Fraud Transaction Detection: Project Report

Nature Inspired Computing

Nikita Zagainov, Dmitry Tetkin, Alisher Kamolov, Nikita Tsukanov

Innopolis University

April 2025

Understanding the Fraud Detection Domain

- Digital and contactless payments are on the rise, increasing the risk of fraudulent transactions.
- Fraud not only results in financial losses, but also reduces customer trust.
- Datasets in this domain tend to be large and anonymized, posing unique challenges in feature interpretation.
- Successful fraud detection requires managing imbalanced data and high-dimensional feature spaces.

Instruments and Methodologies

- **Machine Learning Models:**
  - Gradient boosting frameworks like CatBoost and LightGBM.
  - Decision Trees for baseline performance and interpretability.
- **Optimization Techniques:**
  - Nature-inspired algorithms (e.g., Bat Algorithm, Cuckoo Search, Grey Wolf Optimizer) for optimal feature selection.
  - Utilization of libraries (like NiaPy) to implement these algorithms.
- **Data Preprocessing:**
  - Handling missing data, encoding categorical variables, and addressing imbalances.
  - Dimensionality reduction to mitigate the curse of dimensionality.
- **Evaluation Metrics:**
  - ROC AUC is adopted as the main performance measure.

Reference Work

Our project was inspired by the following project: `https://github.com/pmacinec/transactions-fraud-detection` Our project's contribution is the following:

- Comparison with gradient boosting algorithms
- Application of NIC algorithms to the gradient boosting algorithms

Project Pipeline Overview

Our work follows a systematic pipeline:

1. **Data Preprocessing:**
   - Clean data by filling or removing missing values.
   - Encode categorical variables and standardize numerical features.
   - Reduce dimensionality while retaining key information.
2. **Feature Selection:**
   - Apply nature-inspired algorithms (NIC) for selecting optimal feature subsets.
   - Compare NIC-selected features against traditional methods.
3. **Model Training and Evaluation:**
   - Train robust ML models (CatBoost, LightGBM, Decision Trees) on the refined dataset.
   - Evaluate model performance using ROC AUC as the primary metric.
4. **Optimization and Analysis:**
   - Fine-tune models and re-run experiments to analyze the impact of feature selection.
   - Compare NIC algorithms to determine best performing configurations.

Results

Table: ROC AUC of ML models with NIC feature selection

Method	CatBoost	LightGBM	DecisionTree
Original Score	**0.893**	0.909	0.835
Artificial Bee Colony	0.887	0.906	0.836
Cuckoo Search	0.891	0.905	0.842
Bat Algorithm	0.889	0.909	0.842
Firefly Algorithm	0.892	0.907	0.844
Flower Pollination	0.891	0.906	**0.846**
Grey Wolf Optimizer	0.892	**0.913**	0.844
Particle Swarm	0.892	0.912	0.845
Algorithms better than baseline	0	2	7

Conclusion

- NIC algorithms for feature selection do not improve performance of gradient boosting models.
- We hypothesize that the reason for such difference in results between a single tree and ensemble methods is that single trees are more prone to overfitting.

Thank you for your attention!