# DECOYNET

## Decoy Data Injection for Threat Reduction

Aashana Baldi, Keerthana K S, Proshita Agarwal

# The Problem

Problem

- Databases are prime targets for data theft
- Traditional security fails after attacker gains access
- Stolen data is difficult to trace or detect

Our Idea

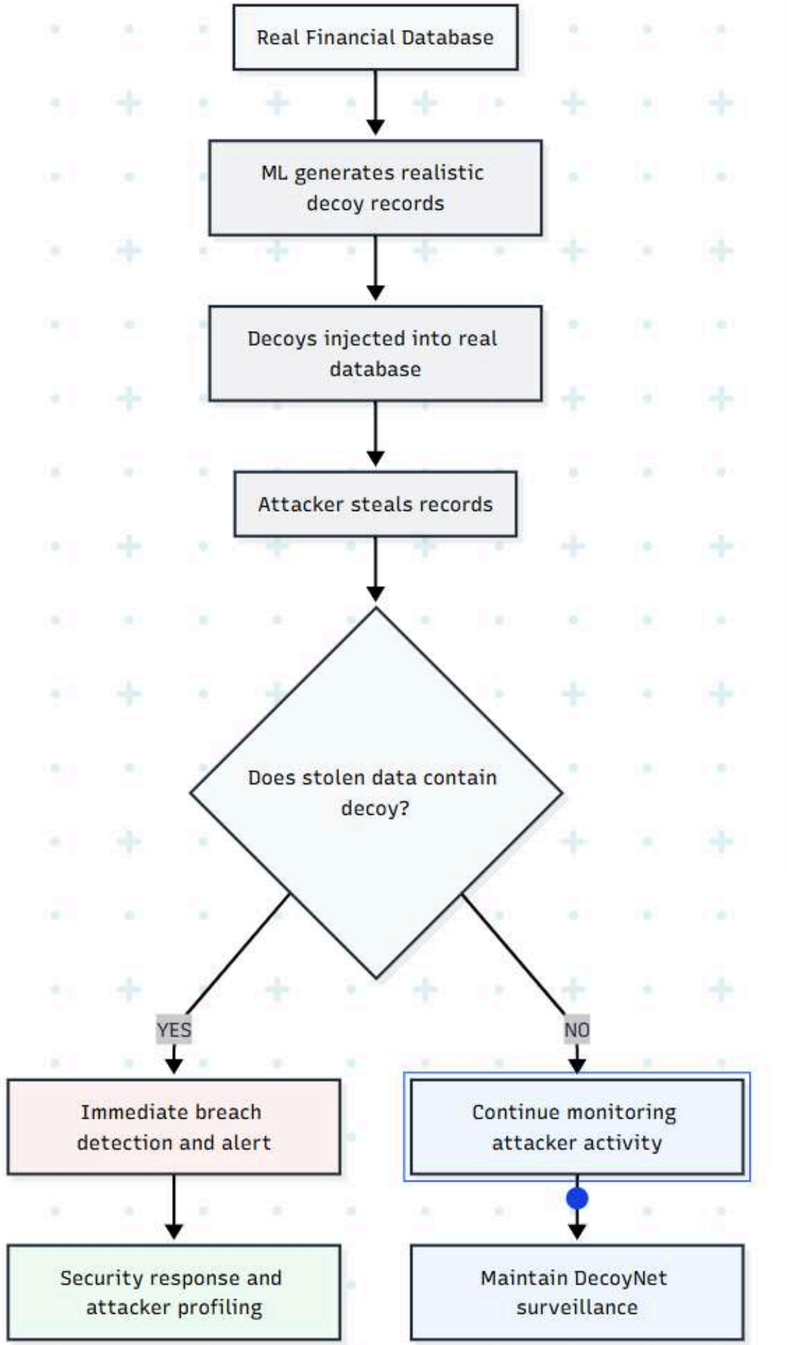
- Inject realistic fake records into database
- ML makes decoys statistically realistic
- Stolen decoy → immediate breach detection

Applications

Finance • Healthcare • E-commerce • Enterprise Security

# How DecoyNet Works

- Learns patterns of legitimate financial transactions
- Generates statistically realistic fake records (decoys)
- Strategically injects decoys into real databases
- Any attacker accessing a decoy immediately triggers detection
- Enables early breach discovery with minimal impact on real data



# Literature Review - I

## ADVANCED NETWORK SECURITY SYSTEM: HONEYPOT-BASED INTRUSION DETECTION WITH MACHINE LEARNING AND VISUALIZATION

The work proposes a network security system that combines honeypots, intrusion detection systems, and machine learning to detect cyber attacks. Honeypots collect attacker data, which is analyzed by ML models to identify both known and unknown threats.

Limitations:

- Limited focus on adaptive deception strategies.
- Honeypots may be fingerprinted by advanced attackers.
- Deception environments are mostly static.
- Weak integration between deception and adaptive ML responses.

Proceedings of the 13th International Conference on Applied Innovations in IT (ICAIIT), July 2025

**Advanced Network Security System: Honeypot-Based Intrusion Detection with Machine Learning and Visualization**

Ashwini Lakshmipathy[1], Muthupandi Gurusamy[1], Siti Fatmawati Lalo[2], Nonia Sakka Lebang[3], Karthikeyan Selvaraj[4] and Aws A. Abdulsahib[5]

[1]*Department of Computer Science and Engineering, Chennai Institute of Technology, 600069 Chennai, Tamil Nadu, India*

[2] *Department of Master of Laws Study Program, Universitas Sulawesi Tenggara, 93121 Kendari, Indonesia*

[3] *Department of Government Science Study Program, Universitas Sulawesi Tenggara, 93121 Kendari, Indonesia*

[4]*Center for Advanced Multidisciplinary Research and Innovation, Chennai Institute of Technology, 600069 Chennai, Tamil Nadu, India.*

[5]*Department of Computer Science, Dijlah University College, 10021 Baghdad, Iraq*

*ashwinil.cse2022@citchennai.net, pg.muthupandi@gmail.com, stfatmawati@un-sultra.ac.id, noniasakkalebang@un-sultra.ac.id, karthikeyans.mech@citchennai.net, aws.abdulkareem@duc.edu.iq*

Keywords: Honeypots, Intrusion Detection Systems (IDS), Machine Learning in Cybersecurity, Network Security Visualization, Anomaly Detection.

