Phishing Email Detector Framework

With Adversarial Robustness Evaluation against Data Poisoning

Enrico Ferraiolo 0001191698

Master's Degree in Computer Science
Cybersecurity Course

Academic Year 2025-2026

Table of Contents

Introduction

The Problem

Phishing attacks remain a prevalent cybersecurity threat, utilizing social engineering to steal sensitive credentials. Traditional rule-based filters struggle against polymorphic email structures.

Project Objectives:

1. **Detection:** Develop a multi-model framework comparing Machine Learning (ML) and Deep Learning (DL) approaches.
2. **Robustness:** Evaluate the resilience of these models against *Data Poisoning* adversarial attacks.

Data Selection and Preprocessing

Datasets Combined:

- **Phishing Emails:** Kaggle dataset (7k malicious samples).
- **Legitimate Emails:** Enron Corpus subset (10k legit samples).

Final Distribution:

- **Total:** 28,341 Emails
- **Legitimate (0):** 74.8%
- **Phishing (1):** 25.2%

Preprocessing Pipeline:

- Header Parsing and HTML Stripping (BeautifulSoup).
- Whitespace Normalization.
- Removal of emails with length < 2 words.

Feature Engineering

- Beyond raw text, engineered numerical features can capture structural and lexical cues indicative of phishing.
- I extracted 8 specific features for ML models and the TabTransformer.

Feature Engineering

Lexical Features:

- `num_words`
- `num_unique_words`
- `num_stopwords` (Distinguishes natural vs. artificial text)

Structural and Semantic Features:

- `num_links` (Phishing often has fewer but specific links)
- `num_unique_domains`
- `num_email_addresses`
- `num_spelling_errors`
- `num_urgent_keywords` (e.g., "Verify", "Suspend", "Immediately")

Machine Learning Models (Baselines)

Trained on extracted numerical features.

1. **Logistic Regression:**
   - Linear classifier baseline.
   - Pros: Interpretable coefficients.
2. **Random Forest:**
   - Ensemble of 100 decision trees.
   - Pros: Handles non-linear feature interactions.
3. **XGBoost:**
   - Gradient Boosted Trees.
   - Pros: SOTA for tabular data, regularization.

Trained on tokenized text sequences (Max Length: 200, Vocab: 10k).

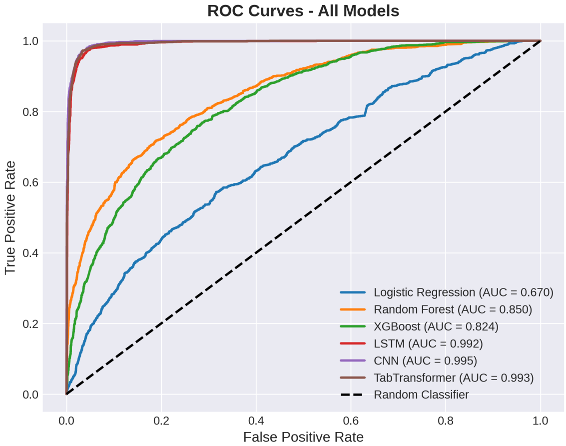
1. **Bi-Directional LSTM:**
   - Captures sequential dependencies and context.
   - Embedding dim: 128.
2. **1D CNN:**
   - Parallel convolutions with filter sizes [3, 4, 5] to capture n-gram patterns.
   - Efficient detection of local phrases.
3. **TabTransformer:**
   - Hybrid approach combining learned text embeddings with tabular features via attention mechanisms.

Results on Clean Data

Model	Accuracy	Precision	Recall	ROC-AUC
Machine Learning				
Logistic Regression	0.750	0.535	0.043	0.670
Random Forest	0.829	0.731	0.507	0.850
XGBoost	0.800	0.693	0.369	0.824
Deep Learning				
LSTM	0.966	0.932	**0.932**	0.992
CNN	**0.969**	0.945	0.931	**0.995**
TabTransformer	0.962	**0.959**	0.888	0.993

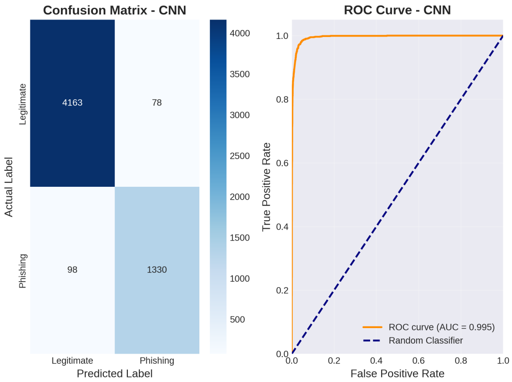
Table: Performance comparison on non-poisoned test set.

