} =degree of node I, $A_{ij}=1$ if nodes connected, else 0.

IV. MATRIX MODELING APPROACHES FOR COMPLEX REAL-WORLD SYSTEMS

Matrix modeling approaches are essential tools for solving complex real-world problems in science, engineering, medicine, economics, and technology. Matrices provide an efficient mathematical framework for representing multidimensional data, interconnected systems, and dynamic relationships. Through operations such as transformation, decomposition, inversion, and eigenvalue analysis, matrix models enable researchers to analyze system behavior, optimize processes, and improve prediction

accuracy. These methods are widely applied in machine learning, image processing, quantum mechanics, transportation systems, financial forecasting, and medical analysis. With the rise of artificial intelligence and big data analytics, matrix-based techniques have become increasingly important for handling large-scale and high-dimensional datasets. This paper presents several real-world applications of matrix modeling in science, engineering, technology, and medical fields.

4.1 Application of Hamiltonian Matrix: Used to determine electron energy states in quantum systems.

The Physical System: Two adjacent hydrogen atoms (labeled A and B), sharing one electron.

We use two basis states: $|1\rangle$ = electron on atom A (1s orbital). $|2\rangle$ = electron on atom B (1s orbital). In the $\{|1\rangle, |2\rangle\}$ basis, the Hamiltonian matrix is: $H = \begin{bmatrix} \alpha & \beta \\ \beta & \alpha \end{bmatrix}$, where (using typical quantum chemistry numerical values in atomic units, eV, or arbitrary units):

$\alpha = \langle 1|\hat{H}|1\rangle = \langle 2|\hat{H}|2\rangle = -13.6eV$ (on-site energy)
 $\beta = \langle 1|\hat{H}|2\rangle = \langle 2|\hat{H}|1\rangle = -2.4 eV$ (hopping integral, negative)

So: $H = \begin{bmatrix} -13.6 & -2.4 \\ -2.4 & -13.6 \end{bmatrix}$ units eV.

Find Eigenvalues (Energy Levels): $det(H - EI) = 0$

Eigenvalues: $E_1 = -16.0 eV, E_2 = -11.2 eV$.

Correspondence Eigenvectors (Molecular Orbitals)

For $E_1 = -16.0 eV$:

$$\begin{bmatrix} -13.6 - (-16.0) & -2.4 \\ -2.4 & -13.6 - (-16.0) \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = 0$$

$$\begin{bmatrix} 2.4 & -2.4 \\ -2.4 & 2.4 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = 0$$

First row: $2.4 c_1 - 2.4 c_2 \Rightarrow c_1 = c_2$

Normalize: $c_1^2 + c_2^2 = 1 \Rightarrow 2c_1^2 = 1 \Rightarrow c_1 = \frac{1}{\sqrt{2}}$

$$\text{So: } \vec{\varphi}_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Bonding orbital (symmetric, lower energy)

For $E_2 = -11.2 eV$:

$$\begin{bmatrix} -13.6 - (-11.2) & -2.4 \\ -2.4 & -13.6 - (-11.2) \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = 0$$

$$\begin{bmatrix} -2.4 & -2.4 \\ -2.4 & -2.4 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = 0$$

First row: $-2.4 c_1 - 2.4 c_2 \Rightarrow c_1 = -c_2$

Normalize: $c_1^2 + c_2^2 = 1 \Rightarrow 2c_1^2 = 1 \Rightarrow c_1 = \frac{1}{\sqrt{2}}, c_2 = -\frac{1}{\sqrt{2}}$.

$$\text{So: } \vec{\varphi}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

Antibonding orbital (antisymmetric, higher energy)

Verify Matrix Diagonalization:

Check $U^T H U = \text{diag}(E_1, E_2)$:

Construct eigenvector matrix $U = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$

First column: $HU = \left\{ \begin{bmatrix} -13.6 & -2.4 \\ -2.4 & -13.6 \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \right\} = (-16.0) \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \rightarrow$
eigenvalue -16.0

Second column:

$HU = \left\{ \begin{bmatrix} -13.6 & +2.4 \\ -2.4 & +13.6 \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \right\} =$
 $(-11.2) \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \rightarrow$ eigenvalue -11.2

Then $U^T H U = U(HU) =$

$$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} -16.0 & -11.2 \\ -16.0 & 11.2 \end{bmatrix} =$$

$$\frac{1}{2} \begin{bmatrix} -32.0 & 0 \\ 0 & -22.4 \end{bmatrix} = \begin{bmatrix} -16 & 0 \\ 0 & 11.2 \end{bmatrix}$$

Verified. This simple 2x2 Hamiltonian matrix encapsulates the core principles of chemical bonding, energy level splitting, and molecular orbital formation all through basic numerical matrix diagonalization.

