

Secure Mobile AI for iOS and Android

On-device inference, efficient integration and responsible app design

FA-0690 · 2 days

Agentic & Generative AI · Software Development · Cybersecurity

Introduction

This two-day course helps experienced mobile developers evaluate and integrate selected AI features in iOS and Android applications. It compares on-device, cloud and hybrid approaches, then uses a bounded sample integration to examine conversion, profiling, model packaging and updates.

Vision, language, recommendation and generative use cases are surveyed rather than all built from scratch. Privacy, bias, consent, model integrity and secure data handling are treated as design and testing concerns. On-device execution can reduce data transmission but does not automatically guarantee privacy, fairness or legal compliance.

Learning outcomes

- Select a suitable on-device, cloud or hybrid approach using latency, resource, privacy and update constraints.
- Compare LiteRT, Core ML, ML Kit and platform-specific generative APIs, including device and feature availability.
- Integrate one selected AI feature in a supplied mobile sample.
- Measure inference latency, memory and energy trade-offs and evaluate suitable model optimisations.
- Plan model packaging, updates and integrity checks.
- Identify bias, permission, consent, data-leakage and tampering risks and propose appropriate review and testing.

Prerequisites

- Solid experience in native mobile development (Android with Kotlin/Java, iOS with Swift/Objective-C).
- Basic understanding of ML/AI concepts (what a neural network is, what overfitting means, etc.).
- Familiarity with standard software engineering practices: version control, testing, CI/CD.
- Some understanding of data handling (APIs, privacy, data permissions) though not necessary to be expert.

Hands-on work assumes a prepared development environment for one chosen platform; demonstrations compare the other platform.

Training outline

1. Day 1 – Mobile ML foundations and deployment choices

- Supervised vs unsupervised learning: classification, regression, clustering
- Neural networks basics: architecture, forward & backward pass, activation, loss functions
- Generative AI: what it is, generative models vs discriminative; recent trends applicable to mobile (e.g. diffusion, lightweight LLMs)
- Performance metrics: accuracy, precision/recall, F1, latency, inference time, model size, energy cost
- Overfitting, underfitting, generalization; dataset quality issues (distribution shift, bias)

On-device, cloud and hybrid architectures

- Latency, bandwidth, offline behaviour, privacy, model size, operating cost and update pathways.
- LiteRT, the successor to TensorFlow Lite, and Core ML for model conversion and inference.
- ML Kit APIs by platform and feature; Android GenAI APIs use Gemini Nano through AICore on supported devices.
- Apple Foundation Models and other platform APIs: check system, model and device availability rather than assuming universal support.
- Cloud fallback or split processing where appropriate; evaluate the added transmission and availability requirements.

2. Day 1 – AI features and a selected sample integration

Computer Vision

- Image classification, object detection, segmentation
- Use-cases: AR overlays, pose detection, OCR, face detection
- SDKs / examples: Vision frameworks (iOS), ML Kit / TensorFlow Lite for Android, open models

Natural Language Processing (NLP)

- Tokenization, embeddings, simple transformer-based models
- Tasks: text classification, summarization, translation, voice assistants, speech-to-text / text-to-speech
- Consider on-device vs cloud approach depending on resource & privacy constraints

Recommendation Engines & Personalization

- Collaborative vs content-based vs hybrid recommendations
- Collecting & processing user behaviour data in privacy preserving ways
- Model freshness, feedback loops

Generative Features (if time)

- Lightweight generation: small LLMs, prompt-based (on-device or via API)
- Use cases: autocomplete, suggestions, image style transfer

Build one selected feature from a prepared model or API; other use cases are demonstrations or design exercises.

3. Day 2 — Tooling, profiling and model deployment

- Set up suitable ML Kit, LiteRT or Core ML tooling for the chosen sample.
- Check supported model conversion paths from TensorFlow or PyTorch; unsupported operators may require adaptation.
- Profile latency, memory and energy on representative devices.
- Discuss quantisation, pruning and distillation as model-dependent techniques, not automatic quality-preserving switches.
- Evaluate acceleration, app-packaged versus downloaded models, versioning, integrity checks and recovery from failed updates.
- Data collection, feedback loops and model freshness; federated learning as a conceptual option.
- Model-IP and tampering risks: client-side protection reduces risk but cannot guarantee prevention of reverse engineering.

4. Day 2 — Privacy, fairness and security review

- Minimise collected data, use platform permissions and suitable consent, and protect local storage and network transmission.
- Assess demographic, device and edge-case performance; review distribution shift and bias mitigation.
- Provide meaningful user explanations and appropriate logs without exposing sensitive inputs.
- Evaluate adversarial inputs, model leakage, API misuse, tampering and update integrity.
- Identify applicable privacy and AI-law questions for qualified review; no universal compliance or certification claim.

- Review the sample integration and document security, resource and evaluation trade-offs.