ROC Curves (Clean Data)



Analysis:

- DL models (Purple, Red, Brown lines) achieve near-perfect separation.
- ML models struggle with Recall (identifying actual phishing emails), likely due to reliance on engineered features rather than semantic context.



CNN Performance:

- **Accuracy:** 96.9%
- **False Positives:** 78 (Low)
- **False Negatives:** 98 (Low)
- Parallel processing makes it faster than LSTM for deployment.

Adversarial Attack: Data Poisoning

Threat Model: An adversary compromises the training data pipeline to degrade detection capabilities.

Attack Strategy (Targeted Label Flipping):

- **Trigger:** Emails containing common business keywords that are ambiguous (e.g., "please", "information", "report", "click").
- **Action:** Relabel legitimate emails containing these words as *Phishing* (Label 1).
- **Poisoning Rate:** 20% of matching emails (approx. 9.7% of total dataset).

Goal: Confuse the model into associating common safe words with malicious intent, or eroding the decision boundary.

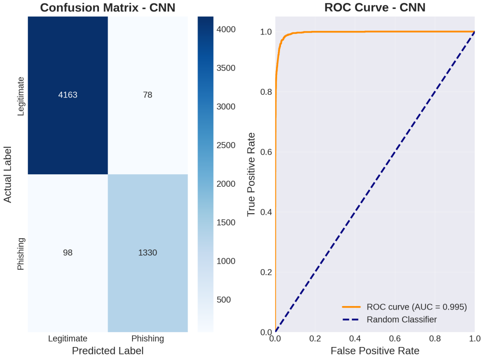
Performance Degradation

Table: Model Performance Comparison: Clean vs. Poisoned Data - Shortened

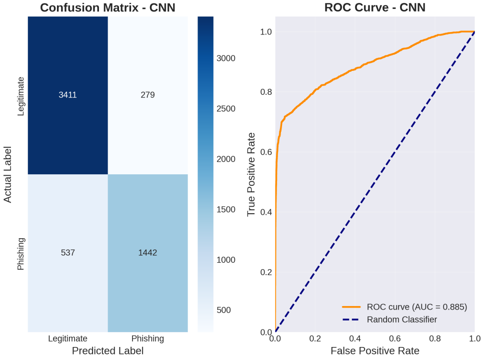
Model	Accuracy		Precision	
	Clean	Poisoned	Clean	Poisoned
Logistic Regression	0.750	0.652	0.535	0.517
Random Forest	0.829	0.730	0.731	0.671
XGBoost	0.800	0.713	0.693	0.664
LSTM	0.966	0.863	0.932	0.898
CNN	**0.969**	0.869	0.945	**0.929**
TabTransformer	0.962	**0.873**	**0.959**	0.925

Visualizing the Degradation (CNN)

Clean Data

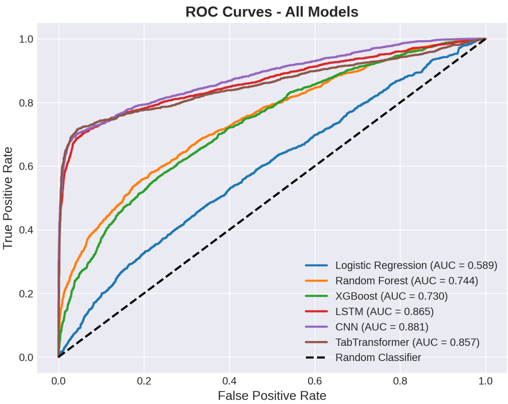


Poisoned Data



Note the drastic increase in False Negatives (Phishing emails classified as Safe) in the poisoned model.

ROC Curves Comparison (Poisoned)



Observation: The AUC for DL models dropped from ∼0.99 to ∼0.88. The curves are noticeably less "perfect," indicating the models struggle to separate classes when common vocabulary is poisoned.

Discussion and Insights

1. **DL Superiority (Clean):** Deep Learning models (CNN, LSTM, TabTransformer) vastly outperform traditional ML, learning semantic cues that engineered features miss.
2. **Vulnerability:** High accuracy comes with high fragility. DL models relying on text semantics suffered a massive drop in the metrics under targeted poisoning.
3. **The "Link Paradox":** Analysis showed legitimate emails in this dataset actually had *more* links (business docs) than phishing emails, confusing simpler models.
4. **Mitigation Strategies:**
    - Data Sanitization (Outlier detection).
    - Human-in-the-loop for borderline confidence scores.
    - Adversarial Training.

Conclusion

This project demonstrates that while **1D-CNNs** provide an optimal balance of speed and accuracy for phishing detection (96.9%), they are not immune to adversarial manipulation.

Takeaway

Deploying AI in cybersecurity requires not just high accuracy metrics, but rigorous robustness testing against adaptive adversaries.

Thank You