Optimizing Feature Selection for Cross-Mission Exoplanet Detection: A CatBoost and Optuna Approach for Generalizable Vetting

Abstract

The abstract summarizes the paper's motivation, methods, and findings. It explains that NASA's Kepler and TESS missions have produced a surge in exoplanet candidates, requiring automated vetting to separate true transits (dips in star brightness caused by planets) from false positives (e.g., noise or stellar activity).

- **Problem Highlighted**: Models trained on Kepler data using mission-specific features (e.g., flags like koi_fpflag_nt) achieve ~99% accuracy but fail in transfer learning to other missions because those features aren't universal.
- **Approach**: You developed a generalized ML model using only 11 physical and stellar features (identified via literature review and experiments). These features are extractable from light flux data in any transit-based mission.
- **Key Results**: Among tested models (e.g., CatBoost, XGBoost, Random Forest, LightGBM), CatBoost performed best with 83.9% test accuracy after hyperparameter tuning with Optuna.
- **Implications**: The model prioritizes scalability and generalizability, enabling validation of candidates from TESS and future missions without overfitting to Kepler-specific data.

3. Introduction

This section sets the stage by explaining exoplanets, missions, detection methods, and feature extraction.

3.1 Missions Introduction

- **Exoplanets**: Planets orbiting stars outside our solar system.
- **Kepler Mission**: A NASA space telescope (2009–2018) focused on specific sky regions to detect Earth-to-Neptune-sized planets. It monitored star brightness over 9+ years, identifying dips called Threshold Crossing Events (TCEs).
- **TESS (Transiting Exoplanet Survey Satellite)**: Launched in 2018, it surveys the entire sky in sectors, targeting bright, nearby stars for short periods. Like Kepler, it produces light flux data (brightness over time) to detect TCEs.
- **Key Point**: Both missions provide raw brightness data, not direct planet confirmations—vetting is needed to classify TCEs as candidates or false positives.

3.2 Exoplanet Detection Methods

- **Transit Method** (Primary Focus): Detects planets by measuring periodic dips in starlight when a planet passes in front of its star. It has discovered 4,478 exoplanets (as of the paper's date). Provides info on planet size (from dip depth) and orbital period (from dip timing). Dips can be false positives from noise, stars, or other factors.
- **Radial Velocity Method**: Measures a star's "wobble" due to a planet's gravity. Detects 1,158 exoplanets; gives mass info (complements transit's size data).
- **Explanation**: Transit is like watching a bug fly across a lightbulb—the shadow tells you about the bug's size and speed. Radial velocity is like feeling the lightbulb shake from the bug's pull.

3.3 Feature Extraction from Light Flux Data

- Raw data from missions is light flux (brightness curves). Tools like Python's Lightkurve library extract features:
    - **Transit Depth**: How much light is blocked (indicates planet radius relative to star).
    - **Transit Duration**: Length of the dip (indicates orbital speed and distance).
    - **Orbital Period**: Time between dips.
- These, combined with stellar parameters (e.g., star temperature), form datasets for ML models to automate detection.

4. Problem Statement

Manual vetting of exoplanet data is time-consuming, error-prone, and inefficient for the thousands of candidates from Kepler (140+ features per candidate) and TESS. ML models can help, but:

- Feeding all features causes "curse of dimensionality" (too many variables make learning hard), noise, and overfitting.
- Mission-specific features (e.g., Kepler flags) yield high accuracy (~99%) but fail on TESS data.
- Need: A minimal, universal feature set for high-accuracy models that enable transfer learning across missions.

In essence: Models "cheat" by relying on Kepler-only hints, but real-world use requires a model that works on any telescope's data.

5. Research Objectives

The study aims to create a transferable ML framework for exoplanet validation:

- Identify a minimal set of features for high-accuracy detection.
- Ensure generalization for TESS and future missions (PLATO, Earth 2.0).
- Experiment with feature subsets from a superset of important ones.

- Test ML models (CatBoost, XGBoost, Random Forest, LightGBM) and compare accuracies.
- Use Optuna for hyperparameter tuning (better than GridSearch or RandomCV).
- Analyze feature importance for further refinements.

