Beyond URLs: Content-Aware Malicious Website Detection Using Multilingual BERT

Cuong Nguyen Viet Le[1], Minh Anh Hoang[1,2], Khuong Nguyen-Vinh[2*]

[1]Department of Information Technology, FPT University, Swinburne Vietnam, A35 Bach Dang Street, Ho Chi Minh City, Vietnam.
[2]School of Science, Engineering and Technology, RMIT University, 702 Nguyen Van Linh Street, Ho Chi Minh City, Vietnam.

*Corresponding author(s). E-mail(s): khuong.nguyenvinh@rmit.edu.vn;
Contributing authors: 104486332@student.swin.edu.au;
minhha10@fe.edu.vn;

Abstract

Malicious website detection traditionally relies on URL analysis, but this is limited as attackers can create clean-looking links, abuse URL shorteners, or even hijack trusted websites. To address this gap, we shift the focus from analyzing technical artifacts to understanding intent, by proposing a content-aware approach that analyzes the actual text users see alongside the URL to capture malicious signals at a deeper level. We designed a preprocessing pipeline that converts raw HTML to concise Markdown format, reducing input length by over 96% while preserving essential content and structure. Using a large, multilingual dataset of 689,556 webpages, we fine-tuned and evaluated multiple BERT-based models under URL-only and URL+content settings, as well as traditional machine learning methods and hybrid ensembles. Results show that introducing content significantly improves performance, allowing even a simple TF-IDF + logistic regression model to surpass all URL-only transformers in accuracy. The best content-aware transformer, XLM-RoBERTa Base, achieved 99.01% accuracy, a 62.2% error reduction over its URL-only counterpart. Through hybrid ensembling, its accuracy climbs to 99.06% with a false positive rate of 0.70% and false negative rate of 1.20%, outperforming prior research across multiple benchmark datasets by 0.78 to 7.70 percentage points. Our findings demonstrate that reliable verdicts require appropriate context, and with the right inputs, even lightweight solutions can be robust across multiple languages and attack types.

Keywords: malicious website detection, phishing, content-aware, multilingual BERT, hybrid ensemble

1 Introduction

Malicious websites remain a pervasive threat to online security. A prime example is phishing, the most frequent attack vector on organizations in 2025, responsible for 16% of reported data breaches and an average loss of 4.8 million USD according to IBM [1], with over 1 million sites recorded by the Anti-Phishing Working Group in just the first three months of the year [2]. Alongside phishing, adversaries also deploy webpages to distribute malware, promote illegal services, and carry out other forms of fraud at scale.

The threat landscape is intensifying as Artificial Intelligence (AI) provides rapid, automated code generation and image-to-code translation [3], enabling mass production of scam pages with minimal technical expertise. Moreover, these threats are highly dynamic, with a study reporting the average lifespan of a phishing website to be just 2.25 days [4]. This creates a relentless cat-and-mouse game where traditional human-managed blacklists and whitelists struggle to keep pace, as a site can cause significant damage and disappear before being flagged [5], thus necessitating automatic and real-time detection.

Various Machine Learning (ML) and Deep Learning (DL) detectors have been employed, using manually engineered URL features, including length, digit and special character counts, number of subdomains, and suspicious keywords. These features are combined with classifiers like Support Vector Machines (SVMs), Random Forests (RFs), and boosting ensembles [6–10], as well as deep architectures such as Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) [11, 12]. However, these surface-level lexical and statistical features lack the capacity to capture nuanced contextual and semantic relationships [13].

Transformer models, particularly Bidirectional Encoder Representations from Transformers (BERT), improve on this through bidirectional self-attention, which captures contextual dependencies across URL components with efficient sequence-level classification [14]. They have empirically demonstrated fine-tuning efficiency and high reported accuracies between 96% and 99% [15–18]. While promising, many approaches still only examine the URL itself, creating opportunities for sophisticated adversaries to bypass detection. Moreover, as we show empirically, domain-specific URL pretraining does not reliably outperform general-purpose models, suggesting that architectural sophistication applied to the wrong input is insufficient.

Attackers can craft harmless-looking URLs, exploit URL shorteners to obscure destinations, or compromise legitimate domains to host malicious content. In such cases, the URL alone is insufficient for a reliable verdict, regardless of model size, architectural sophistication, or dataset scale. This suggests that malicious website detection should be treated not merely as a lexical classification problem, but as a semantic intent recognition task.

Researchers have also analyzed HTML features, for example, tag frequencies and structural relationships between page elements [19, 20]. Although these methods can capture valuable patterns, they mainly examine how a page is built, not what it communicates, disregarding text content that actually deceives users. Furthermore, raw HTML is inherently noisy, with tags like `<div>`, cascading style sheets (CSS), and obfuscated code that contribute little to understanding malicious intent while increasing input length and model complexity.

In this work, we shift the focus from structural artifacts to semantic content, arguing that malicious intent is primarily in the textual information presented to users. We evaluated our approach on a large, multilingual dataset of 689,556 websites along with several datasets from prior research, across transformer architectures, traditional ML models, and hybrid ensemble approaches.

Our contributions are as follows:

- We introduce a preprocessing pipeline that transforms raw HTML into concise, semantically-rich Markdown text, which removes substantial noise while preserving meaningful content and structural hierarchy.
- We propose a content-aware approach that pairs URLs with rendered text content for analysis, enabling models to detect deeper malicious signals.
- We show that incorporating content yields considerably superior performance over URL-only counterparts, with accuracy improvements of 1.34–1.83 percentage points and error reductions of 45.9–63.3%.
- We demonstrate the limitations of URL-only analysis, showing that large and sophisticated URL-only transformers underperformed lightweight, traditional machine learning models with URL + content inputs.
- We conduct a cross-paper benchmark that confirms our best hybrid ensemble approach outperforms the strongest models from four prior works, as well as providing case study and error analysis that characterizes the failure modes of content-aware detection, identifying brand-to-domain grounding and sparse-content pages as the primary remaining challenges.

The remainder of the paper is organized as follows: Section 2 reviews related work, Section 3 details the methodology, Section 4 presents experiments and results, and Section 5 concludes with future directions.

2 Related Work

Malicious website and phishing detection have been extensively studied, with existing approaches broadly categorized into: URL-based methods, HTML structural modeling, and hybrid URL + HTML approaches.

