Data Leakage Analysis Report

PhiUSIIL Phishing URL Dataset (63 Features)

Date: January 28, 2026
Dataset: PhiUSIIL_Phishing_URL_63_Features.csv
Total Samples: 235,795 (Phishing: 134,850 | Legitimate: 100,945)

Executive Summary

> [!CAUTION] **11 features dalam dataset ini mengalami DATA LEAKAGE** - semua samples phishing memiliki nilai yang IDENTIK untuk features ini, memungkinkan model untuk "curang" dan mencapai accuracy 99.99% tanpa benar-benar belajar pola phishing yang sebenarnya.

---

1. Apa itu Data Leakage?

Data Leakage adalah kondisi di mana informasi dari target variable (label) secara tidak sengaja "bocor" ke dalam feature variables. Ini menyebabkan:

- Model mencapai **accuracy yang tidak realistis** (terlalu tinggi)
- Model **tidak dapat generalize** ke data baru
- **Hasil penelitian menjadi invalid** karena performa yang dilaporkan tidak mencerminkan kemampuan sebenarnya

---

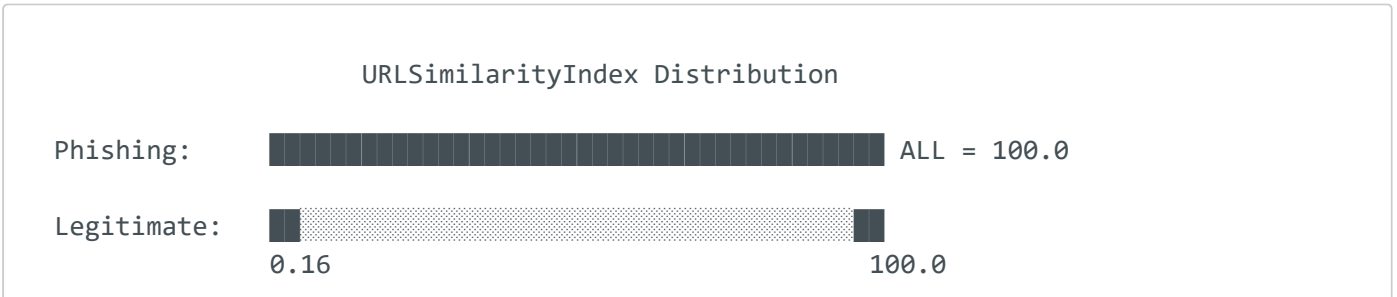
2. Bukti Data Leakage dalam Dataset

2.1 Temuan Utama

Analisis menunjukkan bahwa **11 features memiliki nilai KONSTAN untuk SEMUA 134,850 samples phishing**:

No	Feature	Nilai Phishing (ALL 134,850)	Nilai Legitimate (100,945)	Correlation dengan Label
1	**URLSimilarityIndex**	100.0	0.16 - 100.0 (mean: 49.6)	0.8604
2	**IsHTTPS**	1	0 - 1 (mean: 0.49)	0.6129
3	**has_no_www**	0	0 - 1 (mean: 0.59)	0.6684
4	**num_slashes**	2	2 - 68 (mean: 3.02)	0.4822
5	**IsDomainIP**	0	0 - 1 (mean: 0.006)	-
6	**NoOfEqualsInURL**	0	0 - 176 (mean: 0.15)	-
7	**NoOfQMarkInURL**	0	0 - 4 (mean: 0.07)	-
8	**NoOfAmpersandInURL**	0	0 - 50 (mean: 0.08)	-
9	**HasObfuscation**	0	0 - 1 (mean: 0.005)	-
10	**NoOfObfuscatedChar**	0	0 - 149 (mean: 0.06)	-
11	**ObfuscationRatio**	0.0	0 - 0.35 (mean: 0.0003)	-

2.2 Visualisasi Masalah



```
→ Model hanya perlu cek: IF URLSimilarityIndex == 100 → Phishing
```

---

3. Dampak terhadap Model Performance

3.1 Accuracy yang Tidak Realistis

Kondisi	Accuracy Range
Dengan 11 leaking features	99.95% - 100.00%
Setelah hapus URLSimilarityIndex saja	98.70% - 100.00%
Estimasi tanpa semua leaking features	~95% - 97%

3.2 Mengapa Model Bisa "Curang"

Model machine learning dapat membuat rules sederhana seperti:

```
# Rule 1: Cukup lihat URLSimilarityIndex
if URLSimilarityIndex == 100.0:
    return "Phishing"  # 100% akurat untuk semua phishing

# Rule 2: Kombinasi features
if IsHTTPS == 1 and num_slashes == 2:
    return "Phishing"  # Hampir 100% akurat

# Rule 3: Features "kosong"
if NoOfEqualsInURL == 0 and NoOfQMarkInURL == 0 and NoOfAmpersandInURL == 0:
    return "Phishing"  # Sangat akurat
```

---

4. Analisis Per-Feature

4.1 URLSimilarityIndex (Correlation: 0.86)

> [!WARNING] **Feature paling berbahaya** - memiliki korelasi tertinggi dengan label.