4.2. Application of Adjacency Matrix of a Weighted Directed Graph (Scenario: City Route Network):

Consider a small city with 4 locations (nodes):

- A = Airport
- B = Bus Station
- C = City Center
- D = Downtown Hotel

The roads (edges) have distances in kilometres (weights) and are one-way (directed) due to traffic rules. Find shortest path from A to D.

Defined Network Connections:

- A → B: 5 km (Airport to Bus Station)
- A → C: 8 km (Airport to City Center)
- B → C: 3 km (Bus Station to City Center)
- B → D: 7 km (Bus Station to Downtown Hotel)
- C → D: 2 km (City Center to Downtown Hotel)
- D → A: 4 km (Downtown Hotel to Airport)

No other direct paths exist.

Construct the Adjacency Matrix: Order of nodes:

[A, B, C, D]

Matrix M of size 4x4: $M = \begin{bmatrix} 0 & 5 & 8 & 0 \\ 0 & 0 & 3 & 7 \\ 0 & 0 & 0 & 2 \\ 4 & 0 & 0 & 0 \end{bmatrix}$

Entry M_{ij} = weight of edge from node ii to node jj (0 if no direct edge).

From original M: Direct $M_{AD}=0 \rightarrow$ no path

The shortest distance from the Airport (A) to the Downtown Hotel (D) is 10 km. Two optimal routes are obtained:

- A → C → D
- A → B → C → D

This example demonstrates how adjacency matrices provide an efficient mathematical framework for representing transportation networks and solving shortest-path problems in graph theory, computer science, and urban traffic management.

4.3 Application of Jacobian Matrix for a 2-Input, 2-Output System:

Problem Statement: Consider a nonlinear transformation from input vector (x,y) to output vector

$$(u,v): \begin{cases} u = x^2 + \sin y \\ v = xy + e^x \end{cases}$$

We want to find the Jacobian matrix at the point P=(1,0).

For a vector function $f(x_1, x_2, \dots, x_n)$ with m outputs, the Jacobian matrix J is:

$$J(x,y) = \frac{\partial(u,v)}{\partial(x,y)} = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{bmatrix}$$

So, the general Jacobian matrix is: $J(x,y) = \begin{bmatrix} 2x & \cos y \\ y + e^x & x \end{bmatrix}$

Evaluate Jacobian matrix at Point P (1,0): $J(1,0) = \begin{bmatrix} 2 & 1 \\ e & 1 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 2.718 & 1 \end{bmatrix}$

At point P (1,0), the Jacobian is the best linear approximation of the transformation near P:

$$\begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix} = J(1,0) \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

That is:

$$\Delta u \approx 2\Delta x + 1\Delta y$$

$$\Delta v \approx e\Delta x + 1\Delta y$$

Summary of Jacobian matrix at the point P(1,0) is:

Quantity	Value at P(1,0)
Jacobian matrix	$\begin{bmatrix} 2 & 1 \\ e & 1 \end{bmatrix}$
Determinant	$2 - e \approx -0.718$
Area scaling factor	0.718 (shrinking)
Orientation	Reversed (negative sign)
Linear model	$\Delta u = 2\Delta x + \Delta y, \Delta v = e\Delta x + \Delta y$

4.4 Application of Confusion matrix (Medical Diagnosis Test for a Disease):

A hospital develops a test to detect Disease X. They test 100 patients. Actual Condition: Some patients have the disease (Positive), some do not (Negative). Test Result: The test predicts Positive (has disease) or Negative (no disease).

4.4.1: The Actual vs. Predicted Data (After Testing)

Actual Condition	Number of Patients	Test Predicted
Has Disease	40	Positive
Has Disease	10	Negative
No Disease	5	Positive
No Disease	45	Negative

4.4.2: Fill the Confusion Matrix:

	Predicted: Positive	Predicted: Negative
Actual: Positive	True Positive (TP)	False Negative (FN)
Actual: Negative	False Positive (FP)	True Negative (TN)

From the data above:

TP = Actually diseased & Test said diseased = 40

FN = Actually diseased & Test said healthy = 10

FP = Actually healthy & Test said diseased = 5

TN = Actually healthy & Test said healthy = 45

So, the numerical confusion matrix is:

$$\text{Confusion Matrix} = \begin{bmatrix} 40 & 10 \\ 5 & 45 \end{bmatrix}$$

In standard table form:

	Predicted Positive	Predicted Negative
Actual Positive	40 (TP)	10 (FN)
Actual Negative	5 (FP)	45 (TN)

$$\text{Total patients} = 40+10+5+45=100$$

4.4.3 Result of model:

1. Accuracy (Overall correct): $\text{Accuracy} = \frac{TP+TN}{\text{Total}} = \frac{40+45}{100} = \frac{85}{100} = 0.85 = (85\%)$

2. Precision: $\text{Precision} = \frac{TP}{TP+FP} = \frac{40}{40+5} = \frac{40}{45} = 0.889 = (88.9\%)$

3. Recall / Sensitivity (How well test finds actual diseased):

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{40}{40+10} = \frac{40}{50} = 0.80 = (80\%)$$

4. Specificity (How well test identifies healthy people):

$$\text{Specificity} = \frac{TN}{TN+FP} = \frac{45}{45+5} = \frac{45}{50} = 0.90 = (90\%)$$

5. F1-Score (Harmonic mean of Precision & Recall):

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 \times 0.889 \times 0.80}{0.889 + 0.80}$$

$$F1 = \frac{2 \times 0.7112}{1.689} = 0.842 = (84.2\%)$$

4.4.4 Interpretation of result:

Term	Value	Meaning
TP = 40	Correctly identified diseased patients	Good
FN = 10	Missed diseased patients (false negative)	Dangerous: Sick person told they are healthy
FP = 5	Healthy people wrongly told they have disease	Causes unnecessary anxiety/further testing
TN = 45	Correctly identified healthy patients	Good

The confusion matrix condenses all four outcomes of a classifier into a single grid. From this 2x2 numerical matrix, you can compute accuracy, precision, recall, specificity, and F1-score giving a complete picture of model performance beyond just "percent correct."

4.5 Application of Image Matrix Model: Image Matrix showing how an image is represented, processed, and understood using matrix operations.

1. Original Image: 6x6 Grayscale Chess Pattern:

A simple chessboard pattern (dark = 0, light = 255):

$$I = \begin{bmatrix} 255 & 0 & 255 & 0 & 255 & 0 \\ 0 & 255 & 0 & 255 & 0 & 255 \\ 255 & 0 & 255 & 0 & 255 & 0 \\ 0 & 255 & 0 & 255 & 0 & 255 \\ 255 & 0 & 255 & 0 & 255 & 0 \\ 0 & 255 & 0 & 255 & 0 & 255 \end{bmatrix}$$

Each entry = pixel intensity (0 = black, 255 = white)

Size: 6 rows x 6 columns → 36 total pixels

Operation 1: Brightness Adjustment (Add scalar):

Make image 50% brighter → Add 128 to each pixel, clamped to 255:

$$I_{\text{bright}} = I + 128(\text{cap at } 255)$$

Result:

$$I = \begin{bmatrix} 255 & 128 & 255 & 128 & 255 & 128 \\ 128 & 255 & 128 & 255 & 128 & 255 \\ 255 & 128 & 255 & 128 & 255 & 128 \\ 128 & 255 & 128 & 255 & 128 & 255 \\ 255 & 128 & 255 & 128 & 255 & 128 \\ 128 & 255 & 128 & 255 & 128 & 255 \end{bmatrix}$$

Original 0 → becomes 128 (gray)

Original 255 → stays 255

Operation 2: Edge Detection (Convolution with Kernel)

Define Sobel X kernel (detects vertical edges): $K =$

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Apply convolution at position (row 2, col 2) of I:

Region of interest (3×3 window centered at row 2, col 2):

$$W = \begin{bmatrix} 255 & 0 & 255 \\ 0 & 255 & 0 \\ 255 & 0 & 255 \end{bmatrix}$$

Element-wise multiply with K: $(-1 \times 255) + (0 \times 0) + (1 \times 255) +$

$$(-2 \times 0) + (0 \times 255) + (2 \times 0) +$$

$$(-1 \times 255) + (0 \times 0) + (1 \times 255)$$

$$= (-255 + 0 + 255) + (0 + 0 + 0) + (-255 + 0 + 255) = (0 + 0 + 0) = 0$$

At a checkerboard corner, vertical edge response = 0 (symmetric pattern). Other positions would give non-zero values (true edges).