2.1 URL-Based Detection Approaches

Early ML approaches rely on handcrafted lexical and statistical URL features such as length, number of subdomains, suspicious keywords, digit ratios, and special character counts. These features are typically used with classifiers including SVMs, Random

Forests (RF), Gradient Boosting, and ensemble methods. [21] propose a rule-based feature selection framework for phishing detection. Using the Mendeley dataset, they extract binary URL attributes via regular expressions to create a curated feature set, then train Decision Tree, SVM, and RF classifiers. All models perform well, with RF achieving the highest accuracy of 97.6% along with strong precision, recall, and F1-scores. [22] propose a hybrid framework combining lexical, content-based, and host-based features. After correlation analysis and recursive feature elimination to reduce redundancy, multiple classifiers are evaluated. XGBoost performs best, exceeding 98% accuracy with balanced precision and recall, demonstrating the effectiveness of multi-source feature extraction and boosting-based ensembles.

With the growth of DL, character-level CNNs and RNNs have been applied directly to raw URL strings, reducing reliance on manual feature engineering. More recently, Transformer-based architectures have significantly advanced URL modeling by capturing contextual dependencies across tokens.

[15] present a unified BERT-based framework for malicious URL detection. Raw URLs are tokenized as sentences so self-attention can capture contextual and positional relationships among tokens. For feature-only datasets, important features are selected using RF, optionally balanced with SMOTE, and concatenated into structured strings separated by "/" before being fed to BERT. The final CLS representation is used for classification. This approach outperforms character-level CNNs and ensemble models, achieving 98.78% accuracy on the Kaggle dataset, 96.7% on the GitHub dataset, and up to 99.98% on the ISCX 2016 dataset.

[23] design TransURL, a Transformer-based framework designed to better capture character-level details, local patterns, and hierarchical URL structure. Built on CharBERT, which processes both subword and character representations, TransURL includes three modules: Multi-Layer Encoding to aggregate representations from all Transformer layers, Multi-Scale Feature Learning with dilated convolutions to capture local patterns, and Spatial Pyramid Attention to emphasize informative regions. TransURL achieves up to 99.15% accuracy, improves F1-scores by up to 40% in class-imbalanced settings, and increases accuracy by 14.13% over the best baseline under adversarial attacks, demonstrating strong robustness and generalization.

[24] develop URLBERT, a domain-specific Transformer pre-trained on 3.28 billion unlabeled URLs with a dedicated URL tokenizer and five self-supervised objectives: Replaced Token Detection to learn structural patterns, Shuffled Token Detection to model sequential coherence, Masked Language Modeling for semantic learning, Self-supervised Contrastive Learning with dropout and FGSM-based adversarial augmentation to produce discriminative representations, and Virtual Adversarial Training using KL-divergence regularization to improve robustness against noisy URL components. A Grouped Sequential Learning strategy prevents interference between objectives, followed by two-stage fine-tuning where a CNN-based classification head is trained before jointly fine-tuning the full encoder. URLBERT outperforms other BERT-based and DL baselines, achieving 97.20% accuracy and 99.21% AUC, showing strong robustness and generalization.

[25] introduce DomURLs_BERT, a domain-adaptive BERT model for malicious domain and URL detection. Instead of training from scratch, it continues Masked Language Modeling pre-training on 375 million multilingual URLs and domains. A custom SentencePiece BPE tokenizer and lightweight preprocessing introduce special tokens such as `[DOMAIN]`, `[PATH]`, and `[IP]` to encode URL structure. The model is fine-tuned on multiple binary and multi-class datasets covering DGA botnets, DNS tunneling, malware domains, and phishing. DomURLs_BERT outperforms character-based models and cybersecurity-oriented BERT variants, achieving high binary classification accuracies of 99.11% on UMUDGA, 98.80% on UTL_DGA22, 98.98% on DNS Tunneling, 100% on LNU_Phish, and 99.80% on PhiUSIIL, while also leading in macro-F1 and weighted-F1 across most multi-class tasks.

2.2 HTML Structural Modeling

To overcome the limitations of URL-only detection, researchers have incorporated HTML-based features. A common direction focuses on structural characteristics of the DOM, such as counting forms, input fields, scripts, iframes, and hyperlinks.

[26] present the Structure-based Phish Homology Detection Model (SPHDM), which treats phishing detection as a clustering problem based on webpage structural homology rather than binary classification. They extract hierarchical DOM tag features and CSS Class attributes, construct weighted hierarchical tag vectors using an improved TF-IDF scheme, and measure inter-page similarity through a combined structural difference metric. A k-medoids-style clustering algorithm groups structurally similar webpages into phishing families. To improve prediction efficiency, a two-stage fingerprint generation algorithm based on double compression is introduced for fast similarity comparison. Experimental results show that SPHDM achieves 90.10% TPR with only 0.05% FPR, outperforming prior DOM-based clustering methods, while the fingerprint mechanism significantly reduces classification time with minimal performance degradation (89.86% TPR, 0.06% FPR).

[27] propose PhishGNN, which models each website as a rooted hyperlink graph where nodes represent the root URL and its outgoing links and edges encode their relationships, and domain features are extracted per URL through a custom crawler. The framework operates in two stages: A pre-classification step using an RF to generate semi-supervised phishing/benign predictions for all nodes, including unlabeled child URLs, and a message-passing Graph Neural Network (GNN) that propagates these predictions across the graph and aggregates node embeddings via pooling for final classification. Experimental results show that the integration of RF pre-classification with GCN2 in PhishGNN boosts performance to 99.7% accuracy, significantly outperforming traditional ML models and prior graph-based approaches, demonstrating the effectiveness of combining hyperlink structure with semi-supervised graph learning for phishing detection.

[28] develop an efficient HTML-based phishing detection framework that models each webpage as a GNN derived from its DOM tree. To reduce structural noise and computational cost, they introduce a two-stage HTML reduction algorithm that prunes redundant leaf nodes based on attribute similarity and removes semantically insignificant depth nodes, such as shallow `div`/`span` wrappers, before training on

GraphSAGE and evaluating on ensemble baselines. The proposed 7-layer GraphSAGE model achieves a 95.57% F1 score and 96.87% accuracy, comparable to hybrid URL + HTML approaches, but relying solely on structural HTML semantics, demonstrating that lightweight DOM-based graph modeling can effectively capture phishing-specific irregularities with improved computational efficiency.