Abstract: This paper focuses on developing a proactive approach to network intrusion detection through integration of honeypots with machine learning for improved security in complex network system. The system utilizes honeypots to capture attackers whereby the honeypots capture real-time traffic details which the system maps and analyzes packet content related to protocols. Blending with machine learning, the detection model analyzes the accurate data to detect known as well as unknown forms of cyber threats. There is a new feature called Visualization Dashboard that gives analytics and reports to network administrators. It provides information about honeypot engagements, traffic, and intrusion detected empowering the monitoring and management process. Incorporating the proactive defense measures into the proposed system eliminates the weakness of the conventional intrusion detection approach in managing new forms of cyber threats. The honeypots are designed to contain the attackers and simultaneously acquiring useful information about the intrusive activities. The effectiveness is enhanced by the ability of the Machine Learning model in enhancing the detection rates besides the flexibility in accommodating new techniques of detection of attacks. The Visualization Dashboard improves usability since it contains an easily navigable interface for current security monitoring and past performance examination. This approach guarantees the entirety of network protection by integrating the effectiveness of the deception-based honeypot systems and the machine learning approach based on big data. The paper reveals that the system is capable of enhancing detection rates, reducing false positives and providing valuable information regarding the network status to

# Literature Review - II

## ACTIVE DECEPTION USING FACTORED INTERACTIVE POMDPS TO RECOGNIZE CYBER ATTACKER'S INTENT

This work proposes an active cyber deception framework that uses factored Interactive POMDPs (I-POMDPX) to recognize an attacker's intent. The system strategically deploys decoys and deceptive actions during different phases of an attack to engage the attacker and infer their goals. By modeling the attacker's beliefs and behavior, the system improves intent recognition compared to traditional passive honeypot strategies.

Limitations:

- Computational complexity due to the I-POMDP decision framework.
- Experiments are limited to a single host honeypot environment.
- Relies on accurate log analysis, which may produce noisy observations.
- Scalability to large real-world networks is not fully demonstrated.

**Active Deception using Factored Interactive POMDPs to Recognize Cyber Attacker's Intent**

**Aditya Shinde**
Institute for AI
University of Georgia, Athens
GA 30602
adityas@uga.edu

**Prashant Doshi**
Institute for AI & Dept. of Computer Science
University of Georgia, Athens
GA 30602
pdoshi@uga.edu

**Omid Setayeshfar**
Dept. of Computer Science
University of Georgia, Athens
GA 30602
omid.s@uga.edu

**Abstract**

This paper presents an intelligent and adaptive agent that employs deception to recognize a cyber adversary's intent. Unlike previous approaches to cyber deception, which mainly focus on delaying or confusing the attackers, we focus on engaging with them to learn their intent. We model cyber deception as a sequential decision-making problem in a two-agent context. We introduce factored finitely-nested interactive POMDPs (I-POMDP$_\mathcal{X}$) and use this framework to model the problem with multiple attacker types. Our approach models cyber attacks on a single honeypot host across multiple phases from the attacker's initial entry to reaching its adversarial objective. The defending I-POMDP$_\mathcal{X}$-based agent uses decoys to engage with the attacker at multiple phases to form increasingly accurate predictions of the attacker's behavior and intent. The use of I-POMDPs also enables us to model the adversary's mental state and investigate how deception affects their beliefs. Our experiments in both simulation and on a real host show that the I-POMDP$_\mathcal{X}$-based agent performs significantly better at intent recognition than commonly used deception strategies on honeypots.

**1 Introduction**

An important augmentation of conventional cyber defense utilizes deception-based cyber defense strategies [16]. These are typically based on the use of decoy systems called *honeypots* [21] with additional monitoring capabilities. Currently, honeypots tend to be passive systems with the purpose of consuming the attacker's CPU cycles and time, and possibly logging the attacker's actions. However, the information inferred about the attackers' precise intent and capability is usually minimal.

On the other hand, honeypots equipped with fine-grained logging abilities offer an opportunity to better understand attackers' intent and capabilities. We may achieve this by engaging and manipulating the attacker to perform actions that reveal his or her true intent. One way of accomplishing this is to employ active deception. Active strategies entail adaptive deception which seeks to influence the

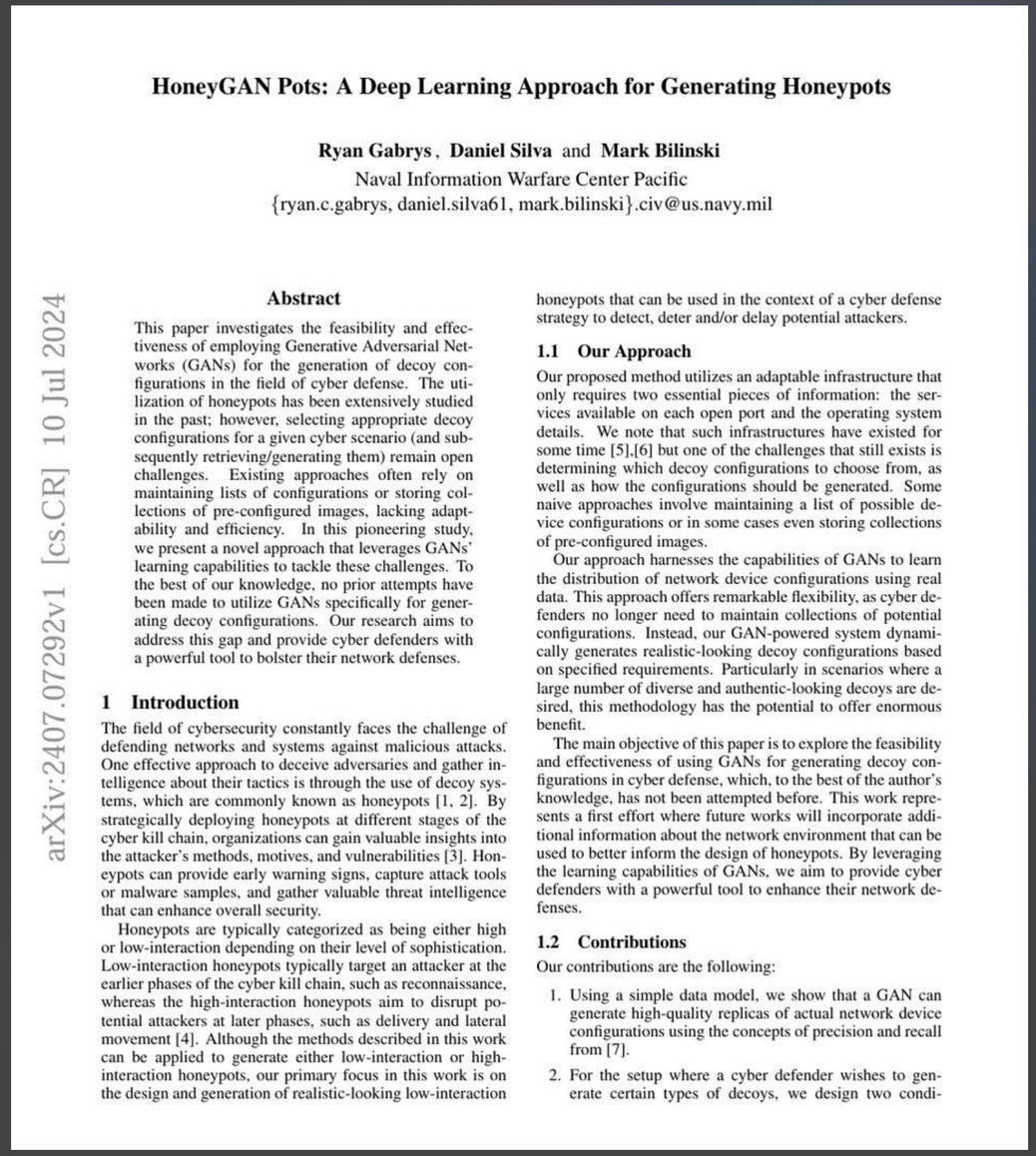
# Literature Review - III

## HONEYGANPOTS: A DEEP LEARNING APPROACH FOR GENERATING HONEYPOTS

This work proposes HoneyGAN Pots, a deep learning approach that uses Generative Adversarial Networks (GANs) to automatically generate realistic honeypot configurations. The model learns patterns from real network device configurations and produces diverse decoy systems, allowing defenders to deploy scalable and realistic honeypots without manually maintaining configuration libraries.

