Artificial Intelligence – Semester Project Guidelines

1. Objective

The purpose of this semester project is to provide students with an opportunity to apply **Artificial Intelligence (AI) concepts, algorithms, and design methodologies** to solve real-world or cybersecurity-related problems.

Students will design, implement, and evaluate a complete **AI-driven system**, integrating:

- **Search algorithms**,
- **Constraint satisfaction and optimization**,
- **Supervised and unsupervised learning**, and
- **Reinforcement learning** concepts.

The project will mirror **industry-level workflows**, emphasizing data pipelines, reproducibility, explainability, and AI ethics.

2. Learning Outcomes

By the end of this project, students will be able to:

- Formulate AI problems with measurable goals and evaluation metrics.
- Implement search and learning algorithms efficiently.
- Handle real-world datasets (preprocessing, feature engineering, scaling).
- Develop and evaluate ML and RL models using modern libraries.
- Apply explainable AI techniques (SHAP, LIME).
- Document and present AI-driven solutions using professional reporting standards.

3. Project Lifecycle and Weekly Integration

Weeks	Course Topics	Project Phase / Expected Deliverables
Week 7	*Introduction to AI & Intelligent Agents*	Define problem domain, objectives, and agent type (reflex, model-based, goal-based, or learning agent). Draft **Project Proposal**.
Week 8	*Search Strategies (Uninformed & Informed)*	Implement initial search/optimization-based solution (e.g., BFS, DFS, A*, Best-First). Prepare **Agent Design Report**.
Week 9	*Constraint Satisfaction Problems (CSPs)*	Model part of your system as a CSP (variables, domains, constraints). Implement backtracking or arc consistency.
Week 10	*Machine Learning Foundations & Supervised Learning*	Acquire/prepare dataset. Apply data preprocessing, feature selection, and train at least two supervised models (e.g., Decision Tree, Logistic Regression, Random Forest).

Week 11	*Advanced ML (SVMs, Neural Networks, CNNs)*	Integrate advanced ML or deep learning model. Perform hyperparameter tuning and visualize learning results.
Week 12	*Unsupervised Learning & Reinforcement Learning*	Extend system using clustering or reinforcement learning (Q-learning, DQN). Analyze model behavior on unseen or adversarial data.
Week 13	*Recap & Final Presentation*	Integrate components, finalize report, and deliver **presentation + demo**.

4. Project Categories (Select One Domain)

A. Cybersecurity Intelligence

- **Examples:** Intrusion Detection, Phishing Detection, Network Traffic Analysis
- **Techniques:** Random Forest, SVM, ANN, CNN, Autoencoders
- **Datasets:** UNSW-NB15, CICIDS2017, NSL-KDD
- **Metrics:** Precision, Recall, F1-score, ROC-AUC

B. Optimization & Search Systems

- **Examples:** Routing optimization, resource allocation, scheduling
- **Techniques:** A*, Hill-Climbing, Genetic Algorithms, CSPs
- **Evaluation:** Cost minimization, time efficiency, convergence

C. Intelligent Agent Simulations

- **Examples:** Multi-agent security systems, autonomous defenders, logistics bots
- **Techniques:** Model-based, goal-based, and learning agents
- **Evaluation:** Decision accuracy, adaptability, and computational efficiency

D. Predictive Analytics

- **Examples:** Threat prediction, log-based anomaly detection, behavior forecasting
- **Techniques:** Regression, Ensemble Learning, Time Series (LSTM, Prophet)
- **Evaluation:** RMSE, MAE, R^2, trend accuracy

E. Reinforcement Learning Applications

- **Examples:** Adaptive intrusion response, defense automation, game-based control
- **Techniques:** Q-learning, SARSA, DQN, PPO
- **Evaluation:** Reward convergence, policy stability, exploration balance

F. NLP & Threat Intelligence

- **Examples:** Text classification, malicious domain detection, sentiment analysis
- **Techniques:** TF-IDF, Word2Vec, BERT, Logistic Regression

- **Datasets:** Phishing Email Corpus, URLNet
- **Evaluation:** F1, ROC-AUC, confusion matrix

5. Deliverables and Timeline

Deliverable	Description	Due Week
1. Proposal Report (4–5 pages)	Problem statement, motivation, objectives, dataset source, AI approach, and evaluation plan.	Week 7
2. Progress Report I	Agent/search algorithm implementation with working example.	Week 9
3. Progress Report II	Data preprocessing pipeline + baseline ML models with performance metrics.	Week 8
4. Progress Report III	Integration of advanced ML/DL or RL model with interpretability and optimization.	Week 10
5. Final Report + Demo + Presentation	Full working system, final evaluation, report, slides, and GitHub repository.	Week 13