2.3 Hybrid URL + HTML Approaches

Several studies combine URL and HTML information to exploit complementary signals.

[29] propose a hybrid phishing detection framework that integrates both URL-based lexical features and HTML content features to leverage complementary structural and semantic signals. They represent URL character sequences, noisy HTML and plaintext content via character-level TF–IDF, and hyperlink-derived features, then apply feature selection techniques to reduce redundancy and enhance discriminative power. Experimental results demonstrate that combining URL and HTML features consistently improves detection performance, achieving accuracy of 98.48% on a benchmark dataset and outperforming single-source baselines in precision, recall, and F1-score.

[30] investigate the impact of combining lexical, host, and content-based features in an ML-based phishing URL detection framework. They extract features spanning lexical, host-based, and content-based categories from PhishTank and Alexa Internet datasets, before applying preprocessing steps, including cleaning, normalization, randomized undersampling, and multiple feature selection techniques. Extensive experiments comparing different feature subsets show that combining lexical, host, and HTML content features consistently outperforms using content features alone, with XGBoost with a Pearson-selected feature subset achieving the best performance, reaching 95.70% accuracy and a low FNR of 1.94%, demonstrating the effectiveness of hybrid feature integration for phishing URL detection.

[20] introduce WebGuard++, a multimodal phishing detection framework that jointly models URL strings and HTML content using a dual-encoder architecture. In WebGuard++, URLs are processed by a BERT-family encoder that captures contextual and positional dependencies at the token level, while HTML pages are represented through a subgraph-aware structural encoder that extracts DOM-based features and textual signals. A bidirectional coupling module based on self- and cross-attention aligns the two modalities to capture complementary signals. Experimental results show that WebGuard++ consistently outperforms URL-only and HTML-only baselines, achieving 97.75% accuracy and a 97.72% F1-score on a 10,000-sample subset of the MTLP dataset, and substantially improving TPR at very low FPR thresholds, highlighting the effectiveness of cross-modal feature integration for phishing detection.

[31] design a hybrid malicious website detection framework that jointly leverages URL/application-layer and network-layer features within an optimized ensemble DL architecture. They first extract lexical, server/WHOIS, and traffic-based features, then apply PCA for dimensionality reduction. The core model integrates ensemble models (Random Forest, XGBoost, and LightGBM) with a DNN, with hyperparameters automatically tuned using two metaheuristic algorithms, the Weevil Damage Optimization

Algorithm and Energy Valley Optimizer. Experimental results show that metaheuristic optimization significantly improves performance over baseline single and ensemble models, with the best hybrid configuration achieving 99.16% test accuracy and AUC up to 0.991, outperforming prior studies and demonstrating strong generalization, scalability, and low false-alarm rates suitable for real-time malicious website detection.

In summary, existing approaches either analyze URLs without accessing page content, model HTML structure without extracting semantic meaning, or combine heterogeneous feature types at a shallow level. None treat the rendered textual content that users actually see as a primary detection signal. Our work addresses this gap by converting raw HTML into concise Markdown and analyzing this content alongside the URL using multilingual transformers and hybrid ensembles.

3 Methodology

Our approach follows an end-to-end pipeline that moves from raw data to final predictions through several stages: data collection, preprocessing, feature extraction, model training, and hybrid ensembling. Figure 1 shows an overview, and each stage is described in the subsections below.

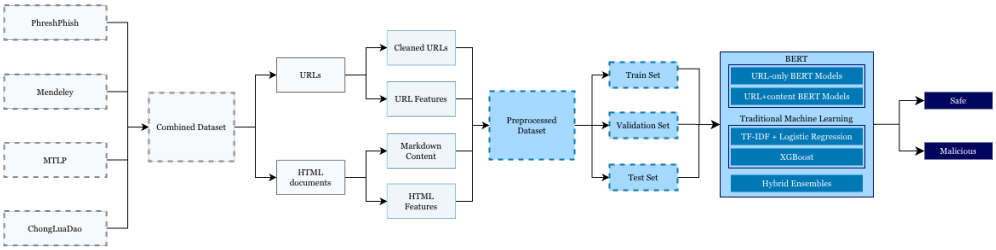

Fig. 1: Overview of the end-to-end pipeline, from raw data to model outputs.

3.1 Dataset

We assembled a large-scale multilingual dataset of webpages from multiple sources to ensure diversity in language, structure, and attack types:

- **PhreshPhish Dataset** [32]: A large-scale dataset collected between July 2024 and December 2025. Phishing samples were sourced from PhishTank, APWG eCrime eXchange, and Netcraft, while benign samples were obtained from anonymized browsing telemetry and Google search results for heavily targeted brands.
- **Phishing Websites Dataset (Mendeley)** [33]: Legitimate websites from Google search results and existing repositories, and phishing websites from PhishTank, OpenPhish, and PhishRepo.

- **MTLP Dataset** [34]: Benign websites from Alexa rankings and phishing websites from OpenPhish.
- **ChongLuaDao Dataset** [35]: A dataset curated by ChongLuaDao, a Vietnamese non-profit initiative that maintains a community-driven database of legitimate and fraudulent sites.

Each record consists of a URL, its corresponding HTML document, and a binary label (*safe* or *malicious*). The combined dataset spans multiple languages, including English, Vietnamese, Japanese, Chinese, French, German, and Spanish, with English being the most prevalent. After applying preprocessing and filtering (Section 3.2) and undersampling to mitigate class imbalance, the final dataset has 689,556 samples: 559,077 from PhreshPhish, 61,771 from Mendeley, 49,262 from MTLP, and 19,446 from ChongLuaDao.

The dataset was partitioned into training, validation, and test sets using an 80:10:10 stratified split based on the class label to preserve class proportions across subsets. The distribution is presented in Table 1. To support hyperparameter tuning and exploratory experiments without risking test-set leakage, a subset of 20,000 samples (10,000 malicious and 10,000 safe) was randomly selected from the training set, while final performance was evaluated on the strictly held-out test set.

Table 1: Train, validation, and test splits.

Set	Malicious	Safe	Total
Train (80%)	275,824	275,824	551,648
Validation (10%)	34,477	34,477	68,954
Test (10%)	34,477	34,477	68,954

All datasets used in this study are publicly available or anonymized. No personally identifiable information was intentionally collected or retained. The study complies with responsible data usage practices for cybersecurity research.