Limitations:

- Uses a relatively sparse data representation of device configurations.
- Training data contained duplicate samples, which may cause overfitting.
- Generated decoys may have reduced diversity as sample size increases.
- Limited contextual awareness of the surrounding network environment.



**HoneyGAN Pots: A Deep Learning Approach for Generating Honeypots**

**Ryan Gabrys**, **Daniel Silva** and **Mark Bilinski**
Naval Information Warfare Center Pacific
{ryan.c.gabrys, daniel.silva61, mark.bilinski}.civ@us.navy.mil



**Abstract**

This paper investigates the feasibility and effectiveness of employing Generative Adversarial Networks (GANs) for the generation of decoy configurations in the field of cyber defense. The utilization of honeypots has been extensively studied in the past; however, selecting appropriate decoy configurations for a given cyber scenario (and subsequently retrieving/generating them) remain open challenges. Existing approaches often rely on maintaining lists of configurations or storing collections of pre-configured images, lacking adaptability and efficiency. In this pioneering study, we present a novel approach that leverages GANs' learning capabilities to tackle these challenges. To the best of our knowledge, no prior attempts have been made to utilize GANs specifically for generating decoy configurations. Our research aims to address this gap and provide cyber defenders with a powerful tool to bolster their network defenses.

**1 Introduction**

The field of cybersecurity constantly faces the challenge of defending networks and systems against malicious attacks. One effective approach to deceive adversaries and gather intelligence about their tactics is through the use of decoy systems, which are commonly known as honeypots [1, 2]. By strategically deploying honeypots at different stages of the cyber kill chain, organizations can gain valuable insights into the attacker's methods, motives, and vulnerabilities [3]. Honeypots can provide early warning signs, capture attack tools or malware samples, and gather valuable threat intelligence that can enhance overall security.

Honeypots are typically categorized as being either high or low-interaction depending on their level of sophistication. Low-interaction honeypots typically target an attacker at the earlier phases of the cyber kill chain, such as reconnaissance, whereas the high-interaction honeypots aim to disrupt potential attackers at later phases, such as delivery and lateral movement [4]. Although the methods described in this work can be applied to generate either low-interaction or high-interaction honeypots, our primary focus in this work is on the design and generation of realistic-looking low-interaction honeypots that can be used in the context of a cyber defense strategy to detect, deter and/or delay potential attackers.

**1.1 Our Approach**

Our proposed method utilizes an adaptable infrastructure that only requires two essential pieces of information: the services available on each open port and the operating system details. We note that such infrastructures have existed for some time [5],[6] but one of the challenges that still exists is determining which decoy configurations to choose from, as well as how the configurations should be generated. Some naive approaches involve maintaining a list of possible device configurations or in some cases even storing collections of pre-configured images.

Our approach harnesses the capabilities of GANs to learn the distribution of network device configurations using real data. This approach offers remarkable flexibility, as cyber defenders no longer need to maintain collections of potential configurations. Instead, our GAN-powered system dynamically generates realistic-looking decoy configurations based on specified requirements. Particularly in scenarios where a large number of diverse and authentic-looking decoys are desired, this methodology has the potential to offer enormous benefit.

The main objective of this paper is to explore the feasibility and effectiveness of using GANs for generating decoy configurations in cyber defense, which, to the best of the author's knowledge, has not been attempted before. This work represents a first effort where future works will incorporate additional information about the network environment that can be used to better inform the design of honeypots. By leveraging the learning capabilities of GANs, we aim to provide cyber defenders with a powerful tool to enhance their network defenses.

**1.2 Contributions**

Our contributions are the following:

1. Using a simple data model, we show that a GAN can generate high-quality replicas of actual network device configurations using the concepts of precision and recall from [7].
2. For the setup where a cyber defender wishes to generate certain types of decoys, we design two condi-

# Literature Review Summary

| Aspect | Paper 1 | Paper 2 | Paper 3 | Our Model |
|---|---|---|---|---|
| Deception Type | Stayic honeypots | Active, phasse- based decoys | Auto generated configs (offline) | Dynamic, real time adaptive |
| ML/ AI used | Threat classification | Intent modelling via POMDP | GAN for configs generation | Real time classification + behavioral analysis |
| Adaptability | None | Partial (per attack phase) | None (offline only) | Full time adaptation |
| Fingerprinting Risk | High | Medium | Medium | Low (environment mutates) |
| Scalability | Moderate | Poor (single host) | Moderate | Multi node ready |
| Drawback addressed | Tight ML deception feedback loop | Lighter weight than POMDP | Runtime diversity maintenance | |

# Data Overview and Motivation

## Dataset: PaySim Synthetic Financial Transaction Dataset

- Source: Edgar Lopez-Rojas et al. (2016)
- Original dataset: 6.3M rows
  10% sampled subset used for experimentation
- Type: Synthetic dataset generated from real transaction behavior patterns

## Key Features:

- Transaction type
- Transaction amount
- Sender/receiver balances
- Origin & destination account IDs
- Fraud labels