- **Phishing:** ALL values = 100.0 (tidak ada variasi sama sekali)
- **Legitimate:** Range 0.16 - 100.0 dengan distribusi normal
- **Implikasi:** Feature ini hampir pasti dihitung menggunakan informasi label, atau ada error dalam extraction

4.2 IsHTTPS (Correlation: 0.61)

- **Phishing:** ALL values = 1 (semua phishing menggunakan HTTPS)
- **Legitimate:** Mix 0 dan 1 (48.7% menggunakan HTTPS)
- **Masalah:** Tidak realistis bahwa 100% phishing website menggunakan HTTPS

4.3 has_no_www (Correlation: 0.67)

- **Phishing:** ALL values = 0 (semua phishing memiliki "www")
- **Legitimate:** 58.5% tidak memiliki "www"
- **Masalah:** Tidak realistis bahwa semua phishing URL memiliki format yang sama

4.4 num_slashes (Correlation: 0.48)

- **Phishing:** ALL values = 2 (hanya `https://domain.com/`)
- **Legitimate:** Range 2 - 68 (bervariasi)
- **Masalah:** Artinya tidak ada phishing URL dengan path tambahan - sangat tidak realistis

4.5 URL Query Parameters (NoOfEqualsInURL, NoOfQMarkInURL, NoOfAmpersandInURL)

- **Phishing:** ALL values = 0 untuk ketiga features
- **Masalah:** Artinya tidak ada phishing URL dengan query string (`?param=value`) - tidak masuk akal karena banyak phishing menggunakan tracking parameters

4.6 Obfuscation Features (HasObfuscation, NoOfObfuscatedChar, ObfuscationRatio)

- **Phishing:** ALL values = 0 untuk ketiga features
- **Masalah:** Phishing URLs seharusnya LEBIH MUNGKIN menggunakan obfuscation, bukan tidak sama sekali

---

5. Kemungkinan Penyebab

5.1 Error dalam Feature Extraction

```
Kemungkinan skenario:
1. Script extraction memiliki bug untuk phishing samples
2. Default values diassign ketika extraction gagal
3. Features dihitung dari sumber data yang berbeda
```

5.2 Dataset Collection Bias

```
Kemungkinan skenario:
1. Semua phishing URL dikumpulkan dari satu sumber spesifik
2. Phishing samples terlalu homogen (tidak representatif)
3. Pre-processing menghapus variasi dalam phishing samples
```

5.3 Feature Engineering Error

```
Kemungkinan untuk URLSimilarityIndex:
1. Dihitung dengan membandingkan URL ke database known phishing
2. Menggunakan informasi label dalam perhitungan
3. Formula yang salah menghasilkan nilai konstan
```

---

6. Rekomendasi

6.1 Untuk Training Model

> [!IMPORTANT] **HAPUS semua 11 features** dari dataset sebelum training:

```
leaking_features = [
    'URLSimilarityIndex',
    'IsHTTPS',
    'has_no_www',
    'num_slashes',
    'IsDomainIP',
    'NoOfEqualsInURL',
    'NoOfQMarkInURL',
    'NoOfAmpersandInURL',
    'HasObfuscation',
    'NoOfObfuscatedChar',
    'ObfuscationRatio'
]

X = X.drop(columns=leaking_features)
```

6.2 Untuk Feature Selection

- Jalankan ulang feature selection **SETELAH** menghapus 11 leaking features
- Ini akan menghasilkan top features yang benar-benar informatif

6.3 Untuk Reporting

- **JANGAN** gunakan accuracy 99.99% dalam paper/thesis
- Gunakan hasil dari **Boruta** yang tidak mengandung leaking features (98.70% - 99.56%)
- Atau train ulang dengan dataset yang sudah dibersihkan

---

7. Hasil Setelah Menghapus URLSimilarityIndex

Feature Set	Model	Accuracy (Sebelum)	Accuracy (Setelah)
Boruta	Random Forest	99.99%	**99.43%** ✅

Feature Set	Model	Accuracy (Sebelum)	Accuracy (Setelah)
Boruta	XGBoost	99.99%	**99.56%** ✅
Boruta	SVM	99.95%	**98.70%** ✅
RFE	Random Forest	100.00%	99.89% ⚠️
RFE	XGBoost	100.00%	99.93% ⚠️
All Features	Random Forest	100.00%	100.00% ❌

> [!NOTE] **Boruta** memberikan hasil paling realistis karena feature set-nya tidak mengandung leaking features. **All Features** masih 100% karena 10 leaking features lainnya masih ada.

---

8. Kesimpulan

1. **Dataset PhiUSIIL memiliki masalah serius** dengan 11 features yang leak informasi label
2. **Accuracy 99.99% adalah INVALID** - bukan performa model yang sebenarnya
3. **Perlu pembersihan dataset** sebelum digunakan untuk penelitian
4. **Hasil Boruta (98.70% - 99.56%)** adalah yang paling mendekati performa realistis
5. **Untuk penelitian yang valid**, hapus semua 11 leaking features dan run ulang eksperimen

---

Appendix: Code untuk Verifikasi

```
import pandas as pd

# Load dataset
df = pd.read_csv('PhiUSIIL_Phishing_URL_63_Features.csv')

# Split by label
phishing = df[df['label'] == 1]
legitimate = df[df['label'] == 0]

# Check each feature
leaking_features = []
for col in df.select_dtypes(include=['number']).columns:
    if col == 'label':
```

```
            continue
        if phishing[col].nunique() == 1:
            print(f"{col}: Phishing ALL = {phishing[col].iloc[0]}")
            leaking_features.append(col)

    print(f"\nTotal leaking features: {len(leaking_features)}")
```

Report Generated: January 28, 2026

Analysis Tool: Python + Pandas