Project Proposal: Reproducing NetML Baseline Results for Network Traffic Application Classification

Group: Connor McGraw, Faiz Hilaly, Hewitt Watkins

(edits with chat & claude)

Project Summary

This project aims to **reproduce the baseline classification results** from "NetML: A Challenge for Network Traffic Analytics" (Barut et al., 2020) for the application identification task. The original paper introduced three network traffic datasets and established baseline performance using Random Forest, SVM, and Multi-layer Perceptron models. However, like many machine learning papers, the reproducibility of these results depends on implementation details that may not be fully specified in the publication.

Why reproduction matters: Reproducibility is a cornerstone of scientific research, yet ML research often faces reproducibility challenges due to dependencies on specific tools, hyperparameters, random seeds, and data preprocessing steps. By attempting to reproduce the NetML baseline results, we can: (1) validate the published findings, (2) identify what information is sufficient (or missing) for reproduction, (3) understand practical challenges in network traffic analysis research, and (4) provide a foundation for future improvements to these models.

The original problem: The paper addresses application identification in network traffic flows—a critical task for network management, security monitoring, and quality of service. Traditional port-based methods fail with modern applications that use dynamic ports and encryption, making machine learning approaches necessary.

Data

We will use the **NetML VPN-nonVPN Application Identification Dataset**, specifically the **non-vpn2016 subset** used in Section 3.3 of the original paper for traffic classification tasks.

This dataset / focus reflects our interest in understanding how VPN usage and encryption might change network flow behavior. None of us has deep prior experience with VPN traffic, and the VPN–nonVPN setting seems relevant to current practice because it raises questions about balancing user privacy with operational needs like performance and visibility. By centering our

reproduction on this dataset, we aim to learn these dynamics while following the NetML setup as closely as possible.

Dataset specifications:

- **Size**: <1 GB (uncompressed)
- **Source**: Derived from ISCX VPN-nonVPN 2016 dataset
- **Protocols**: IPv4, TCP, UDP
- **Number of samples**: 163,711 flows total (131,065 in training set)
- **Three annotation levels**:
    - **Top-level (non-vpn-t)**: 7 classes (P2P, audio, chat, email, file_transfer, tor, video)
    - **Mid-level (non-vpn-m)**: 18 classes (specific applications: facebook, skype, hangouts, etc.)
    - **Fine-grained (non-vpn-f)**: 31 classes (detailed actions: facebook_audio, skype_chat, etc.)

Train/test split: Following the paper's methodology:

- 80% training set
- 10% test-std set (for evaluation)
- 10% test-challenge set
- The paper states samples were "randomly selected" for each split

Key reproducibility consideration: The original paper extracted features using an "Intel proprietary flow feature extraction tool." The dataset we're using is described as a "faithful recreation" from the nPrint paper (2021). We will need to verify whether the feature representations match the original NetML features or document any differences that may affect reproduction.

Machine Learning Approaches

We will reproduce the three baseline models exactly as specified in the paper:

1. Random Forest (RF)

- Number of estimators: 100
- Maximum depth: 10
- The paper found RF to be the best-performing baseline overall
- Uses 80% of training set for fitting, 20% for validation

2. Support Vector Machine (SVM)

- Kernel: RBF (Radial Basis Function)
- Regularization parameter C: 1.0 (default)

- **Computational constraint**: The paper notes SVM training is computationally expensive, so they used only 10% of the training set for fitting and 90% for validation
- We will follow this same sampling approach

3. Multi-layer Perceptron (MLP)

- Architecture: Single hidden layer with 121 units
- Regularization: L2 with alpha = 0.0001
- Optimizer: Adam
- Uses 80% of training set for fitting, 20% for validation

Feature set: Following the baseline methodology, we will use **only Metadata features** (31 features total). These protocol-independent statistical features include:

- Packet counts and byte counts (inbound/outbound)
- Flow duration (time_length)
- Port numbers
- Statistical features: mean, max, median, variance of header and payload lengths
- Histograms: compact histograms of packet intervals, header lengths, payload lengths
- TCP flags histogram

Preprocessing: The paper specifies that data is "standardized in the preprocessing step so that data in each feature column follows standard normal distribution" (Section 5.2).

Evaluation

Our evaluation will directly compare our reproduced results against the original paper's published metrics.