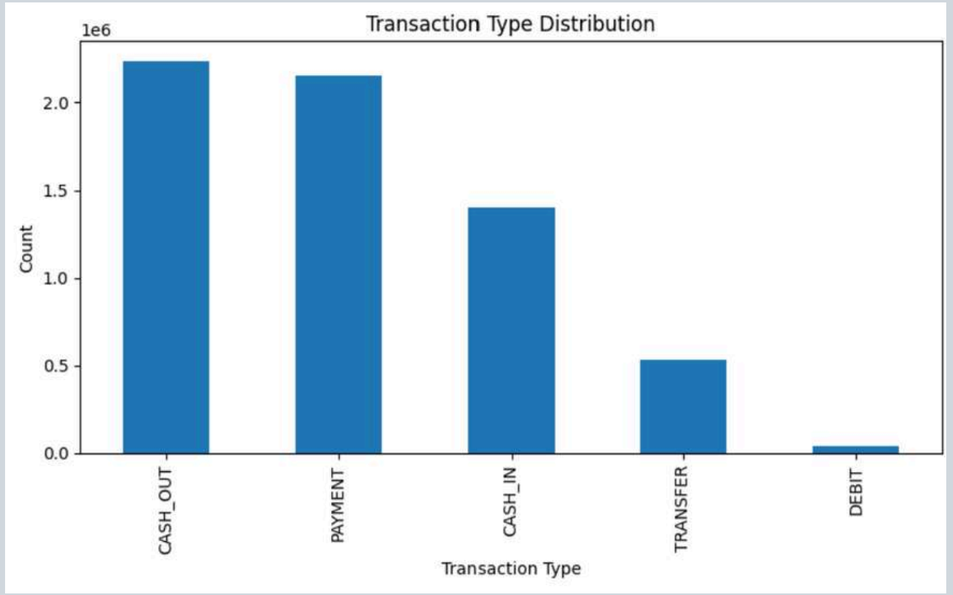
| | A | B | C | D | E | F | G | H | I | J | K |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | step | type | amount | nameOrig | oldbalanceOrg | newbalanceOr | nameDest | oldbalanceDest | newbalanceDest | isFraud | isFlaggedFraud |
| 2 | 1 | PAYMENT | 9839.64 | C1231006815 | 170136 | 160296.36 | M1979787155 | 0 | 0 | 0 | 0 |
| 3 | 1 | PAYMENT | 1864.28 | C1666544295 | 21249 | 19384.72 | M2044282225 | 0 | 0 | 0 | 0 |
| 4 | 1 | TRANSFER | 181 | C1305486145 | 181 | 0 | C553264065 | 0 | 0 | 1 | 0 |
| 5 | 1 | CASH_OUT | 181 | C840083671 | 181 | 0 | C38997010 | 21182 | 0 | 1 | 0 |
| 6 | 1 | PAYMENT | 11668.14 | C2048537720 | 41554 | 29885.86 | M1230701703 | 0 | 0 | 0 | 0 |
| 7 | 1 | PAYMENT | 7817.71 | C90045638 | 53860 | 46042.29 | M573487274 | 0 | 0 | 0 | 0 |
| 8 | 1 | PAYMENT | 7107.77 | C154988899 | 183195 | 176087.23 | M408069119 | 0 | 0 | 0 | 0 |
| 9 | 1 | PAYMENT | 7861.64 | C1912850431 | 176087.23 | 168225.59 | M633326333 | 0 | 0 | 0 | 0 |
| 10 | 1 | PAYMENT | 4024.36 | C1265012928 | 2671 | 0 | M1176932104 | 0 | 0 | 0 | 0 |
| 11 | 1 | DEBIT | 5337.77 | C712410124 | 41720 | 36382.23 | C195600860 | 41898 | 40348.79 | 0 | 0 |
| 12 | 1 | DEBIT | 9644.94 | C1900366749 | 4465 | 0 | C997608398 | 10845 | 157982.12 | 0 | 0 |
| 13 | 1 | PAYMENT | 3099.97 | C249177573 | 20771 | 17671.03 | M2096539129 | 0 | 0 | 0 | 0 |
| 14 | 1 | PAYMENT | 2560.74 | C1648232591 | 5070 | 2509.26 | M972865270 | 0 | 0 | 0 | 0 |
| 15 | 1 | PAYMENT | 11633.76 | C1716932897 | 10127 | 0 | M801569151 | 0 | 0 | 0 | 0 |
| 16 | 1 | PAYMENT | 4098.78 | C1026483832 | 503264 | 499165.22 | M1635378213 | 0 | 0 | 0 | 0 |
| 17 | 1 | CASH_OUT | 229133.94 | C905080434 | 15325 | 0 | C476402209 | 5083 | 51513.44 | 0 | 0 |

## Why PaySim?

- Large-scale dataset
- Fully interpretable features
- Privacy-preserving (no real banking data)
- Suitable for ML-based fraud & security experiments

## Why Used in DecoyNet

- Realistic transaction distributions
- Ideal for simulating data exfiltration attacks
- Supports generation and evaluation of decoy records



# Data Preprocessing

## Dataset & Sampling

Load PaySim (636,262 transactions). Randomly sample 10% for efficient pipeline execution while preserving class distribution.

## Feature Engineering & Encoding

Drop identifier/leaky columns. Engineer ratio features (amount-to-balance). One-hot encode 5 transaction types → 13 total features.

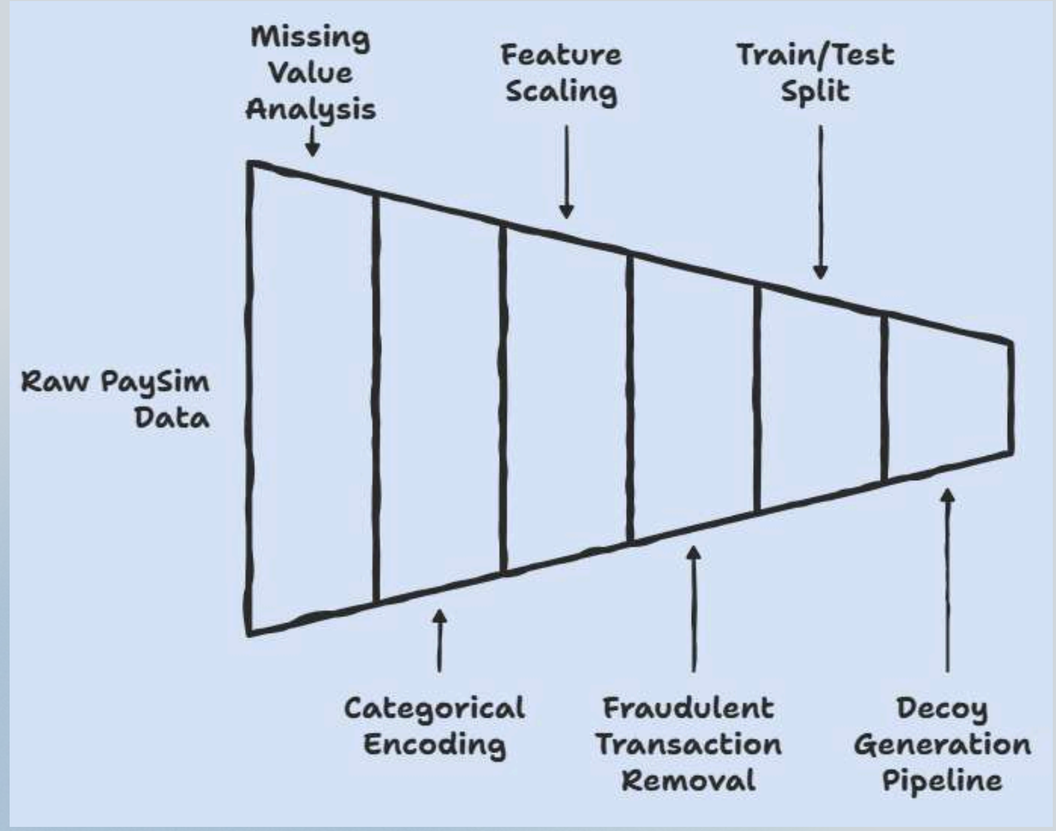
## Stratified Train/Val/Test Split

Split 70/10/20 with stratification on fraud label. Fraud rate preserved across all splits (Train: 0.128%, Val: 0.129%, Test: 0.128%)

