GEOSPATIAL ARTIFICIAL INTELLIGENCE (GeoAI)

By

Dr. Prafulla Parlewar

Professor and Head, Department of Urban Planning

School of Planning and Architecture, New Delhi

Course Rationale

This course introduces GeoAI as the integration of spatial data, machine learning, deep learning and graph-based intelligence for solving problems in the built and natural environment. The emphasis is on vector, tabular, network, sensor, mobility, environmental, urban and regional datasets. Students learn each major method through KNIME-based codeless workflows and equivalent Python implementations, culminating in research project.

Course Objectives

- Understand GeoAI concepts, spatial data structures, spatial dependence and responsible AI.
- Build reproducible codeless machine learning workflows in KNIME and equivalent workflows in Python.
- Apply classical, probabilistic, ensemble, neural, sequential and graph-based models to spatial problems.
- Evaluate GeoAI models using appropriate predictive, spatial and interpretability measures.
- Develop a discipline-specific research project suitable for submission to a peer-reviewed conference.

Six-Module Course Structure

Module 1: Foundations of GeoAI and Spatial Data Analytics
Weeks 1-2

Module Content

- GeoAI definition, scope, evolution and applications in spatial disciplines.
- Spatial data models: points, lines, polygons, networks, grids, trajectories and spatio-temporal tables.
- Spatial relationships, topology, scale, spatial autocorrelation, heterogeneity and the modifiable areal unit problem.

- Data sources: census, surveys, OpenStreetMap, urban services, mobility, traffic, environmental sensors, IoT and administrative records.
- Data quality, missing values, outliers, encoding, normalization, feature construction and spatial joins.
- Introduction to QGIS / ArcGIS for preparation and visualization, KNIME for analytics, and Python with pandas, GeoPandas and scikit-learn.
- Ethics, fairness, privacy, explainability and reproducibility in GeoAI.
- GeoAI Visualization in KNIME, Python and ArcGIS/QGIS

Applied Work

- Prepare and document a spatial dataset using GIS, KNIME and Python.
- Create a reproducible data-preparation workflow and a spatial exploratory analysis report.

Module 2: Codeless Machine Learning in KNIME and Python Equivalents
Weeks 3-5

Module Content

- KNIME interface, nodes, ports, workflow organization, components, metanodes and reusable workflows.
- Data ingestion, cleaning, transformation, joins, rule engines, feature engineering and visual analytics in KNIME.
- Machine learning workflow: problem formulation, train-test split, spatial cross-validation, model training, tuning and evaluation.
- Artificial neural networks: perceptron, multilayer networks, activation functions, loss functions, backpropagation and regularization.
- ANN implementation through KNIME deep-learning integrations and Python using TensorFlow / Keras or PyTorch.
- Convolutional Neural Networks: Image Processing, Image classification, segmentation, prediction
- Regression: linear regression, regularized regression, decision trees, random forest, gradient boosting and XGBoost.
- Classification: logistic regression, k-nearest neighbours, decision trees, random forest, support vector machines and gradient boosting.
- Unsupervised learning: k-means, hierarchical clustering, DBSCAN, principal component analysis and anomaly detection.
- Equivalent Python workflows using scikit-learn pipelines, preprocessing, model selection and reproducible notebooks.

Applied Work

- Build a complete KNIME workflow for a spatial regression or classification problem.

- Reproduce the same model in Python and compare accuracy, transparency, runtime and reproducibility.

Module 3: Probabilistic and Bayesian GeoAI
Weeks 6-7

Module Content

- Probability concepts for classification and spatial decision-making.
- Naive Bayes models: Gaussian, multinomial and Bernoulli variants; assumptions, advantages and limitations.
- Bayesian reasoning, prior and posterior probabilities, likelihood and uncertainty updating.
- Bayesian networks: directed acyclic graphs, conditional dependencies, structure design and probabilistic inference.
- Applications to land-use suitability, hazard risk, travel behaviour, environmental exposure and infrastructure failure.
- Implementation in KNIME using probabilistic classifiers and workflow components; implementation in Python using scikit-learn and Bayesian-network libraries.
- Calibration, uncertainty communication and sensitivity analysis.

Applied Work

- Develop a Naive Bayes classifier and compare it with a non-probabilistic baseline.
- Construct a small Bayesian network for a spatial decision problem and interpret conditional dependencies.

Module 4: Sequential and Graph-Based GeoAI
Weeks 8-10

Module Content

- Recurrent neural networks, LSTM and GRU for traffic flow, mobility demand, air quality, energy use and urban time-series forecasting.
- Sequence preparation, sliding windows, temporal validation and forecasting metrics.
- Graph fundamentals: nodes, edges, adjacency, centrality, communities and graph embeddings.
- Graph neural networks: graph convolutional networks, GraphSAGE, graph attention networks and spatio-temporal GNN concepts.
- Applications to road networks, public transport networks, pedestrian systems, infrastructure systems, regional interaction and service accessibility.
- KNIME-Python integration for models not available as native nodes, including Python Script nodes, model serialization and reusable components.

Applied Work

- Create an ANN model in both KNIME and Python.
- Build either an RNN/LSTM forecasting model or a GNN-based network prediction prototype.