Operation 3: Matrix Transpose (Rotate 90° conceptually)

Transpose of I:

$$I^T = \begin{bmatrix} 255 & 0 & 255 & 0 & 255 & 0 \\ 0 & 255 & 0 & 255 & 0 & 255 \\ 255 & 0 & 255 & 0 & 255 & 0 \\ 0 & 255 & 0 & 255 & 0 & 255 \\ 255 & 0 & 255 & 0 & 255 & 0 \\ 0 & 255 & 0 & 255 & 0 & 255 \end{bmatrix}$$

Operation 4: Matrix Multiplication (Blur/Filter)

Define a simple 3×3 averaging kernel:

$$B = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Apply to top-left 3×3 block of I:

$$Block = \begin{bmatrix} 255 & 0 & 255 \\ 0 & 255 & 0 \\ 255 & 0 & 255 \end{bmatrix}$$

$$\text{Sum of block} = 255+0+255+0+255+0+255+0+255=1275$$

$$\text{Average} = 1275/9 \approx 141.67$$

So new pixel value at center = 142 (rounded)

These smoothest sharp transitions → blur effect.

Summary Table of Matrix Operations on Image:

Operation	Matrix Formula	Effect
Brightness	$I+c$	Increases all pixel values
Edge detection	$I*K$ (Convolution)	Highlights intensity changes
Transpose	I^T	Rotates 90° + mirror
Blur	$I*B$ (Mean filter)	Reduces noise, softens edges
Scaling (resize)	$I_{new} = SIS^T$	Changes dimensions

An image is just a matrix of numbers. Every filter, effect, or transformation you see in Photoshop or Instagram is a matrix operation addition, multiplication, convolution, transpose, or eigenvalue decomposition (PCA for compression).

4.6 Application of Transition Matrix: Transition Matrix (also known as a Stochastic Matrix or Markov Matrix):

We will use a two-state weather model: "Sunny" (S) and "Rainy" (R).

Daily Weather Transition:

Step 1: Define the States

State 1: Sunny (S)

State 2: Rainy (R)

Step 2: Define the Transition Rules (Probabilities)

From historical data:

If today is Sunny:

Tomorrow is Sunny with probability 0.8

Tomorrow is Rainy with probability 0.2

If today is Rainy:

Tomorrow is Sunny with probability 0.6

Tomorrow is Rainy with probability 0.4

Step 3: Write the Transition Matrix

The transition matrix P is organized as:

$$P = \begin{bmatrix} P(S \rightarrow S) & P(R \rightarrow S) \\ P(R \rightarrow S) & P(R \rightarrow R) \end{bmatrix}$$

Note: Sometimes the matrix is written with "From" states as columns and "To" states as rows. Below is the standard row-stochastic form (rows sum to 1, each

entry P_{ij} = probability of going from state i to state j):

$$P = \begin{bmatrix} 0.8 & 0.2 \\ 0.6 & 0.4 \end{bmatrix}$$

Row 1 (Today = Sunny): Sunny tomorrow (0.8), Rainy tomorrow (0.2)

Row 2 (Today = Rainy): Sunny tomorrow (0.6), Rainy tomorrow (0.4)

Verification: Each row sums to $0.8+0.2=1.00$ and $0.6+0.4=1$.

Numerical Example: Predicting Weather

Given: Today is Sunny.

Represent initial state as a row vector: $v_0 = [1,0]$

Step 4: Predict Tomorrow's Weather: $v_1 = v_0 \times P$

$$v_1 = [1,0] \begin{bmatrix} 0.8 & 0.2 \\ 0.6 & 0.4 \end{bmatrix}$$

$$= [1 \times .8 + 0 \times .6 \quad 1 \times .2 + 0 \times .4] = [0.8 \quad 0.2]$$

Interpretation: Tomorrow is Sunny with 80% probability, Rainy with 20%.

Step 5: Predict the Day After Tomorrow (2 days from now)

$$v_2 = v_1 \times P$$

$$v_2 = [0.8 \quad 0.2] \begin{bmatrix} 0.8 & 0.2 \\ 0.6 & 0.4 \end{bmatrix}$$

$$= [.8 \times .8 + .2 \times .6 \quad .8 \times .2 + .2 \times .4]$$

$$= [0.76 \quad 0.24]$$

Interpretation: Two days from now: Sunny 76%, Rainy 24%.

Step 6: Find the Steady-State (Stationary Distribution)

After many days, the probabilities converge to a fixed vector π such that: $\pi P = \pi$

and $\pi_1 + \pi_2 = 1$

Let $\pi = [x \quad y]$.