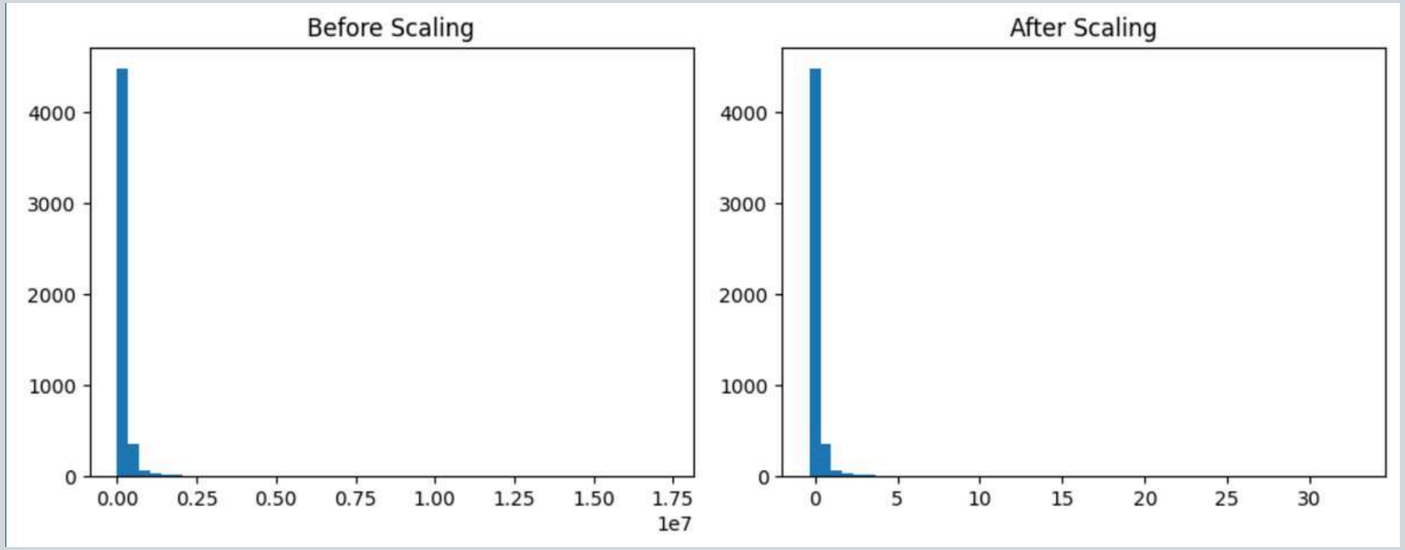
## Feature Scaling

Fit StandardScaler on training set only → prevent data leakage. Transform val and test sets using train statistics. Scaler saved for downstream layers.

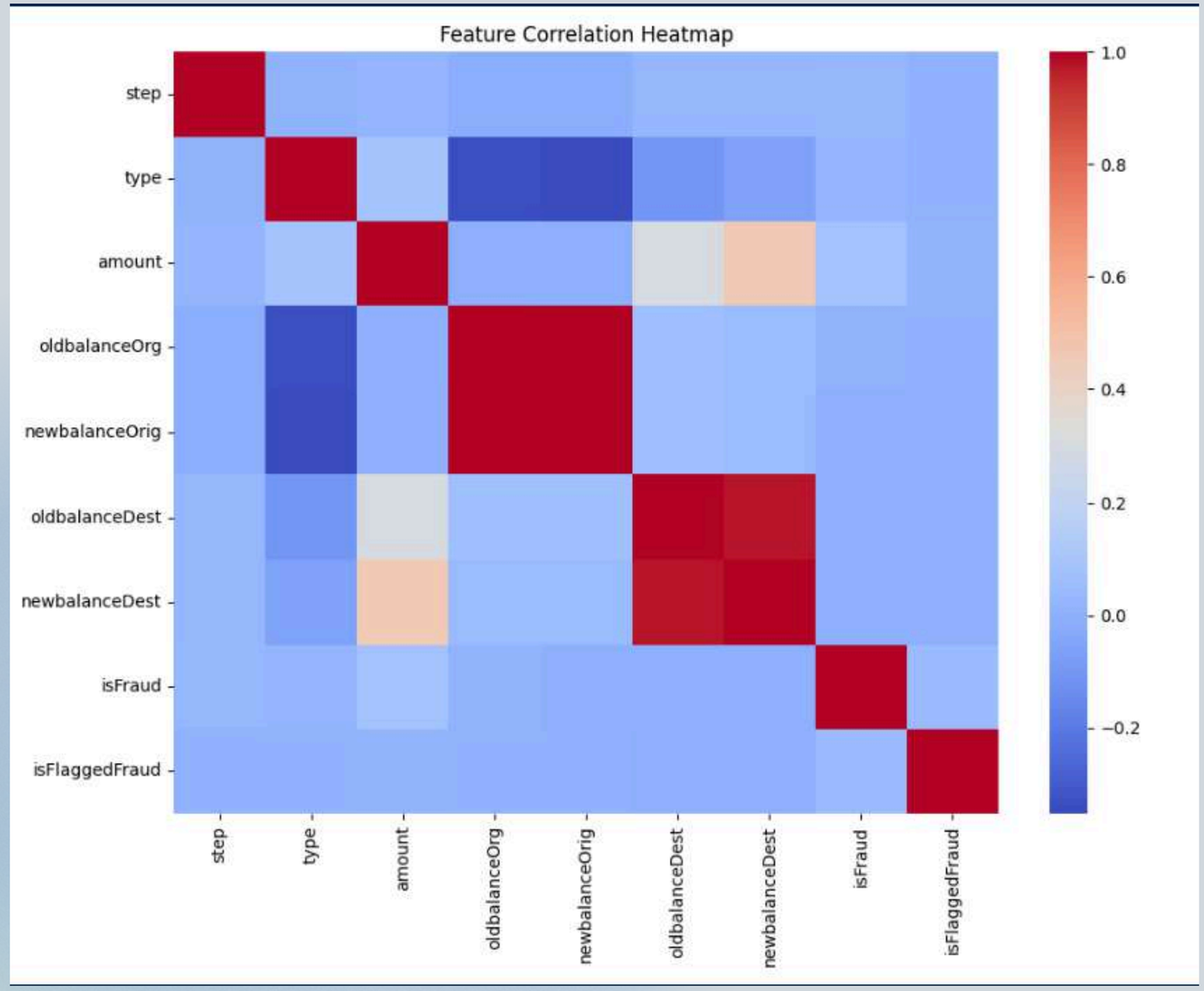
# Feature Preprocessing Pipeline




- No significant missing values observed
- Transaction types numerically encoded
- Standard scaling applied for stable latent learning
- Only legitimate transactions used for decoy synthesis
- Fraud labels retained for downstream evaluation
- PCA used in fallback decoy-generation pipeline