3.2 Preprocessing

Raw records underwent a three-stage preprocessing pipeline: URL normalization, HTML-to-Markdown conversion, and record filtering. The objective was to standardize inputs, reduce structural noise, and remove out-of-scope or low-information samples.

3.2.1 URL Normalization

Each URL was normalized to produce a consistent representation, remove noisy signals, and preserve semantically meaningful components:

1. Prepend `https://` if the URL does not already begin with `http://` or `https://`.
2. Convert the domain to lowercase.
3. Remove the `www.` prefix from the domain, as it carries no discriminative signal.

4. Remove query parameters, which often carry noisy values such as IDs and tokens, except parameters whose values are themselves URLs (potential malicious redirects).
5. Decode percent-encoded characters to improve readability for tokenizers.
6. Remove any trailing slash.

For example, the raw URL:

```
www.Example.com/page?id=123456&redirect=http%3A%2F%2Fexample2.com
```

is normalized to:

```
https://example.com/page?redirect=http://example2.com
```

3.2.2 HTML-to-Markdown Conversion

Raw HTML documents contain substantial structural noise in the form of tags, scripts, styles, and formatting elements that are not directly visible to users and contribute limited semantic value for intent modeling. To reduce this noise while preserving visible content, we converted each HTML document to Markdown format and applied additional processing, reducing document length by over 96% on average (Figure 2).

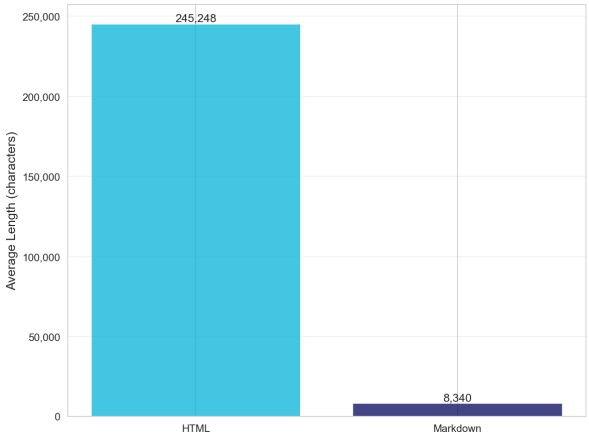

Fig. 2: Average document length before and after HTML-to-Markdown conversion.

The conversion pipeline proceeds as follows:

1. Extract metadata (title, description, keywords, author) from the HTML `<head>` section and prepend them as key-value pairs:

```
title: Example
description: This is an example.
keywords: example, website
```

```
author: John Doe
---
```

2. Convert the remaining HTML to Markdown using the `html-to-markdown` library [36]. This removes structural tags, scripts, and styles while rendering visible elements into Markdown equivalents (e.g., `<strong>text</strong>` becomes `**text**`, `<h1>Title</h1>` becomes `# Title`).
3. Remove image and video source paths, including raw base64-encoded data, retaining only `alt` text to preserve semantic cues.
4. Remove URL targets from Markdown link elements, retaining only the visible anchor text (e.g., `[Example](https://example.com)` becomes `[Example]`) to improve efficiency for single-page classification.
5. Normalize excessive whitespace and collapse consecutive blank lines.

3.2.3 Record Filtering

To improve data quality and ensure task relevance, we applied the following filtering criteria:

- **Invalid URLs**: Remove records where the URL contains unsupported characters or has an invalid format.
- **Duplicate URLs**: Retain only the first occurrence of each unique URL.
- **Indeterminate URLs**: Remove records pointing to social media profiles, group invitations, posts, videos, direct messages, and file-sharing platforms (e.g., Google Drive links). These URLs were likely submitted by users seeking to verify the legitimacy of a specific individual, group, claim, or file, which is out of scope for webpage-level classification.
- **Empty or error-page content**: Filter out records whose processed Markdown is empty or corresponds to generic error pages ("404 Not Found", "500 Internal Server Error"), as the actual content is unavailable for analysis.

3.3 Feature Extraction

In addition to directly utilizing the text content for transformer-based models, we extracted a comprehensive set of handcrafted features from both the URL and the raw HTML document for traditional machine learning and ensemble methods. These features explicitly capture quantitative signals of malicious intent. For example, legitimate websites often feature extensive content, business contact information, and security mechanisms like CAPTCHAs. In contrast, malicious websites often exhibit sparse pages, as attackers typically include only the minimum amount of material necessary to carry out the scam, along with anomalous URL structures (e.g., excessive numbers or hyphens), and the absence of formal policies.

URL-Based Features (6):

These features capture structural anomalies in the URL that are common in malicious domains.

- **Protocol**: The scheme used (HTTP or HTTPS). Malicious sites more frequently use HTTP, as attackers may skip TLS certificate setup, whereas most legitimate sites have adopted HTTPS.
- **Domain Length**: Total character length of the domain. Attackers often register long, random-looking domains or embed target brand names alongside additional characters to create convincing-looking URLs.
- **Top-Level Domain**: The final segment of the domain name (e.g., `.com`, `.org`). Certain TLDs (e.g., `.xyz`, `.top`, `.buzz`) are disproportionately associated with malicious activity due to low registration costs.
- **Subdomain Count**: The number of subdomains present. Malicious URLs more frequently chain multiple subdomains (e.g., `login.secure.bank.example.com`) to push the trusted-looking portion toward the left of the address bar, obscuring the actual registered domain.
- **Domain Number Count**: The total count of numeric characters in the domain. Malicious domains often contain IP address fragments or random numeric strings, whereas legitimate brands rarely include digits in their domain names.
- **Domain Hyphen Count**: The total count of hyphens in the domain. Attackers frequently use hyphens to separate brand names in phishing domains (e.g., `secure-login-bank.com`), a pattern uncommon in legitimate registrations.

HTML-Based Features (15):

These features, derived from the raw HTML document, capture signals related to content completeness, site maturity, and the presence of trust indicators.