6. Technical Standards & Tools

Component	Requirements / Recommendations
Language	Python (mandatory for ML/RL tasks)
Libraries	NumPy, Pandas, scikit-learn, TensorFlow, PyTorch, OpenAI Gym, XGBoost
Data Tools	Kaggle, UCI, UNSW-NB15, CICIDS2017
Visualization	Matplotlib, Seaborn, TensorBoard, Plotly
Explainability Tools	SHAP, LIME
Version Control	GitHub/GitLab (with commit logs)
Experiment Tracking (optional)	Weights & Biases (W&B), MLflow
Deployment (optional)	Streamlit, Flask, or Docker

7. Evaluation Rubrics (Detailed and Technical)

Evaluation Component	Description	Weight
1. Problem Definition & Relevance	Problem clarity, industry linkage, measurable objectives	10%
2. System Architecture & Design	Modular pipeline (data → model → evaluation), scalability, architecture diagram	15%

3. Algorithmic Implementation	Correct and efficient implementation of AI algorithms (search, CSP, ML, RL)	20%
4. Data Handling & Feature Engineering	Data quality, preprocessing, normalization, feature extraction, ethics	15%
5. Model Development & Evaluation	Model selection, parameter tuning, comparative analysis, validation (e.g., K-fold CV)	15%
6. Interpretability & Explainability	Use of SHAP/LIME, visualization of model insights, explainable decision-making	10%
7. Documentation & Presentation	Report quality, formatting, technical writing, demo clarity	10%
8. Scalability & Innovation	Creativity, reusability, performance optimization, or deployment	5%
Total		**100%**

8. Detailed Evaluation Matrix

Criteria	Excellent (A)	Good (B)	Fair (C)	Poor (D/F)
Problem Formulation	Domain-specific, measurable, innovative	Clear but lacks technical novelty	Vague objectives	Unclear or trivial
Algorithmic Design	Correct, optimized, modular	Correct but limited optimization	Partial or inefficient	Incorrect implementation
Data Engineering	Robust pipeline, visual EDA, engineered features	Good preprocessing	Minimal effort	Raw/unprocessed
Model Evaluation	Multiple metrics, proper validation	Partial evaluation	Basic accuracy only	No evaluation
Explainability	SHAP/LIME or visualization used	Partial interpretation	Minimal effort	None
Report & Presentation	Industry-grade structure, visuals, clarity	Organized	Limited structure	Disorganized

9. Reporting Format

Each report (especially final) must follow a structured **academic–industrial hybrid** format:

1. **Abstract**
2. **Introduction & Motivation**
3. **Problem Definition & Objectives**
4. **Literature Review**

5. **System Architecture** (with diagram)
6. **Data Description & Preprocessing**
7. **Algorithmic Implementation** (Search / ML / RL / CSP)
8. **Model Evaluation & Comparison**
9. **Explainability & Visualization**
10. **Results & Discussion**
11. **Ethical AI & Limitations**
12. **Conclusion & Future Work**
13. **References (IEEE format)**

10. Technical Best Practices

Aspect	Requirement
Code Quality	Follow PEP8; use docstrings and modular functions.
Reproducibility	Include `requirements.txt` or `environment.yml`.
Version Control	Use Git; frequent commits with messages.
Experiment Tracking	Log results, models, and parameters.
Visualization	Include model performance plots (loss, accuracy, ROC).
AI Ethics	Avoid biased datasets; justify all data sources.

11. Submission Checklist

1. Proposal (Week 3)
2. Progress Reports I–III (Weeks 5, 8, 12)
3. Final Report (Week 13)
4. Presentation Slides + Demo
5. GitHub Repository (Code + README + requirements.txt)
6. (Optional) Streamlit/Flask Deployment or MLflow Logging

12. Bonus Marks (Up to +10%)

Criterion	Bonus
Integration of Explainability (SHAP/LIME)	+2%
ML Experiment Tracking (MLflow/W&B)	+3%
Deployed Prototype (Flask/Streamlit/Docker)	+5%

13. Example Real-World Project Ideas

Domain	Project Title	AI Techniques
Cybersecurity	Intrusion Detection using Random Forest & SHAP	Ensemble Learning, Explainable AI
Cyber Defense	Q-Learning based Intrusion Response Agent	Reinforcement Learning
Optimization	Hybrid A* + CSP Resource Allocation	Heuristic Search, Constraint Propagation
NLP Threat Intel	Phishing Email Detection using BERT	Transformer, Text Classification
Predictive Analysis	Time-Series Attack Forecasting	LSTM, ARIMA
Autonomous Systems	Multi-Agent Security Simulation	Goal-Based & Learning Agents

14. Final Presentation

- **Duration:** 12–15 minutes per team
- **Format:**
    1. Problem introduction & motivation
    2. System architecture & algorithms
    3. Data pipeline & results
    4. Visualization and interpretability results
    5. Demo (if applicable)
    6. Q&A

Evaluation will focus on **clarity, technical mastery, and result justification**.

15. Group Policy

- **Group Size:** 2–3 members.
- Each member must contribute technically (e.g., ML module, visualization, RL model, or documentation).
- **Individual performance** verified via contribution logs, commits, and oral defense.