1. **ML Goals**

Classification Granularity
https://www.oreateai.com/blog/understanding-data-granularity-the-key-to-effective-analysis/20925c0082873f5e4ec9e5f4d31a1698#:~:text=High%20(Fine)%20Granularity:%20This,trend%20identification%20without%20overwhelming%20detail.

	High Granularity (Fine-Grained)	**Low Granularity (Coarse-Grained)**
Pros	- High-Precision Insights - Reduced "Shortcut Learning"	- Better Generalization - Easier to Manage
Cons	- Overfitting Risk - High Data Requirements	- Missed Subtleties - Limited Insights

For this Project: Use intermediate or low granularity. This balances model stability with interpretability and avoids excessive data requirements.

Classification Type
(https://www.analyticssteps.com/blogs/binary-and-multiclass-classification-machine-learning)

Parameters	**Binary classification**	**Multi-class classification**
No. of classes	It is a classification of two groups, i.e. classifies objects in at most two classes.	There can be any number of classes in it, i.e., classifies the object into more than two classes.
Algorithms used	The most popular algorithms used by the binary classification are- • Logistic Regression • k-Nearest Neighbors • Decision Trees • Support Vector Machine • Naive Bayes	Popular algorithms that can be used for multi-class classification include: • k-Nearest Neighbors • Decision Trees • Naive Bayes • Random Forest. • Gradient Boosting
Examples	Examples of binary classification include- • Email spam detection (spam or not). • Churn prediction (churn or not). • Conversion prediction (buy or not).	Examples of multi-class classification include: • Face classification. • Plant species classification. • Optical character recognition.

For this Project: multi-class classification, since we want to distinguish between multiple activity types.

ML can have two parallel paths:
- Supervised: multi-class classification for known activity types.

- Unsupervised: anomaly detection for unknown abnormal flows.

---

2. Model Selection

A) **Supervised learning:**

Decision Trees, Random Forest, Gradient Boosting, SVM, KNN, Logistic Regression.
https://www.datacamp.com/blog/classification-machine-learning

Random Forest	Works well with tabular data, robust to overfitting, provides feature importance, interpretable.
Gradient Boosting	High accuracy, handles imbalanced classes well, but more complex and slower.
Decision Tree	Simple and interpretable, good baseline model.
SVM / KNN	Suitable for small datasets; SVM sensitive to scaling, KNN sensitive to noise.
Logistic Regression	Simple and interpretable linear classifier, good baseline, requires well-scaled features.

Based on Project requirements :

- Primary model: Random Forest
- Backup model: Gradient Boosting

B) Unsupervised Learning:

Isolation Forest, One-Class SVM, Autoencoders.

Isolation Forest	Lightweight, effective on tabular flow data, isolates anomalies efficiently.
One-Class SVM	Works for anomaly detection but scales poorly for large datasets.
Autoencoders	Can detect anomalies via reconstruction error, suitable if dataset grows and deep learning is feasible.

Based on Project requirements:

- Primary model: Isolation Forest

Evaluation Metrics

- Multi-class classification: Accuracy, Precision, Recall, F1-score, Confusion matrix
- Anomaly detection ML: ROC-AUC

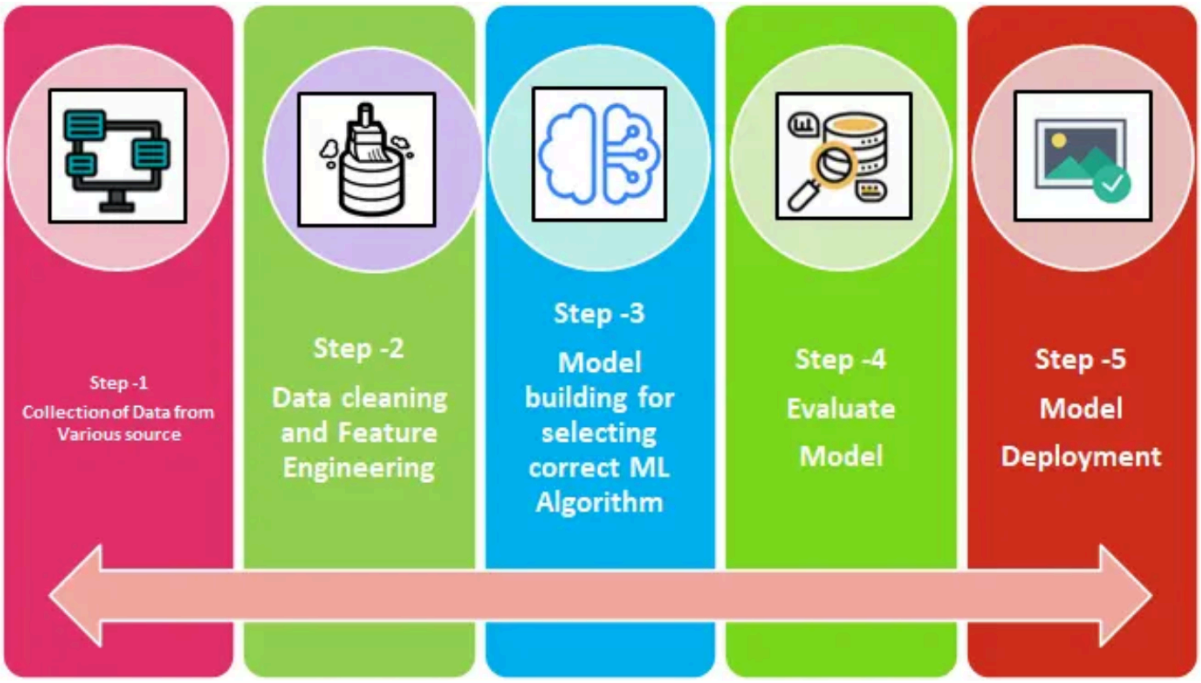
https://www.kaggle.com/code/nkitgupta/evaluation-metrics-for-multi-class-classification

3. Model Training & Validation

- Train/test split: Should keep to either: 70/30 or 80/20 split (https://builtin.com/data-science/train-test-split) (https://medium.com/data-science/beyond-80-20-a-practical-guide-to-train-test-splits-in-machine-learning-5fc62ebe276f)

- Decide evaluation metrics:
    - Multi-class: Accuracy, Precision, Recall, F1-score per class; Confusion matrix.
    - Anomaly detection: ROC-AUC, Precision@K.

- Handling overfitting: Regularization, pruning, ensemble methods, early stopping
    - Random Forest: maximum tree depth, minimum number of samples required at each leaf node
    - Gradient Boosting: early stopping



Machine learning steps: A complete guide for beginner in ML

https://www.labellerr.com/blog/machine-learning-steps-a-complete-guide-to-the-ml-process/

ML Process for this project::
Step 1: Data Preparation
Step 2: Dataset Split
Step 3: Supervised Training (Train Random Forest classifier on labeled flows)
Step 4: Unsupervised Anomaly Detection
Step 5: Model Evaluation

- Supervised: Accuracy, Precision, Recall, F1-score, Confusion Matrix.
- Unsupervised: ROC-AUC, Precision@K, flag percentage of anomalies.

Step 6: Model Deployment