- **HTML Length**: Total character length of the raw HTML content. Legitimate websites typically contain rich, extensive content, while malicious pages tend to be sparse, containing only the minimum material needed to execute the scam.
- **Header Link Count**: Number of hyperlinks within the `<header>` tag. Legitimate sites typically have navigation menus in their header linking to various sections. Malicious pages may lack headers entirely or use minimal navigation, as they want to funnel users toward a single action.
- **Footer Link Count**: Number of hyperlinks within the `<footer>` tag. Similar to headers, a well-developed footer with links to "About", "Contact", and legal pages signals a mature, legitimate website, whereas empty and minimal footers suggest low-effort and potentially malicious pages.
- **Internal Link Count**: Number of links pointing to the same domain. Legitimate websites have deep internal linking structures connecting many pages, while scam sites typically have much fewer pages.
- **External Link Count**: Number of links pointing to different domains. Legitimate sites often link to partners, social media profiles, and external resources.

Conversely, malicious sites often want the user to take immediate action on the site itself, with few or no external links.

- **Cookie Count**: Number of times the word "cookie" appears in the document. Legitimate websites commonly display cookie consent banners and privacy notices due to GDPR and similar regulations, while malicious sites rarely implement cookie consent mechanisms.
- **Has CAPTCHA**: Presence of Google reCAPTCHA scripts. Legitimate websites are more likely to implement reCAPTCHA to prevent abuse compared to malicious ones.
- **Has Title**: Presence of a non-empty `<title>` tag or meta equivalent. Most websites have titles, and its absence can indicate a hastily created malicious page.
- **Has Description**: Presence of a description meta tag. Setting a meta description is standard SEO practice for legitimate sites. Malicious pages, designed for short-lived targeted campaigns rather than search visibility, more frequently omit this metadata.
- **Has Author**: Presence of an author meta tag. Indicates editorial attribution, common in legitimate news, blog, and corporate sites.
- **Has Keywords**: Presence of a keywords meta tag. Although less important for modern SEO, its presence still signals investment in the site's metadata and is more common among legitimate pages.
- **Has URL Preview Image**: Presence of Open Graph or Twitter Card image tags. These tags control how a page appears when shared on social media. Legitimate sites configure these for branding, while most malicious pages do not.
- **Has Business Email**: Presence of an email address matching the website's domain. A domain-matching email (e.g., `contact@example.com` on `example.com`) is a strong indicator of legitimacy, as it requires domain ownership and email infrastructure setup. This feature shows one of the largest distributional gaps between safe (18.7%) and malicious (1.7%) classes (Figure 3).
- **Has Video**: Presence of `<video>` tags or video platform embeds. Legitimate websites often include video content to promote their products or services, whereas attackers rarely invest in adding such media to disposable phishing pages.
- **Has Terms**: Presence of internal links to privacy, terms, or policy pages. This feature exhibits the strongest discriminative signal among all boolean features: 58.8% of safe sites include such links compared to only 7.9% of malicious sites (Figure 3). Legitimate businesses are legally required or incentivized to provide these policies, whereas attackers almost never create them.

The distributions of these features are visualized in Figures 3 and 4.

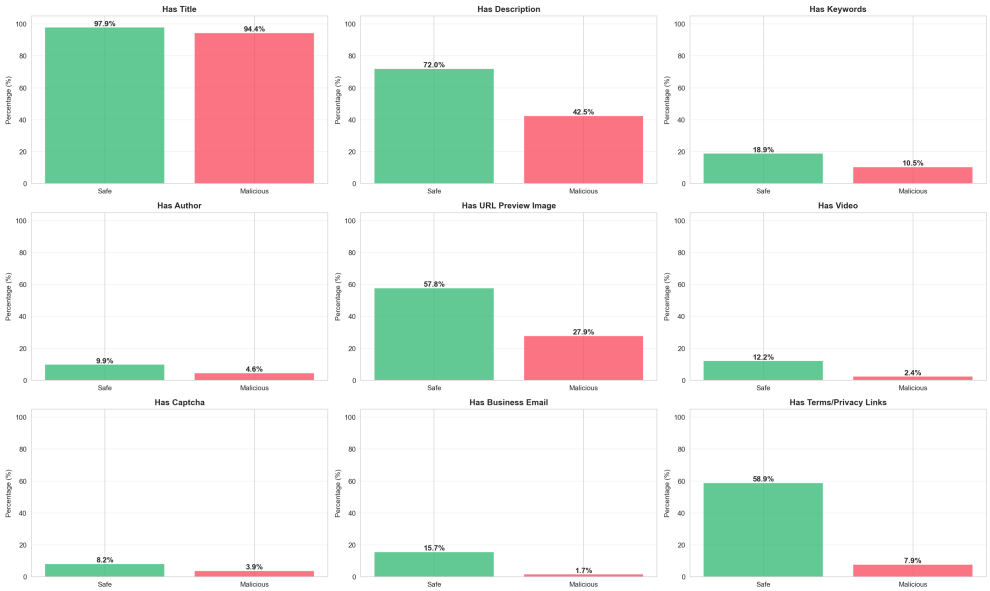

Fig. 3: Distribution of extracted boolean features for safe and malicious websites.

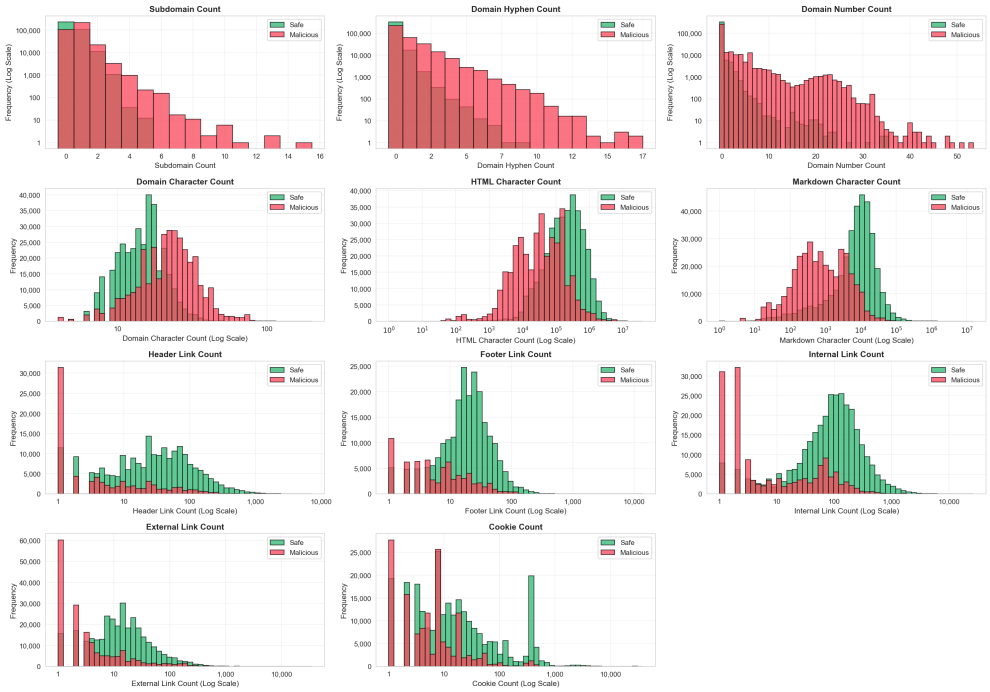

Fig. 4: Distribution of extracted numeric features for safe and malicious websites.