6. Research Questions

- Can minimal physical features achieve accuracy rivaling human vetting across missions?
- How does performance differ with mission-specific vs. cross-mission features?
- Is stellar metallicity (host star's metal content) as crucial as in Huang et al. (2024)? Will it make the final feature set?
- Does Optuna-tuned CatBoost outperform Random Forest (commonly used in prior studies)?

7. Literature Review

This section traces the evolution from manual to automated vetting:

1. **Robovetter**: First automated system using decision trees to mimic human judgment.
2. **AstroNet (Yu et al., 2019)**: CNN on raw Kepler light curves; 98.8% accuracy, but less precise on early TESS data.
3. **Random Forest (Sturrock et al., 2019)**: 98% on Kepler, but relied on mission-specific features, ignoring stellar params.
4. **Huang et al. (2024)**: Combined transit/stellar features; 83.9% accuracy on Kepler with Random Forest in cross-mission context (4 ML models + 1 neural net).
- Gap Filled by Your Work: Focus on minimal, generalizable features for transfer learning.

8. Methodology

8.1 Data

- Source: Kepler KOI (Kepler Objects of Interest) cumulative table from NASA Exoplanet Archive (9,564 rows, 140 features).
- Reduced to 11 key features via literature and experiments (physical/transit/stellar, not mission-specific).
- Target: koi_pdisposition (CANDIDATE = 1, FALSE POSITIVE = 0). Balanced dataset (4,847 candidates, 4,717 false positives; see Figure 1: Bar and pie chart showing near-equal distribution).

8.2 Tools and Technologies

8.2.1 Data Preprocessing

- Missing Values: Imputed with KNNImputer (n_neighbors=5) for key columns; some experiments dropped NaNs to compare accuracy.
- Encoding: Binary labels (1 for candidates, 0 for false positives).
- Split: 80% train, 20% test (pseudo-random with fixed seed for reproducibility).
- Scaling: StandardScaler to normalize features for better model convergence.

8.3 Machine Learning Models

- Started with a superset of 21 features (vetting flags, transit properties, positional, physical/stellar). Detailed explanations for each (e.g., koi_fpflag_nt: Flags impossible planet signals like noise; importance: Reduces false positives).
- Initial Random Forest (with flags) achieved 99.16% accuracy (Table 1: Accuracy 99.16%, Recall/Precision 99.15%) but overfit due to "cheat sheet" flags (Figure 2: Flags had highest importance).
- Issue: "Picture Perfect Accuracy Trap"—High accuracy but no generalization; flags unavailable in TESS.

8.4 Refining Model for Transfer Learning

- Excluded mission-specific flags (koi_fpflag_*).
- Experimented with subsets and models: Random Forest, XGBoost, CatBoost, LightGBM, and ensemble stacking.
- Refined Features: Transit (period, duration, depth, impact, SNR); Stellar (temperature, gravity, metallicity).

