PROJECT DOCUMENTATION - IOT TRAFFIC ANOMALY DETECTION

This document describes the full architecture, methodology, and implementation details for the project in this repository. It is a faithful reproduction of the pipeline in "IoT Dataset Validation Using Machine Learning Techniques for Traffic Anomaly Detection" (Vigoya et al., Electronics 2021, DOI:10.3390/electronics10222857) and extends it with a production-style web UI and optional Gemini assistant.

1) OBJECTIVE

* Reproduce the paper’s methodology end-to-end using proxy datasets from Hugging Face.
* Provide a clean ML package with modular preprocessing and model pipelines.
* Offer a web dashboard for dataset upload, model configuration, training, and evaluation.
* Add blue-team analysis: CVE similarity, defensive controls, and response playbooks.

2) REPOSITORY STRUCTURE

* iot_anomaly_detection/
  - data/: dataset loading, feature mapping, preprocessing
  - models/: model adapters, registry, nested CV trainer
  - utils/: constants, metrics, CV utilities
* iot-anomaly-ui/
  - backend/: FastAPI services
  - frontend/: React + MUI dashboard
* notebooks/: end-to-end reproduction notebook + generated charts
* config/: params.yaml for preprocessing and hyperparameter configuration
* docs/: project documentation and tutorial pages

3) PAPER METHODOLOGY MAPPING

The implementation mirrors the paper’s main steps.

3.1 PREPROCESSING

* Cyclical time encoding for frame.time (sin/cos of time-of-day).
* Numeric binning for frame.len and port fields.
* One-hot encoding for categorical features.
* Flow-based aggregation: flow.packets, flow.bytes, flow.duration, flow.rate.
* Text-derived features when categorical columns are highly cardinal (length, token count).

Core attributes (14 features) used as the canonical mapping:
* ip.src, ip.dst
* tcp.srcport, tcp.dstport
* udp.srcport, udp.dstport
* frame.len, frame.time
* tcp.flags, protocol
* label

3.2 CLASS IMBALANCE

* SMOTE is applied inside the inner loop of nested stratified CV.
* This prevents leakage across outer folds while balancing the training split.

3.3 FEATURE SELECTION

* Recursive Feature Elimination (RFE) with a Decision Tree estimator.
* RFE runs after preprocessing and SMOTE and before model fitting.

3.4 MODELS

Five shallow ML models (as in the paper):
* Logistic Regression
* Bernoulli Naive Bayes
* Random Forest
* AdaBoost
* Linear SVM

Optional extra models (for exploration):
* ExtraTrees
* GradientBoosting

3.5 HYPERPARAMETER TUNING

Grid search is performed inside the inner CV using ROC AUC.

* Logistic Regression: C, penalty, solver
* BernoulliNB: alpha, binarize
* Random Forest: n_estimators, max_features, max_depth
* AdaBoost: n_estimators, learning_rate
* Linear SVM: C, loss

3.6 EVALUATION

* Inner CV scoring: ROC AUC
* Outer CV reporting: Accuracy, Precision, Recall, F1

4) DATA SOURCES AND MAPPING

The original DAD dataset is not available on Hugging Face. We use proxy datasets:

* fenar/iot-security
* schooly/Cyber-Security-Breaches
* stu8king/securityincidents
* kutay1907/scadaphotodataset
* kutay1907/ScadaData100k
* vossmoos/vestasv52-scada-windturbine-granada

When a dataset does not contain a required field, the mapping layer simulates it with safe defaults. Labels are derived from domain heuristics (incident types, scenarios, text keywords) to enable end-to-end reproduction.

5) ML PACKAGE DETAILS

5.1 FEATURE MAPPING

* infer_feature_mapping() identifies likely source columns for canonical fields.
* apply_feature_mapping() renames columns and synthesizes missing features.
* label is coerced to binary; if missing, it is derived from domain cues.

5.2 PREPROCESSING

FeaturePreprocessor does:
* Build canonical features
* Add flow features
* Encode time cyclically
* Bin lengths and ports
* One-hot encode categorical columns
* Add text length/token features for high-cardinality columns

5.3 NESTED CV PIPELINE

The pipeline in each outer fold is:

1) Preprocess
2) Impute
3) SMOTE
4) RFE (DecisionTree)
5) Model

Grid search optimizes ROC AUC using inner CV. Outer fold metrics are aggregated with mean/std.