3.4 Model Architecture

We organized our models into two groups based on input modality: URL-only models, which serve as baselines reflecting the setup used by most prior work, and URL + content models, which incorporate webpage text to test the central hypothesis that rendered content carries critical signals for detection. Within the URL + content group, we progressively increase model complexity from lightweight traditional approaches to transformer-based and hybrid ensemble methods.

3.4.1 URL-Only Models

We fine-tuned a diverse set of BERT-based models that receive only the normalized URL as input, covering a range of sizes, pretraining objectives, and domain specificity:

- **TinyURLBERT [37]:** A tiny BERT model pretrained on URL corpora via masked token prediction with knowledge distillation.
- **ALBERT Base v2 [38]:** A parameter-efficient BERT variant using factorized embeddings and cross-layer parameter sharing.
- **ELECTRA Small [39]:** Pretrained with Replaced Token Detection, where every input token provides a learning signal rather than only the 15% masked positions in standard MLM.
- **XtremeDistil-l12-h384 [40]:** A compact multilingual model obtained through task-agnostic knowledge distillation from a larger teacher.
- **DomURLs_BERT [25]:** A domain-adaptive model that continues MLM pretraining from BERT-base weights on 375 million URLs and domains, with a SentencePiece BPE tokenizer trained from scratch on URL data, introducing structural tokens (`[DOMAIN]`, `[PATH]`, `[IP]`, `[IPv6]`).
- **URLBERT [24]:** A domain-specific BERT pretrained on 3 billion URLs with five self-supervised objectives and a URL-specialized tokenizer.
- **BERT Base Uncased [14]:** The original bidirectional Transformer encoder pretrained with Masked Language Modeling and Next Sentence Prediction.
- **RoBERTa Base [41]:** An optimized BERT variant trained with dynamic masking, no Next Sentence Prediction, and 10× more data.
- **mmBERT Small / Base [42]:** Modern multilingual encoders pretrained on 3T+ tokens across 1,833 languages using Annealed Language Learning.
- **XLM-RoBERTa Base [43]:** A multilingual Transformer pretrained on 2.5 TB of CommonCrawl data spanning 100 languages.

Among these, TinyURLBERT, DomURLs_BERT, and URLBERT were pretrained specifically on URL corpora, while the remaining models were pretrained on general-purpose text. We included both to assess whether domain-specific pretraining provides an advantage for URL-only classification.

For all URL-only Transformer models, we used each model's default pretrained tokenizer without modification. The normalized URL was passed directly as input with a maximum sequence length of 128 tokens. Each tokenizer applied its own subword

segmentation strategy (e.g., WordPiece for BERT-based models, SentencePiece for XLM-RoBERTa).

3.4.2 URL + Content Models

The models in this group receive information from both the URL and the webpage content, testing whether this additional modality improves detection. We present them in order of increasing complexity.

Content as a Lightweight Signal:

We first evaluated two traditional machine learning models that incorporate content-derived information without requiring GPU resources:

- **XGBoost** [44]: A gradient-boosted decision tree classifier trained on the 21 hand-crafted features from Section 3.3, which encode structural and metadata signals from both the URL and the raw HTML document such as length, metadata presence, and content completeness.
- **TF-IDF + Logistic Regression**: A logistic regression classifier trained on TF-IDF vectors derived from concatenated URL and Markdown content, using byte-pair encoding (BPE) tokenization [45] at both word and character levels with concatenated feature vectors. This provides a lightweight, content-aware baseline.

Content-Aware Transformers:

We fine-tuned multilingual transformer models on the concatenation of the URL and the processed Markdown content. Only multilingual models were used because the dataset spans multiple languages. The models evaluated in this setting were: **XtremeDistil-l12-h384**, **mmBERT Small**, **mmBERT Base**, and **XLM-RoBERTa Base**. For these models, the URL and Markdown content were concatenated with a `[SEP]` delimiter (`{url} [SEP] {content}`) and truncated to 512 tokens, as preliminary experiments (Section 4.2) showed that longer context windows did not improve performance.

Hybrid Ensembles:

To leverage the complementary strengths of models operating on different input modalities, we constructed hybrid ensembles that combine predictions from the URL-only, content-aware transformer, and traditional machine learning models described above. The architecture is illustrated in Figure 5. Each base model produces a probability $P(\text{malicious})$ for every sample, and the hybrid ensemble aggregates these using one of the following strategies:

- **Hard voting**: Each model casts a binary vote ($P > 0.5 \Rightarrow$ malicious); the final prediction is the majority class.
- **Soft voting**: Predicted probabilities are averaged, with the final prediction thresholded at 0.5.

- **Weighted average**: Model-specific weights are optimized on the validation set to maximize accuracy, using SLSQP with 50 Dirichlet-sampled random restarts.
- **Stacking**: A meta-learner is trained on base-model probabilities from the validation set. We evaluated six meta-learners: Logistic Regression, LightGBM [46], CatBoost [47], Random Forest, XGBoost, and MLP.