From $\pi P = \pi$:

$$[x \quad y] \begin{bmatrix} 0.8 & 0.2 \\ 0.6 & 0.4 \end{bmatrix} = [x \quad y]$$

This gives two equations:

$$0.8x + 0.6y = x \rightarrow 0.6y = 0.2x \rightarrow x = 3y$$

$$0.2x + 0.4y = y \rightarrow 0.2x = 0.6y \rightarrow x = 3y \text{ (same)}$$

$$\text{Using } x + y = 1: \quad 3y + y = 1 \Rightarrow 4y = 1 \Rightarrow y =$$

$$0.25 \Rightarrow x = 0.75$$

Steady-state vector: $\pi = (0.75 \quad 0.25)$

Meaning: In the long run, 75% of days are Sunny, 25% are Rainy, regardless of the starting weather.

Summary Table

Day (from today=Sunny)	Sunny Probability	Rainy Probability
Today (0)	1	0
Tomorrow (1)	0.8	0.2
Day after (2)	0.76	0.24
Steady-state (∞)	0.75	0.25

V. RESULT DISCUSSION

The review demonstrates that matrix theory plays a vital role in modern science, technology, engineering, artificial intelligence, and medicine by providing efficient tools for modeling and solving complex real-world problems. Various matrix applications, including Hamiltonian matrices, adjacency matrices, Jacobian matrices, confusion matrices, image matrices, and transition matrices, were discussed to illustrate their practical significance. Hamiltonian matrices explained molecular bonding and energy levels in quantum systems, while adjacency matrices simplified transportation and network optimization problems. Jacobian matrices supported nonlinear system analysis, robotics, and machine learning through local transformation and sensitivity analysis. Confusion matrices proved essential in evaluating medical diagnosis and AI classification performance using measures such as accuracy, precision, recall, and F1-score. Image matrix models demonstrated the role of matrices in digital image processing and deep learning through operations like filtering, convolution, and edge detection. Transition matrices highlighted the importance of probabilistic forecasting and Markov processes in weather prediction and dynamic system modeling. Overall, the review confirms that matrices provide a unified mathematical framework for representing multidimensional relationships, improving computational efficiency, supporting high-dimensional data analysis, and enabling technological advancements in artificial intelligence, healthcare, and scientific research.

VI. CONCLUSION

Matrices have emerged as one of the most powerful and versatile mathematical tools in modern science, technology, engineering, and medicine. This review explored several important matrix applications including Hamiltonian matrices in quantum chemistry,

adjacency matrices in network analysis, Jacobian matrices in nonlinear systems, confusion matrices in medical diagnosis, image matrices in digital image processing, and transition matrices in probabilistic forecasting. Each application demonstrated how matrix operations provide efficient methods for representing, analyzing, and solving complex real-world problems.

The study revealed that matrices play a critical role in simplifying multidimensional computations, improving prediction accuracy, and enabling systematic analysis of interconnected systems. Matrix decomposition, eigenvalue analysis, convolution, transformation, and probabilistic modeling form the mathematical foundation of many modern computational techniques used in artificial intelligence, robotics, medical imaging, quantum computing, transportation systems, and data science. Furthermore, the increasing availability of large-scale data and the rapid advancement of machine learning technologies have significantly expanded the importance of matrix-based methods. Many modern algorithms in deep learning, neural networks, optimization, and predictive analytics rely heavily on matrix algebra for efficient computation and high-dimensional data representation. In medical sciences, matrices contribute to disease diagnosis, imaging technologies, patient monitoring systems, and healthcare decision-making models.

In conclusion, matrix theory is not limited to theoretical mathematics; it serves as a fundamental bridge connecting mathematical concepts with practical technological applications. The interdisciplinary nature of matrices makes them indispensable in solving present and future scientific challenges. Continued research in matrix computation, numerical linear algebra, and AI-driven matrix modeling will further enhance innovation across science, engineering, medicine, and emerging technologies.

ACKNOWLEDGEMENT

The author sincerely thanks Amity University Uttar Pradesh, Greater Noida Campus for providing academic support and a positive research environment. Gratitude is also extended to colleagues, faculty members, and researchers whose valuable guidance and published works contributed to this review. The author further acknowledges the encouragement and

support of family members and well-wishers throughout the research process.

This study was completed independently by the author and did not receive any financial support, sponsorship, or funding from any government or private agency.