Module 5: Advanced GeoAI, Model Evaluation and Spatial Decision Support
Weeks 11-12

Module Content

- Ensemble learning, stacking, voting, bagging and boosting.
- Spatial feature engineering: proximity, density, accessibility, centrality, morphology, land-use mix and neighbourhood indicators.
- Spatially aware validation: blocked cross-validation, leave-location-out validation and transferability testing.
- Metrics for regression, classification, clustering, forecasting and imbalanced data.
- Model interpretation: feature importance, permutation importance, partial dependence, SHAP and local explanations.
- Uncertainty, robustness, bias, data leakage and generalization across neighbourhoods, cities and regions.
- Multi-criteria decision support, scenario analysis, optimization and model-to-map communication.
- Deployment concepts: KNIME components, Python APIs, dashboards and reproducible research packages.

Applied Work

- Compare at least three models using spatially appropriate validation.
- Prepare an explainability and uncertainty note supported by maps, charts and decision implications.

Module 6: Applied GeoAI Studio and Research Project
Weeks 13-15

Module Content

- GeoAI applications for architecture and built form: typology classification, space-use prediction, building performance, facility placement and design evaluation.
- GeoAI applications for transport: demand forecasting, route choice, accessibility, congestion, crash risk and network resilience.
- GeoAI applications for environment: air quality, heat exposure, flooding, waste systems, urban ecology and environmental justice.
- GeoAI applications for urban and regional planning: compactness, polycentricity, land-use suitability, growth modelling, service equity, regional interactions and policy scenarios.

Applied Work

- Complete a discipline-specific GeoAI research project.
- Submit a conference-paper-format manuscript, reproducible workflow, data documentation, maps and oral presentation.

Research-Project Framework

The semester project must be equivalent in scope and quality to a conference research paper. Individual or two-student interdisciplinary teams may be permitted, provided individual contributions are documented.

Required Project Components

- Problem statement linked to architecture, transport, environment, urban planning or regional planning.
- Focused literature review identifying a clear research gap.
- Research questions or hypotheses and a conceptual framework.
- Justified spatial dataset, data dictionary and reproducible preprocessing pipeline.
- At least one baseline model and one advanced model selected from the course.
- Implementation in KNIME, Python, or a documented hybrid KNIME-Python workflow.
- Appropriate non-spatial and spatial validation, error analysis and uncertainty assessment.
- Explainability analysis and translation of results into design, planning or policy implications.
- Original maps, figures, tables and a reproducibility package.
- Conference-format paper of approximately 5,000-7,000 words and a 10-12 minute presentation followed by viva.

Suggested Project Themes

Architecture and Built Environment

- Predict building energy use, thermal comfort or daylight performance using building, urban-form and contextual variables.
- Classify building, block or neighbourhood typologies using spatial, morphological and functional indicators.
- Optimize facility location, public-space performance or design alternatives using machine learning and spatial accessibility.

Transport Planning

- Forecast traffic speed, passenger demand or transit ridership using RNN, LSTM or GRU models.
- Model crash risk and transport safety using Naive Bayes, ensemble learning or Bayesian networks.

- Analyse public-transport and street networks using graph metrics, network science and graph neural networks.

Environmental Planning and Management

- Forecast air quality, urban heat, noise or environmental exposure using spatio-temporal machine learning.
- Model flood, waste, ecological or pollution risk using probabilistic, ensemble and explainable ML approaches.
- Assess environmental justice and unequal exposure through interpretable GeoAI and spatial equity indicators.

Urban Planning

- Model compact-city structure, urban form, polycentricity or mixed-use development using spatial indicators and machine learning.
- Assess urban service accessibility, walkability, liveability and spatial equity using network and graph-based methods.
- Predict land-development pressure, redevelopment potential or infrastructure demand using interpretable GeoAI.

Regional Planning

- Model regional interaction, migration, commuting and settlement systems using spatial and graph-based methods.
- Evaluate regional growth suitability, infrastructure priorities and development scenarios using Bayesian networks and machine learning.
- Analyse regional disparities, service gaps and economic or environmental resilience using multi-source spatial data.

Heritage Conservation and Management

- Develop heritage-building condition, vulnerability or conservation-priority models using spatial indicators and probabilistic ML.
- Analyse historic urban landscapes, heritage precincts and cultural-route networks using clustering, graph analysis and GeoAI.
- Create decision-support models for adaptive reuse, visitor management, risk preparedness and heritage-impact assessment.

Assessment Scheme

Component	What will be assessed	Marks in elective
Research proposal and literature review	Research gap, question, objectives, framework and sources	20

Methodology and reproducible workflow review	Data, preprocessing, baseline model, advanced model and validation plan	40
Conference-paper-equivalent final project and submission for conference	Manuscript, results, maps, interpretation and reproducibility package	20
Final Presentation	Research communication	20

Learning Outcomes

- Explain the foundations, opportunities and limitations of GeoAI for spatial disciplines.
- Construct end-to-end machine learning workflows using KNIME and Python.
- Apply Naive Bayes, Bayesian networks, classical ML, ensemble methods, ANN, RNN and GNN approaches to spatial problems.
- Use spatially appropriate validation, explainability and uncertainty assessment.
- Integrate GeoAI outputs into architecture, transport, environmental, urban and regional decision-making.
- Produce and defend a reproducible research study in conference-paper format.

Core Software and Libraries

KNIME: Analytics Platform, components, workflow automation, Python integration and deep-learning extensions. **Python:** pandas, GeoPandas, scikit-learn, XGBoost, TensorFlow/Keras or PyTorch, NetworkX, PyTorch Geometric or DGL, SHAP and Bayesian-network libraries. **GIS:** QGIS and/or ArcGIS for spatial preparation, cartography and communication.