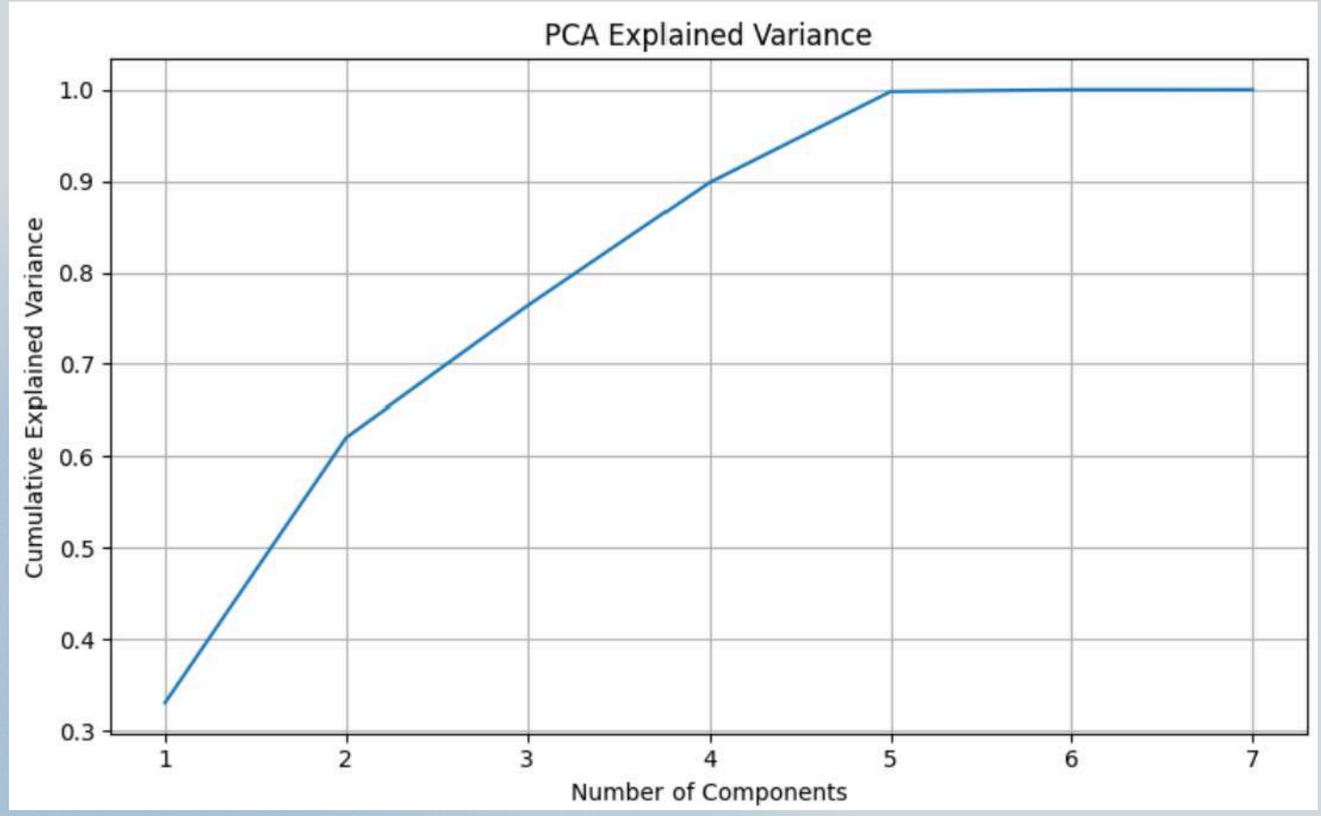


# FEATURE RELATIONSHIP ANALYSIS



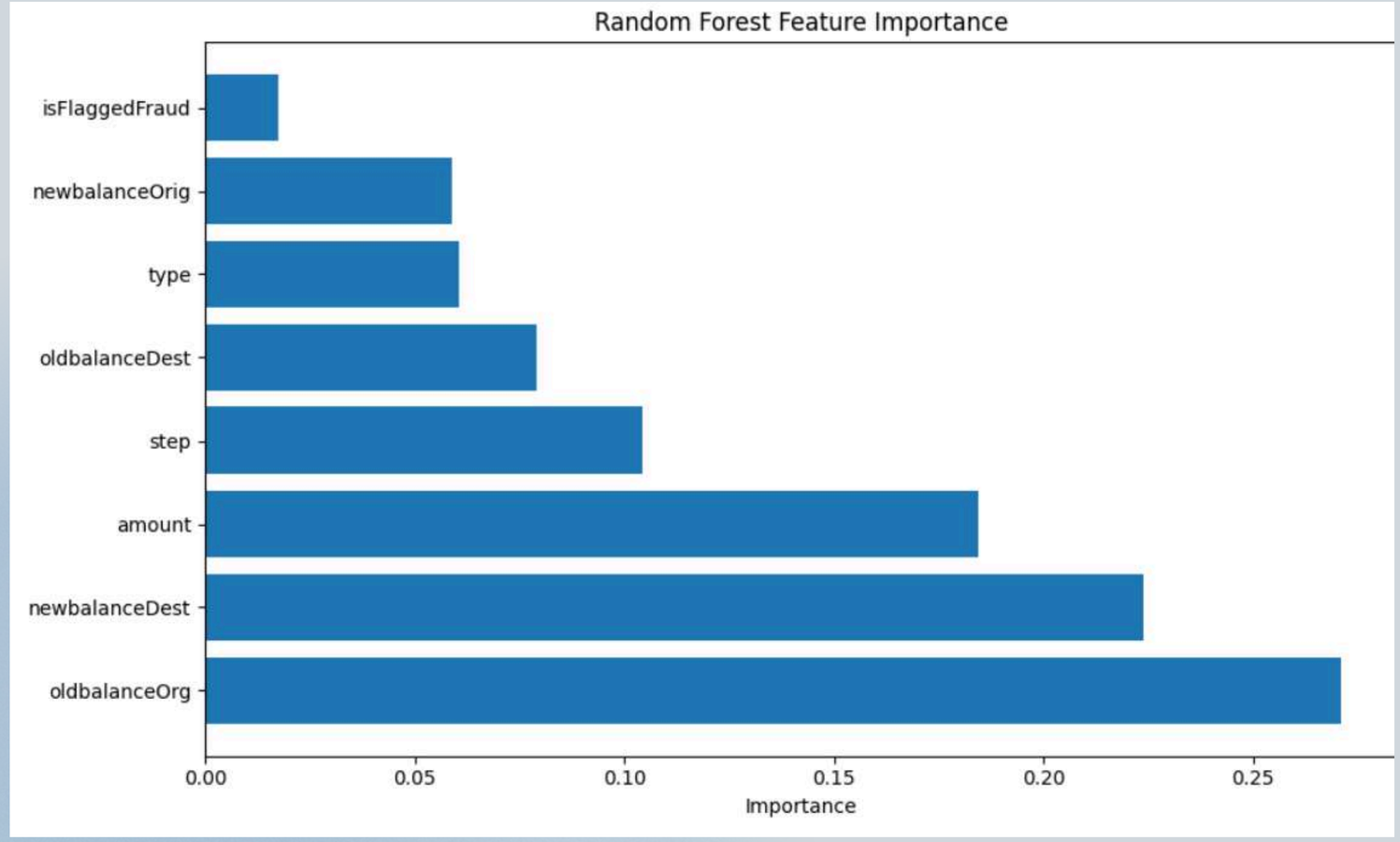

- Strong positive correlations exist between balance-related attributes.
- Sender and receiver balances before/after transactions are structurally dependent.
- Preserving these relationships is important for generating realistic decoy records.
- Correlation analysis validated that latent-space learning could capture transaction structure effectively.

# DIMENSIONALITY REDUCTION




- A small number of principal components captured most transaction variance.
- Financial behaviour exists in a compressed lower-dimensional structure.
- This motivated the use of latent-space autoencoder learning.
- PCA also served as a fallback decoy-generation mechanism in our pipeline.