8.5 Multi-Model Optimization and Hyperparameter Tuning

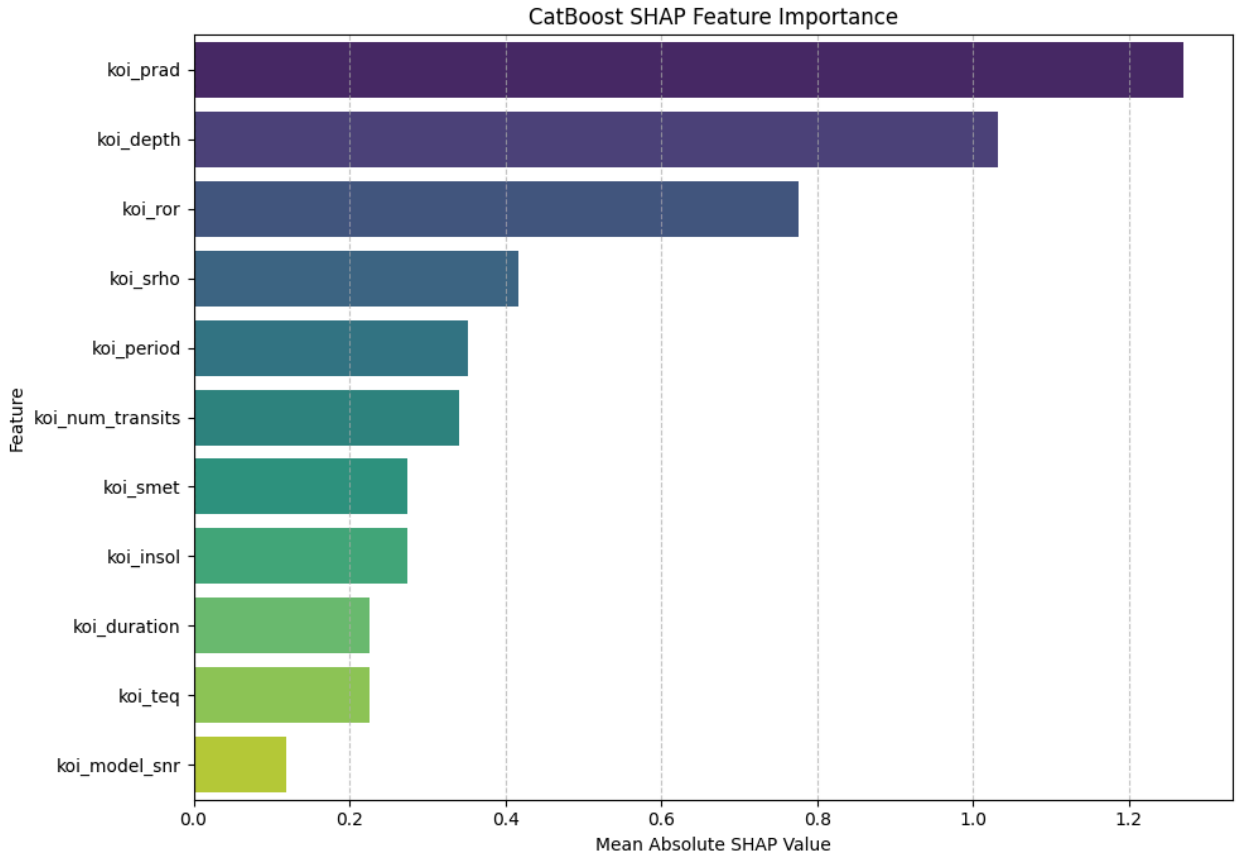
- Used Optuna (Bayesian optimization) for tuning.
- Best Hyperparams (detailed in paper for each model, e.g., CatBoost: iterations=400, learning_rate≈0.042, etc.).
- Ensemble Weights (tuned via Optuna): CatBoost 0.4174, XGBoost 0.0739, LightGBM 0.5086.

8.6 Model Performance Metrics and Results

- Table 2: CatBoost topped with 0.8390 accuracy, 0.8342 recall, 0.8378 precision, 0.8360 F1.
- Feature Importance Analyzed: Across models (Figures 3–6: Bar charts showing rankings, e.g., koi_prad often high).

8.7 Final Features Subset Result

- Final 11 Features: koi_period, koi_duration, koi_depth, koi_ror (radius ratio), koi_srno (likely a typo for SNR), koi_prad (planetary radius), koi_teq (equilibrium temperature), koi_insol (insolation flux), koi_smet (metallicity), koi_model_snr, koi_num_transits.
- SHAP Importance (Figure 7: Bar chart; e.g., koi_prad highest).
- Highest Accuracy: 83.9% with standalone CatBoost—balances performance and generalizability.



9. Conclusion

- Developed a framework using 11 generalizable features from Kepler data.
- CatBoost + Optuna outperformed others (83.9% accuracy), avoiding overfitting.
- Addresses limitations of prior models; enables cross-mission vetting.
- Limitations: Binary classification, Kepler biases, no TESS real-time testing.
- Future Work: Validate on TESS, integrate deep learning (e.g., Lightkurve for raw curves), multi-class, habitable zone focus, real-time pipelines.