Dr. D P Singh

Amity University Uttar Pradesh Greater Noida Campus

REFERENCES

- [1] L. N. Trefethen and D. Bau, *Numerical Linear Algebra*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1997.
- [2] C. Kak and M. Slaney, *Principles of Computerized Tomographic Imaging*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2001.
- [3] D. B. West, *Introduction to Graph Theory*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2001.
- [4] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2003.
- [5] L. Sörnmo and P. Laguna, *Bioelectrical Signal Processing in Cardiac and Neurological Applications*. Amsterdam, Netherlands: Elsevier, 2005.
- [6] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.
- [7] E. Kreyszig, *Advanced Engineering Mathematics*, 10th ed. Hoboken, NJ, USA: Wiley, 2011.
- [8] M. Rowland and T. N. Tozer, *Clinical Pharmacokinetics and Pharmacodynamics: Concepts and Applications*, 4th ed. Philadelphia, PA, USA: Lippincott Williams & Wilkins, 2011.
- [9] J. T. Bushberg, J. A. Seibert, E. M. Leidholdt, and J. M. Boone, *The Essential Physics of Medical Imaging*, 3rd ed. Philadelphia, PA, USA: Lippincott Williams & Wilkins, 2012.
- [10] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 2nd ed. Cambridge, U.K.: Cambridge University Press, 2013.
- [11] H. Anton and C. Rorres, *Elementary Linear Algebra: Applications Version*, 11th ed. Hoboken, NJ, USA: Wiley, 2014.

- [12] Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
- [13] T. Jolliffe and J. Cadima, "Principal component analysis: A review and recent developments," *Philosophical Transactions of the Royal Society A*, vol. 374, no. 2065, pp. 1–16, 2016.
- [14] D. C. Lay, S. R. Lay, and J. J. McDonald, *Linear Algebra and Its Applications*, 5th ed. Boston, MA, USA: Pearson, 2016.
- [15] G. Strang, *Introduction to Linear Algebra*, 5th ed. Wellesley, MA, USA: Wellesley-Cambridge Press, 2016.
- [16] L. J. S. Allen, *Mathematical Epidemiology*. New York, NY, USA: Springer, 2017.
- [17] R. Diestel, *Graph Theory*, 5th ed. Berlin, Germany: Springer, 2017.
- [18] S. S. Rao, *Mechanical Vibrations*, 6th ed. Boston, MA, USA: Pearson, 2017.
- [19] J. J. Craig, *Introduction to Robotics: Mechanics and Control*, 4th ed. Boston, MA, USA: Pearson, 2018.
- [20] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 4th ed. Boston, MA, USA: Pearson, 2018.
- [21] D. J. Griffiths and D. F. Schroeter, *Introduction to Quantum Mechanics*, 3rd ed. Cambridge, U.K.: Cambridge University Press, 2018.
- [22] M. Newman, *Networks*, 2nd ed. Oxford, U.K.: Oxford University Press, 2018.
- [23] U. Alon, *An Introduction to Systems Biology: Design Principles of Biological Circuits*, 2nd ed. Boca Raton, FL, USA: Chapman and Hall/CRC, 2019.
- [24] M. Feinberg, *Foundations of Chemical Reaction Network Theory*. Cham, Switzerland: Springer, 2019.
- [25] D. C. Lay, S. R. Lay, and J. J. McDonald, *Linear Algebra and Its Applications*, 6th ed. Boston, MA, USA: Pearson, 2021.
- [26] D. P. Singh, "An extensive analysis of machine learning models to predict the breast cancer recurrence," *Tuijin Jishu/Journal of Propulsion Technology*, vol. 45, no. 2, pp. 5208–5223, 2024.
- [27] D. P. Singh, "An extensive examination of machine learning methods for identifying diabetes," *Tuijin Jishu/Journal of Propulsion Technology*, vol. 45, no. 2, 2024.
- [28] D. P. Singh, "An extensive analysis of machine learning techniques for predicting the onset of lung cancer," *Tuijin Jishu/Journal of Propulsion Technology*, vol. 45, no. 4, pp. 1419–1433, 2024.
- [29] D. P. Singh, "Comprehensive analysis of machine learning models for cardiovascular disease detection and diagnosis," *Tuijin Jishu/Journal of Propulsion Technology*, vol. 45, no. 4, pp. 2805–2820, 2024.
- [30] D. P. Singh, "A comparative analysis of machine learning techniques for hypertension risk prediction and diagnostic classification," *International Journal of Innovative Research in Technology*, vol. 11, no. 12, pp. 7183–7195, 2025.