**UNIT I: Introduction to Computational Intelligence (CI)**

**Hours: 6 | Course Outcome: CO1**

**1. Introduction to Computational Intelligence**

* CI is a set of nature-inspired computational methodologies and approaches to address complex real-world problems.
* Unlike traditional AI (logic/rule-based), CI focuses on *adaptive*, *intelligent*, and *learning* capabilities.

**2. Paradigms of CI**

* **Fuzzy Logic** – Deals with uncertainty and imprecise information.
* **Neural Networks** – Learn patterns and relationships from data.
* **Evolutionary Computing** – Algorithms inspired by biological evolution (e.g., Genetic Algorithms).
* **Swarm Intelligence** – Inspired by collective behavior of decentralized systems (e.g., Ant Colony Optimization).

**3. Difference between AI and CI**

| **Aspect** | **Artificial Intelligence** | **Computational Intelligence** |
| --- | --- | --- |
| Origin | Symbolic logic, rules | Nature-inspired |
| Adaptiveness | Less adaptive | Highly adaptive |
| Data Handling | Structured | Works well with uncertain/noisy data |

**4. Approaches to CI**

* **Connectionist** (Neural Networks)
* **Evolutionary** (Genetic Algorithms)
* **Fuzzy Systems**
* **Hybrid Models** (e.g., Neuro-Fuzzy systems)

**5. Synergies of CI Techniques**

* Combining multiple CI techniques improves performance.
  + **Example:** Fuzzy-Neural Controllers

**6. Applications of CI**

* Medical Diagnosis
* Financial Forecasting
* Robotics
* Image & Speech Recognition
* Waste Management Systems

**7. Grand Challenges of CI**

* Human-level intelligence
* General-purpose learning
* Real-time CI systems
* Autonomous decision-making

**Case Study: *Intelligent Waste Classification System***

* Uses sensors + fuzzy logic + neural nets to classify and sort waste.

**UNIT II: Fuzzy Logic**

**Hours: 6 | CO2**

**1. Fuzzy Set**

* Introduced by Lotfi Zadeh.
* Unlike classical sets (0 or 1), fuzzy sets allow partial membership between 0 and 1.

**2. Key Concepts**

* **Membership Function** – Curve defining how each input maps to a membership value.
* **Fuzzy Operators** – AND, OR, NOT (using Min, Max, Complement).
* **Fuzziness vs. Probability** – Fuzziness represents vagueness; probability represents likelihood.

**3. Fuzzy Reasoning**

* Uses **linguistic variables** (e.g., Temperature = Hot, Cold).
* **Fuzzy Rules** – IF-THEN rules, e.g., "IF temperature is high THEN fan speed is high."

**4. Fuzzy Inferencing**

* **Fuzzification:** Converts crisp input into fuzzy values.
* **Defuzzification:** Converts fuzzy output back to crisp values.

**5. Fuzzy Logic Controllers (FLCs)**

* Used in control systems (e.g., air conditioners, robots).
* **Types:**
  + *Mamdani-Type FLC* – Easy to interpret
  + *Sugeno-Type FLC* – Computationally efficient

**Case Study: *Object Detection Robot using Fuzzy Controller***

* Uses fuzzy rules to determine distance and direction based on sensor input.

**UNIT III: Evolutionary Computing**

**Hours: 6 | CO3**

**1. Introduction**

* Solves optimization problems using biologically inspired methods.

**2. Key Terminologies**

* **Population** – Set of candidate solutions.
* **Fitness Function** – Evaluates solution quality.
* **Selection, Crossover, Mutation** – Mimic natural evolution.

**3. Evolutionary Algorithms**

* **Genetic Algorithm (GA)** – Binary encoded population
* **Evolution Strategies** – Real-valued parameters
* **Evolutionary Programming** – Focuses on mutation
* **Genetic Programming (GP)** – Evolves programs (tree structures)

**4. Performance Metrics**

* Convergence Rate
* Solution Diversity
* Computational Time

**5. Advanced Topics**

* **Constraint Handling** – Deals with restricted search spaces.
* **Multi-objective Optimization** – Optimizes multiple objectives.
* **Dynamic Environments** – Handles changing problems over time.

**6. Swarm Intelligence**

* Collective behavior inspired from nature:
  + **Ant Colony Optimization (ACO)** – Solves path-finding problems using pheromone trails.