6) BACKEND ARCHITECTURE (FASTAPI)

Key services:

* dataset_loader.py: load HF datasets, upload local files
* preprocessor.py: load preprocessing config, build FeaturePreprocessor
* trainer.py: orchestrate nested CV and aggregate results
* analysis.py: CVE similarity and blue-team briefing generation
* llm_chat.py: Gemini assistant with dataset and leaderboard context

Key endpoints:

* /datasets/hf, /datasets/upload, /datasets/summary
* /train, /train/{job_id}, /train/history
* /analysis/cve-similarity, /analysis/briefing

* /assistant/chat

7) FRONTEND ARCHITECTURE (REACT + MUI)

Pages:

* Dashboard: overview of dataset and training status
* Dataset: upload and column mapping
* Training: select models and hyperparameters
* Results: leaderboard, confusion matrix, feature importance
* Threat Intel: CVE similarity and blue-team playbooks
* Assistant: Gemini-driven security analysis chat

8) THREAT INTEL AND CVE SIMILARITY

* TF-IDF similarity against the ahadda5/cve150k dataset.
* Returns top-k CVE descriptions and keyphrases related to the dataset.
* Intended for triage and hypothesis generation only.

9) GEMINI ASSISTANT (OPTIONAL)

* Uses google-genai SDK with a security-focused system prompt.
* Context includes dataset summary, leaderboard, and optional CVE matches.
* API key is provided via GEMINI_API_KEY environment variable.

10) NOTEBOOK REPRODUCTION

notebooks/experiment_reproduction.ipynb executes:
* dataset load
* preprocessing and feature mapping
* RFE sweep
* nested CV evaluation
* plots and leaderboard output

Generated figures are in notebooks/outputs/.

11) PERFORMANCE SNAPSHOT (SAMPLE_SIZE=2000)

The following table reflects the nested CV results stored in notebooks/outputs/leaderboard_multi.csv.

Dataset	Model	Accuracy	Precision	Recall	F1
fenar/iot-security	random_forest	1.0000	1.0000	1.0000	1.0000
fenar/iot-security	adaboost	1.0000	1.0000	1.0000	1.0000
fenar/iot-security	logistic_regression	0.1005	0.1005	1.0000	0.1826
fenar/iot-security	linear_svm	0.1005	0.1005	1.0000	0.1826
fenar/iot-security	naive_bayes	0.8995	0.0000	0.0000	0.0000
schooly/Cyber-Security-Breaches	random_forest	0.7147	0.5368	0.6979	0.5940

schooly/Cyber-Security-Breaches	linear_svm	0.5100	0.4100	0.7788	0.4144
schooly/Cyber-Security-Breaches	naive_bayes	0.5507	0.4156	0.6814	0.4126
schooly/Cyber-Security-Breaches	logistic_regression	0.5062	0.3727	0.7542	0.4070
schooly/Cyber-Security-Breaches	adaboost	0.6474	0.3760	0.4947	0.4032
vossmoos/vestasv52-scada-windturbine-granada	random_forest	1.0000	1.0000	1.0000	1.0000
vossmoos/vestasv52-scada-windturbine-granada	adaboost	1.0000	1.0000	1.0000	1.0000
vossmoos/vestasv52-scada-windturbine-granada	logistic_regression	0.9980	1.0000	0.9973	0.9986
vossmoos/vestasv52-scada-windturbine-granada	linear_svm	0.9945	1.0000	0.9926	0.9963
vossmoos/vestasv52-scada-windturbine-granada	naive_bayes	0.6980	0.8917	0.6752	0.7683

12) REPRODUCIBILITY AND CONFIGURATION

* config/params.yaml controls preprocessing bins and thresholds.
* Random seeds are fixed in model training and CV splits.
* Notebook is executable top-to-bottom.

13) LIMITATIONS

* Proxy datasets are not identical to DAD; labels and fields are mapped or derived.
* Some datasets are small and may lead to optimistic scores.
* CVE similarity is heuristic and not a vulnerability scanner.