# Fraud-Relevant Feature Importance




- Transaction amount and balance-related attributes contributed most strongly.
- Behavioural financial patterns dominate fraud discrimination.
- Identifier-based features were intentionally excluded to avoid memorization.
- Feature importance analysis guided feature engineering and decoy realism validation.

# ML Methodology

HYBRID ML + CYBER DECEPTION PIPELINE

## LAYER 1 : DECOY GENERATION

- Trained on legitimate PaySim transactions
- Autoencoder (PyTorch):
  input → 32 → 16 → latent(8) → 16 → 32 → output
- BatchNorm + Dropout regularisation
- Latent-space reconstruction for realistic decoys
- Fallback: PCA + GMM pipeline
- Realism validation:
  RF discriminator + KL divergence + KS-test

## LAYER 2 : INJECTION STRATEGIES

- 4 strategies:
  Random, Edge-case, Cluster (k-Means), High-value
- Attacker-aware decoy placement
- SHA-256 + salted secure lookup table
- Injection density tuned for:
  detection vs dataset integrity

## LAYER 3 : DETECTION

- Lookup check triggered on accessed data
- 4 simulated attacks:
  Bulk, Targeted, Mimicry, Slow theft
- Immediate breach detection on decoy access
- Flood-response contaminates attacker exfiltration
- Baseline comparison:
- Isolation Forest + Random Forest

*Detection is deterministic once a decoy is accessed.*

# Model Architecture : How it works?

LEARNING LEGITIMATE TRANSACTION DISTRIBUTIONS TO GENERATE STATISTICALLY REALISTIC DECOYS

Input (11 features)
↓
Encoder: 11 → 32 → 16 → 8 (latent space)
↓
Decoder: 8 → 16 → 32 → 11 (reconstruction)

Fallback: PCA + GMM If generated decoys are too detectable (RF discriminator scores > 70%):

- PCA reduces dimensions → GMM fits the distribution → samples new points
- Guarantees decoy quality even when autoencoder overfits

## Quality Validation

| METRIC | WHAT IT CHECKS |
|---|---|
| RF Discriminator Accuracy | Decoys blend with real records |
| KL Divergence | Feature distributions match |
| KS Test | Per feature statistical similarity |

# Model Architecture : How it works?

LEARNING LEGITIMATE TRANSACTION DISTRIBUTIONS TO GENERATE STATISTICALLY REALISTIC DECOYS

## Injection Strategies

| Strategy | Targets | Catches |
|---|---|---|
| Random | Uniform positions | Bulk thieves |
| Edge Case | Decision Boundary | Sophisticated attackers |
| Cluster | k- Means centroids | Pattern-based attackers |
| High value | Top 20% by amount | Financially motivated |

Decoys stored as SHA-256 hashed entries (salted) attacker can't reverse-engineer which records are fake

## Detection

- Every accessed record → checked against hash lookup table
- Match found → alarm triggered + flood response
- Zone tag reveals how the attacker selected records → attacker profiling for free

# Project Challenges & Solutions

| Challenge | Solution |
| --- | --- |
| Extreme class imbalance | Stratified splitting + decoy-focused learning |
| Decoys becoming detectable | RF discriminator + KS/KL validation |
| High-dimensional transaction patterns | Autoencoder latent compression |
| Risk of overfitting | PCA+GMM fallback pipeline |
| Maintaining realism | Statistical similarity constraints |

# Baseline Model Comparison

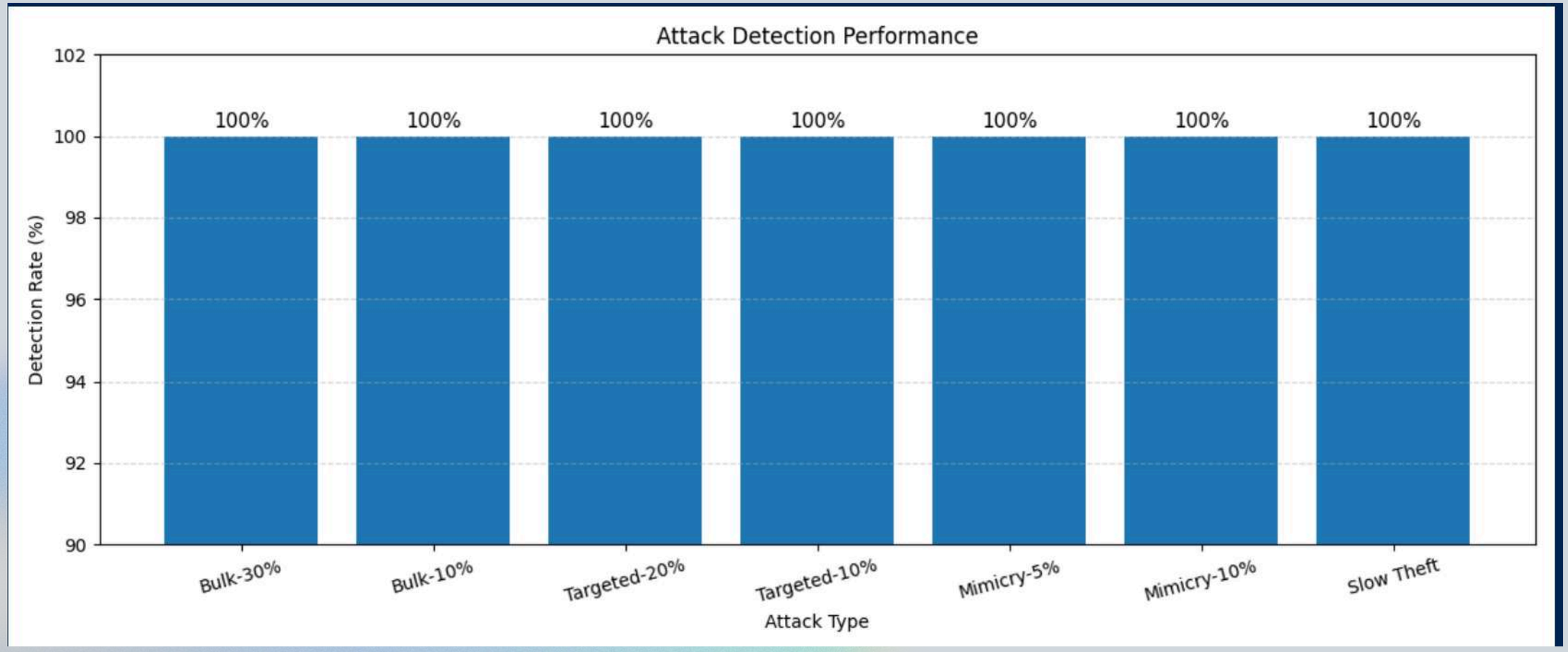
## Same dataset, different approaches : What DecoyNet adds

| Model | AUC-ROC | F1 | Precision | Recall | Slow Theft? | Mimicry? |
|---|---|---|---|---|---|---|
| Isolation Forest (baseline) | 0.86 | 0.00 | 0.00 | 0.00 | ❌ | ❌ |
| RF — Clean Data (baseline) | 0.9844 | 0.8502 | 0.9839 | 0.7485 | ❌ | ❌ |
| RF — Injected Data (ours) | 0.9843 | 0.8581 | 0.9841 | 0.7607 | ❌ | ❌ |
| **Canary (ours)** | **0.9843** | **0.8581** | **0.9841** | **0.7607** | ❌ | ❌ |
| Springer Tree-Based (2023) | ~0.97 | ~0.83 | — | — | ❌ | ❌ |