**Case Study: *Artificial Hummingbird Algorithm for Engineering Optimization***

**UNIT IV: Genetic Algorithm (GA)**

**Hours: 6 | CO4**

**1. Terminologies**

* **Chromosome** – Candidate solution
* **Gene** – Part of chromosome (a feature)
* **Allele** – Value of a gene
* **Population** – Group of chromosomes
* **Genotype** – Internal structure
* **Phenotype** – Observable behavior

**2. GA Process**

1. **Initialization** – Random generation of population.
2. **Selection** – Choose parents (e.g., Roulette Wheel).
3. **Crossover** – Combine genes (e.g., Single/Double-point).
4. **Mutation** – Random gene change to add diversity.
5. **Evaluation** – Fitness scoring.
6. **Stopping Condition** – Max generations or satisfactory solution.

**3. GA Representation**

* **Binary Encoding** – 0s and 1s
* **Floating-Point** – Real values (more accurate)

**4. GA Variants**

* **Canonical GA** – Standard GA (Holland Classifier System)
* **Messy GA** – Works with variable-length chromosomes

**5. Applications**

* Optimization, Scheduling, Game Strategy, TSP

**Case Study: *Traveling Salesman Problem using GA***

* Encode path as permutation → Define objective (distance) → Apply GA cycle → Get optimized route.

**UNIT V: Computational Intelligence and NLP**

**Hours: 6 | CO5**

**1. Word Embeddings**

* **Bag of Words (BoW)** – Frequency count of words.
* **TF-IDF** – Highlights rare but important words.
* **Word2Vec / GloVe** – Dense vector representations capturing word context.
* **Neural Word Embedding** – Learned via neural networks.

**2. Neural Machine Translation (NMT)**

* Uses **Seq2Seq models** – Encoder-Decoder architecture with RNNs/LSTMs/Transformers.
* **BLEU Score** – Measures accuracy of machine translation.
* **BERT Score** – Contextual and semantic evaluation.

**3. Neural Style Transfer – Applying artistic styles using deep learning.**

**4. Pretrained NLP Models**

* **BERT (Bidirectional Encoder Representations from Transformers)** – For Q&A, sentiment analysis, etc.

**Case Studies:**

* **Patient Triage using ChatGPT** – Classifies severity of symptoms.
* **Q&A System using BERT** – Finds answers from documents.

**UNIT VI: Artificial Immune Systems (AIS)**

**Hours: 6**

**1. Natural Immune System**

* Detects and neutralizes harmful pathogens (antigens) via:
  + **Antibodies**, **T-cells**, **Memory cells**

**2. Artificial Immune Models**

* Algorithms inspired by biological immune systems.

**3. AIS Algorithms**

* **Clonal Selection Algorithm** – Replication and mutation of high-affinity antibodies.
* **Negative Selection Algorithm** – Detects anomalies by recognizing "non-self".
* **Immune Network Algorithm** – Maintains diversity in solutions.

**4. AIS Theories**

* **Clonal Selection Theory** – Adaptation through cloning and mutation.
* **Network Theory** – Interaction among antibodies.
* **Danger Theory** – System reacts to threats (not just foreign agents).
* **Dendritic Cell Algorithm** – Fusion of signals to detect anomalies.

**5. Applications of AIS**

* Intrusion Detection Systems (IDS)
* Fraud Detection
* Pattern Recognition

**Unit I: Introduction to Distributed Computing**

**1. Fundamentals of Distributed Computing**

* **Characteristics of Distributed Systems**: Multiple autonomous computers communicate and coordinate using message passing. These systems appear as a single coherent system to the user.
* **Issues**: Resource sharing, concurrency, scalability, fault tolerance, transparency.
* **Goals**: Make resources accessible, support resource sharing, openness, scalability, reliability.
* **Types of Distributed Systems**: Client-server, peer-to-peer, three-tier, N-tier, cloud-based systems.
* **Distributed System Models**: Architectural models (client-server, peer-to-peer), interaction models (synchronous/asynchronous communication).

**2. AI and Data Science in Distributed Computing**

* **Distributing Computational Tasks**: Breaking tasks into subtasks distributed over various nodes.
* **Handling Large Volumes of Data**: Distributed storage and processing using technologies like Hadoop and Spark.
* **Parallel Processing**: Simultaneous data processing across multiple nodes.
* **Issues**:
  + **Data Storage and Retrieval**: Ensuring availability and accessibility.
  + **Data Consistency**: Uniform data view across nodes.
  + **Communication Overhead**: Message passing and synchronization delays.
  + **Fault Tolerance**: System resilience to node failures.

**3. Applications**: Predictive Maintenance, Fraud Detection, ITS, Supply Chain Optimization, Energy Management, Healthcare Diagnostics, Customer Behavior Analysis, NLP.

**Case Study**: E-commerce systems using distributed computing for load balancing, recommendation systems, and transaction management.