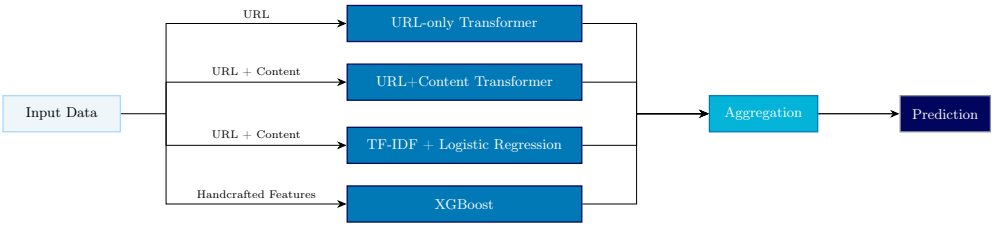

Fig. 5: Architecture of the hybrid ensemble. The aggregation layer combines individual probabilities to produce the final prediction. Not all configurations use all four models.

For each URL + content Transformer, we systematically varied which other models (ELECTRA Small, TF-IDF, XGBoost) were included and applied all nine aggregation strategies to each combination. From the results, we selected the best-performing configuration in each of the three parameter-count ranges (Small: <50M, Medium: 50–150M, Large: >150M):

- **Hybrid Ensemble Small** (47.1M parameters): XtremeDistil (URL + content) + ELECTRA Small (URL only) + TF-IDF + XGBoost.
- **Hybrid Ensemble Medium** (154.3M parameters): mmBERT Small (URL + content) + ELECTRA Small (URL only) + TF-IDF + XGBoost.
- **Hybrid Ensemble Large** (291.5M parameters): XLM-RoBERTa Base (URL + content) + ELECTRA Small (URL only).

3.5 Training

3.5.1 Hyperparameter Tuning

We performed a grid search on the 20,000-sample tuning subset (split 80:20, stratified by label) to select hyperparameters. Using XtremeDistil-l12-h384 on the URL + content task as a proxy for rapid iteration, we evaluated all combinations of learning rate $\in \{1\times10^{-5}, 2\times10^{-5}, 4\times10^{-5}\}$, batch size $\in \{16, 32, 64\}$, and warmup ratio $\in \{0.05, 0.1\}$ (18 configurations). The best configuration (learning rate 2×10^{-5}, batch size 32, warmup ratio 0.05) was applied to all BERT-based models for consistent and fair comparison. These values fall within the range recommended in [14] for fine-tuning BERT models and are therefore reasonable defaults across architectures. The same data subset was also used for XGBoost and TF-IDF hyperparameter tuning.

3.5.2 Training Procedure

All Transformer models were fine-tuned for binary classification by adding a two-class linear head on top of the pretrained encoder. We trained all models using the AdamW optimizer [48] with weight decay of 0.01, linear warmup scheduling, dropout of 0.1, and gradient clipping at 1.0, for up to 10 epochs with early stopping (patience of 2), selecting the checkpoint with the highest validation accuracy. All experiments were conducted on a single NVIDIA GeForce RTX 5090 GPU with 32 GB of GDDR7 VRAM.

3.6 Evaluation Metrics

We evaluated all models using four standard classification metrics. Because the dataset is balanced after undersampling, accuracy is a reliable aggregate measure. Additionally, we reported precision, recall, and F1-score to capture class-specific detection performance. Let TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively.

- **Accuracy:** Measures the overall correctness of the classifier:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{1}$$

- **Precision:** Measures the fraction of predicted malicious samples that are correct:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

- **Recall:** Measures the fraction of actual malicious samples that are found:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

- **F1-score:** The harmonic mean of precision and recall, balancing both metrics:

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

4 Experiments and Results

4.1 Results

All results in this section are reported on the held-out test set (68,954 samples). Table 2 summarizes the results of all benchmarked model architectures, organized by input modality.

Table 2: Performance of all models on the test set.

Model	Params (M)	Acc. (%)	Prec. (%)	Rec. (%)	F1 (%)
URL-Only Models					
URLBERT	109.5	96.04	96.67	95.36	96.01
TinyURLBERT	3.7	96.48	97.39	95.52	96.45
ALBERT Base v2	11.8	96.94	97.48	96.39	96.93
XtremeDistil-l12-h384	33.4	97.08	97.57	96.57	97.06
mmBERT Base	307.5	97.11	97.63	96.58	97.10
ELECTRA Small	13.5	97.14	97.58	96.66	97.12
mmBERT Small	140.6	97.18	97.43	**96.92**	97.18
DomURLs_BERT	109.5	97.31	97.82	96.78	97.30
BERT Base Uncased	109.5	97.37	97.82	96.90	97.36
RoBERTa Base	124.6	**97.38**	97.90	96.85	**97.37**
XLM-RoBERTa Base	278.0	**97.38**	**98.54**	96.19	97.35
URL + Content Models					
XGBoost (21 features)	0.06	95.22	95.72	94.68	95.20
TF-IDF + Log. Reg.	0.2	97.82	97.66	98.00	97.83
XtremeDistil-l12-h384	33.4	98.42	98.57	98.26	98.41
Hybrid Ensemble Small	47.1	98.83	99.08	98.57	98.83
mmBERT Small	140.6	98.94	99.00	**98.88**	98.94
mmBERT Base	307.5	98.94	99.29	98.58	98.93
XLM-RoBERTa Base	278.0	99.01	**99.36**	98.65	99.00
Hybrid Ensemble Medium	154.3	99.04	99.22	98.86	99.04
Hybrid Ensemble Large	291.5	**99.06**	99.31	98.80	**99.05**

4.1.1 URL-Only Models

RoBERTa Base and XLM-RoBERTa Base achieved the highest accuracy of 97.38%, closely followed by BERT Base Uncased at 97.37%. ELECTRA Small reached 97.14% with only 13.5M parameters, just 0.24 percentage points below the best result from models over 100M parameters, suggesting that URL-only classification has a low complexity ceiling where scaling model size offers little benefit. A consistent pattern across all URL-only models is that precision exceeds recall, indicating a tendency to miss malicious URLs rather than incorrectly flag safe ones. For instance, XLM-RoBERTa Base achieves the highest precision (98.54%) but the lowest recall (96.19%) among the top models.

4.1.2 URL + Content Models

Incorporating webpage content consistently and substantially improved detection across all model types. XGBoost, operating on 21 handcrafted features, reached 95.22%. TF-IDF + Logistic Regression achieved 97.82% accuracy, surpassing all URL-only transformer models despite being substantially simpler. Content-aware transformers showed even larger gains, with accuracy improvements of 1.34–1.83 percentage points, corresponding to error reductions between 45.9% (XtremeDistil: 2.92%→1.58%) and 63.3% (mmBERT Base: 2.89%→1.06%). XLM-RoBERTa Base achieved the highest individual accuracy of 99.01%, with precision of 99.36% and recall of 98.65%, substantially narrowing the precision–recall gap observed in URL-only models.