Primary metrics (as used in the original paper):

- **F1 Score**: Harmonic mean of precision and recall
- **mAP (mean Average Precision)**: Average of per-class average precision scores

Target results to reproduce (from Table 7 of the original paper):

Task	Metric	RF (Target)	SVM (Target)	MLP (Target)
non-vpn-t (7 classes)	F1	0.6273	0.5868	0.6066

	mAP	0.3257	0.1934	0.2304
non-vpn-m (18 classes)	F1	0.3693	0.3441	0.3609
	mAP	0.3223	0.1398	0.2041
non-vpn-f (31 classes)	F1	0.2486	0.2036	0.2359
	mAP	0.2127	0.0768	0.1404

Analysis plan:

1. **Quantitative comparison**: Report our F1 and mAP scores alongside the original results with absolute and relative differences
2. **Per-class performance**: Examine confusion matrices to verify which applications are most/least distinguishable, as shown in Figure 9 of the original paper
3. **Class imbalance verification**: The paper attributes poor performance to class imbalance (e.g., facebook_audio, hangouts_audio, skype_audio being oversampled). We will verify this finding
4. **Sensitivity analysis**: If results differ, investigate the impact of random seeds, library versions, and preprocessing variations
5. **Documentation**: Carefully document any discrepancies between our implementation and what was described in the paper

Learning Objectives

Through this reproduction study, we expect to learn:

1. Reproducibility assessment

- Can the published results be reproduced with the information provided in the paper?
- What is the variance in results across different random seeds or train/test splits?
- How sensitive are the results to implementation choices?

2. Implementation details matter

- Which details specified in the paper are critical for reproduction (e.g., data standardization, exact train/validation splits)?
- Which details are missing or ambiguous (e.g., random seed, exact feature extraction process)?
- How do different ML library versions (scikit-learn versions from 2020 vs. now) affect results?

3. Challenges in network traffic ML research

- The original paper used a proprietary Intel tool for feature extraction—how does this affect reproducibility?
- What barriers exist for researchers trying to build on this work?
- How do dataset access and preprocessing affect the ability to validate or extend published work?

4. Understanding the problem domain

- Why does performance degrade significantly as granularity increases (7 → 18 → 31 classes)?
- Which applications are inherently difficult to distinguish based on flow statistics alone?
- How does class imbalance affect model performance in this specific application?

5. Best practices for reproducible ML research

- What information should papers include to facilitate reproduction?
- How should datasets and code be released to enable validation?
- What documentation standards would improve reproducibility in this field?

Expected Challenges

We anticipate several challenges:

1. **Feature extraction**: The original paper's Intel proprietary tool may produce features different from our dataset's recreation
2. **Random seed effects**: The paper doesn't specify random seeds for train/test splitting or model initialization
3. **Library versions**: Scikit-learn and other libraries have evolved since 2020
4. **Computational environment**: Hardware differences may affect SVM training
5. **Ambiguous specifications**: Some preprocessing or hyperparameter details may be underspecified

We will document all these challenges and their resolutions, making our own reproduction study a resource for future researchers.

Success Criteria

We will consider this reproduction study successful if we:

1. Achieve F1 and mAP scores within ±10% of the original paper's results
2. Observe the same relative ranking of models (RF > MLP > SVM in most cases)
3. Confirm the performance degradation pattern as classes increase
4. Identify and document any discrepancies with clear hypotheses for their causes
5. Produce well-documented, runnable code that others can use to validate our reproduction

This project will provide valuable insights into the reproducibility of network traffic classification research and create a foundation for future improvements to these baseline models.

Deliverables

- **Hewitt Watkins:** Owns data acquisition and preprocessing (standardization, 80/10/10 splits), implements and trains the **Random Forest** baseline with ablations, and documents all data/model setup for reproducibility.
  Connor McGraw: Implements and trains the **SVM** (RBF, sampling constraints), profiles runtime/memory trade-offs, and co-builds the shared evaluation pipeline (F1/mAP tables, confusion matrices).
- **Faiz Hilaly:** Implements and trains the **MLP** (121 units, Adam, L2), leads sensitivity studies (seeds, scaler/library versions) across **all three models**, and produces the analysis figures and results/discussion text.
- **All members:** Collaborate to finalize the **Colab notebook**, refine code readability and documentation, and co-write the **final project report and presentation**.