**Unit II: Distributed Data Management and Storage**

**1. Frameworks and Technologies**

* **Parallel & Distributed Computing**: Running tasks on multiple processors/nodes.
* **Message Passing**: MPI model.
* **Distributed File Systems**:
  + **HDFS**: Stores large datasets across clusters.
  + **GFS**: Google's scalable distributed file system.
* **Cloud Platforms**: AWS, Azure, GCP provide scalable computing.
* **Message Brokers**: Kafka, RabbitMQ; manage communication.
* **Edge Computing**: Data processing near source for faster results.

**2. Data Replication & Consistency**

* **Replication Models**:
  + Eager (synchronous), Lazy (asynchronous), Quorum-based, Consensus-based.
* **Consistency Models**:
  + Strong, Eventual, Read-your-writes, Consistent Prefix, Causal.

**3. Indexing and Retrieval Techniques**

* **DHTs**: Key-value storage across peers.
* **Inverted Indexing**: Efficient text search.
* **Range-based Partitioning**: Organizing data into ranges.
* **Content-based**: Based on file/content characteristics.
* **P2P Indexing**: Peer-based file look-up.
* **Hybrid**: Mix of multiple approaches.

**Case Study**: Healthcare systems managing patient records using distributed storage.

**Unit III: Distributed Computing Algorithms**

**1. Coordination & Communication**

* Algorithms manage communication across nodes to perform tasks collaboratively.

**2. Consensus Algorithms**

* **Viewstamped Replication, RAFT, ZAB, Mencius, Paxos Variants**: Ensure all nodes agree on shared data/state.

**3. Fault Tolerance & Recovery**

* Techniques to recover from hardware or software failures.

**4. Load Balancing & Resource Allocation**

* Algorithms like Weighted Round Robin, Least Connection, Randomized, Dynamic.
* Centralized vs Distributed load balancing.
* Predictive balancing using AI.

**5. AI in Optimization**

* **ML for Resource Allocation**
* **RL for Load Balancing**
* **GA for Task Scheduling**
* **Swarm Intelligence**: Decentralized algorithms like ACO, PSO.

**Case Study**: Weather Prediction using AI-enhanced distributed algorithms.

**Unit IV: Distributed Machine Learning and AI**

**1. Distributed ML Algorithms**

* **Types**:
  + Data Parallelism: Same model across partitions.
  + Model Parallelism: Split model computation.
* **Key Techniques**: Distributed Gradient Descent, Federated Learning, AllReduce, Hogwild, Elastic Averaging SGD.

**2. Tools**

* Spark, GraphLab, TensorFlow, Parallel ML Systems (e.g., Petuum).

**3. AI Integration in Distributed Systems**

* **Intelligent Resource Management**: Smart allocation based on load.
* **Anomaly Detection**: Detect unusual patterns.
* **Predictive Analytics**: Forecasting outcomes.
* **Task Offloading**: Deciding task placement.

**Case Study**: Distributed ML for Fraud Detection.

**Unit V: Big Data Processing in Distributed Systems**

**1. Frameworks**

* Hadoop, Spark, Storm, Samza, Flink.

**2. Processing Techniques**

* **SISD, MISD, SIMD, MIMD, SPMD, MPP**: Different instruction/data models for parallelism.

**3. Scalable Data Ingestion**

* **Types**: Batch, real-time, hybrid.
* **Tools**: Apache Kafka, NiFi, Flume.
* **Challenges**: Latency, format inconsistency.

**4. Real-time & Streaming Analytics**

* **Real-time**: Instant analysis.
* **Streaming**: Continuous data streams.
* **Comparison**: Latency, volume, application fit.
* **AI Applications**: Sentiment analysis, fraud alerts.

**Case Study**: Social media analytics using distributed big data tools.

**Unit VI: Distributed Systems Security and Privacy**

**1. Security Challenges**

* Node compromise, man-in-the-middle attacks, denial-of-service.

**2. Insider Threats**

* Malicious insiders exploiting access.

**3. Secure Communication**

* **TLS/SSL**: Secure transport.
* **PKI**: Certificates for identity.
* **VPN, AMQP**: Secure messaging.

**4. Privacy Techniques**

* **Differential Privacy**: Adds noise.
* **Homomorphic Encryption**: Compute on encrypted data.
* **SMPC**: Compute without revealing inputs.
* **Federated Learning**: Model training without raw data.
* **Anonymization/Pseudonymization**: Obscuring user identity.

**5. AI-based Security**

* **Anomaly Detection**: Find abnormal behavior.
* **Threat Intelligence**: Data-driven threat awareness.
* **Real-time Response**: Dynamic countermeasures.
* **UEBA**: Analyze behavior patterns.

**Case Study**: AI-based security in distributed systems using real-time intrusion detection and threat visualization.