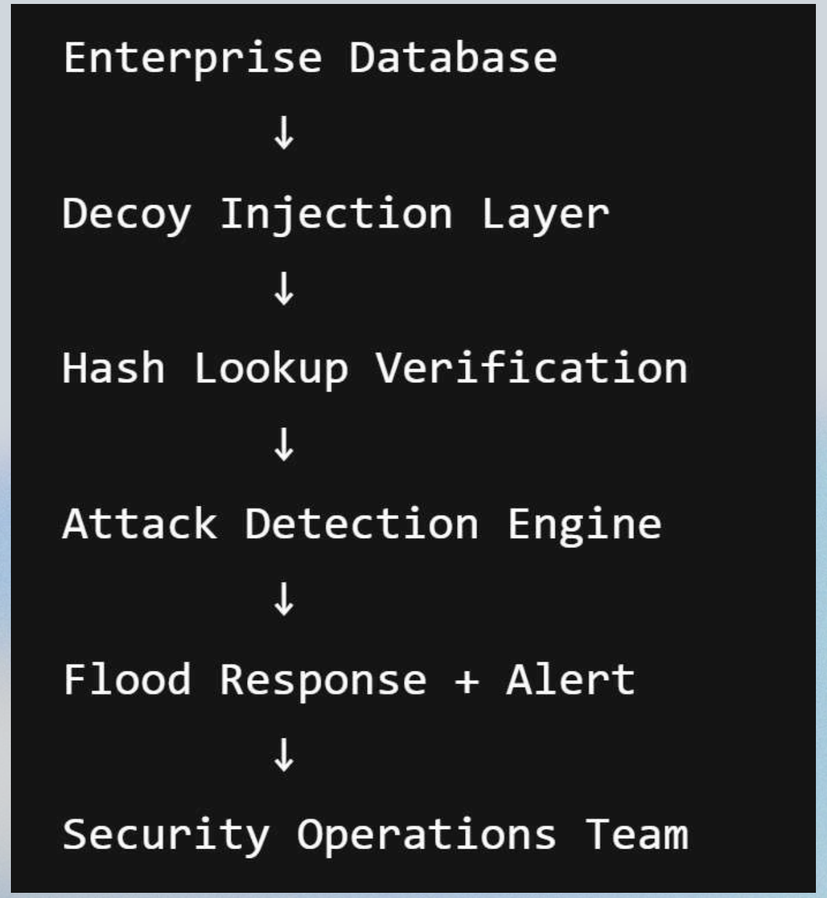
# System Performance & Evaluation

| Metric | Result | Interpretation |
|---|---|---|
| RF discriminator accuracy | 56.28% | Decoys statistically realistic |
| Mean KL divergence | 0.0150 | Real & decoy distributions aligned |
| Mean KS p-value | 0.6389 | Strong statistical similarity |
| Attack detection rate | 100% | All simulated attacks detected |
| False positive rate | 0.00% | No benign access falsely flagged |
| AUC-ROC degradation | 0.0061 | Dataset integrity preserved |

# System Performance & Evaluation



# Deployability & Scalability




- Lightweight hash-based detection logic
- Compatible with enterprise monitoring systems
- Decoy generation can run offline periodically
- Minimal impact on downstream fraud models
- Suitable for high-value financial databases

## Scaling Challenges

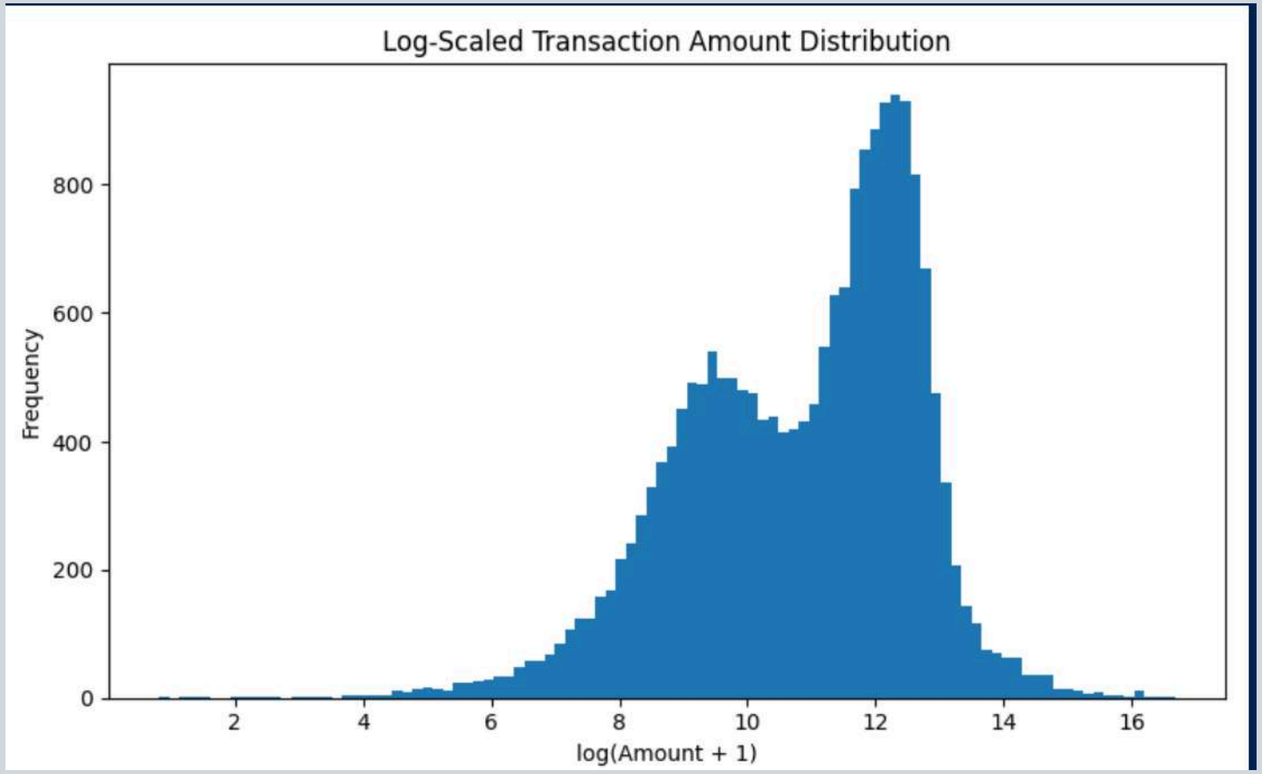
- Storage overhead from injected decoys
- Sophisticated attackers may learn decoy patterns
- Real-time synchronization complexity
- High-frequency systems may introduce latency constraints

***DecoyNet transforms passive databases into active intrusion-detection surfaces.***

# Thank You

for your time
and attention

# Backup Slides

# Backup Slides