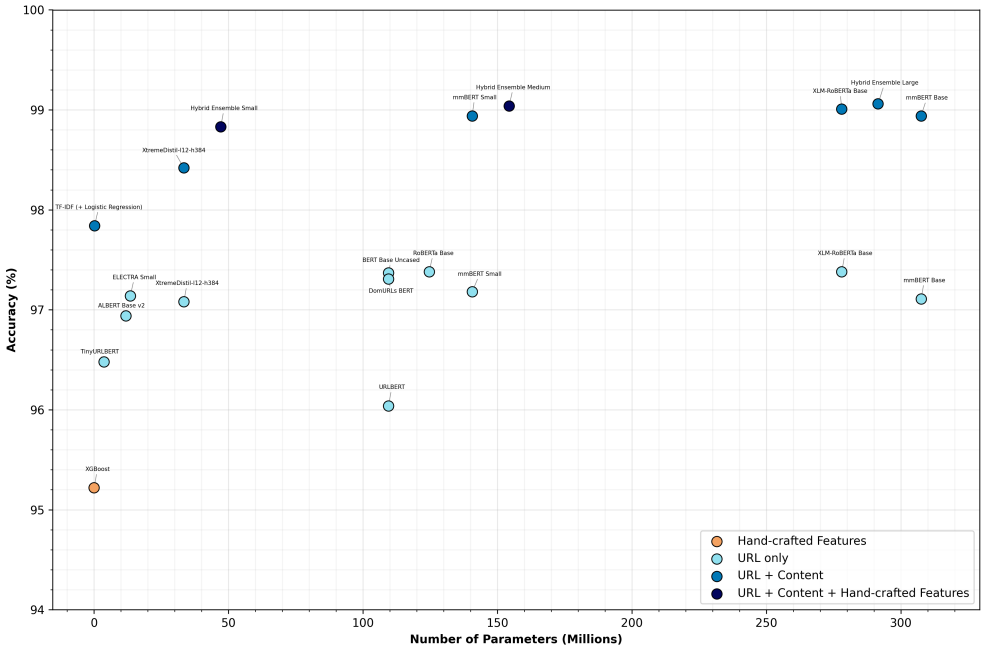

Fig. 6: Accuracy vs. parameter count across all model configurations.

The hybrid ensembles pushed performance further. The Large variant achieved 99.06% accuracy using Random Forest stacking over XLM-RoBERTa Base (URL + content) and ELECTRA Small (URL only), as well as reducing both error types to a false positive rate of 0.70% and false negative rate of 1.20% (Figure 8). Notably, this best configuration uses only two transformer models with no traditional ML components, suggesting that at sufficient capacity, transformers can implicitly learn the patterns that handcrafted features and TF-IDF capture explicitly. It outperforms the Medium and Small variants (99.04% and 98.83% via soft voting), which compensate for smaller transformers by incorporating TF-IDF and XGBoost predictions. The Small variant, with only 47.1M total parameters, still outperforms every individual model except XLM-RoBERTa, demonstrating that the hybrid ensemble strategy is effective even under tight computational constraints.

Figure 6 visualizes the relationship between model size and accuracy across all configurations, highlighting that the primary driver of performance is input modality rather than parameter count: URL + content models consistently outperform URL-only models regardless of size.

4.1.3 Comparison with Prior Work

To evaluate generalizability, we retrained the best configuration (Hybrid Ensemble Large) from scratch on datasets used by four recent studies, using the same dataset sizes and split ratios as the original papers. The record filtering step (Section 3.2) was not applied to ensure fairness. Where authors did not use a validation set, we

further split training data into training and validation subsets (80:20, stratified) for early stopping and meta-learner fitting. Results are consolidated in Table 3.

On the MTLP dataset [34] (65,595 samples), our approach outperformed MultiText-LP (a fusion of MLP and NLP branches over tabular and textual HTML features) by 2.10 points in accuracy and 2.38 in F1, suggesting that a unified transformer encoder over rendered text is more effective than fusing separate models over raw HTML features.

On Aljofey's dataset [29] (60,252 webpages), the improvement was 8.28 points over Aljofey et al.'s character-level TF-IDF approach and 7.70 over MultiText-LP, suggesting that end-to-end transformer-based text understanding captures phishing signals more effectively than prior feature-engineering methods.

On the full Mendeley dataset [33] (79,847 webpages), our model surpassed GraphSAGE (a GNN trained on DOM-tree graphs with structural HTML reduction) [28] by 1.96 points in accuracy and 2.86 in F1, indicating that rendered textual content carries stronger semantic signals than structural HTML DOM features.

On a smaller Mendeley subset used by Qasim and Flayh (a Random Forest classifier with URL-based feature selection) [21] (8,000 samples), our approach outperformed their best model by 0.78 points despite a training set of only 5,120 samples, demonstrating strong generalization even in low-data regimes.

Table 3: Comparison with prior work across four external datasets.

Dataset	Model	Acc. (%)	Prec. (%)	Rec. (%)	F1 (%)
MTLP	MultiText-LP [34]	97.18	–	–	96.80
	Hybrid Ensemble Large	**99.28**	99.30	99.05	**99.18**
Aljofey	Aljofey et al. [29]	89.00	88.20	87.68	87.94
	MultiText-LP [34]	89.58	89.61	87.03	88.17
	Hybrid Ensemble Large	**97.28**	**98.41**	**95.53**	**96.95**
Mendeley (Full)	GraphSAGE [28]	96.87	95.66	95.48	95.57
	Hybrid Ensemble Large	**98.83**	**98.91**	**97.97**	**98.43**
Mendeley (Subset)	Random Forest [21]	97.60	98.20	97.00	97.60
	Hybrid Ensemble Large	**98.38**	**99.11**	**97.62**	**98.36**

4.2 Ablation Studies

We conducted three ablation studies to quantify the contribution of key design choices in our pipeline. Experiments were performed on the 20,000-sample tuning subset using mmBERT Small as a representative multilingual transformer due to its balance between performance and computational efficiency. In each experiment, all settings except the variable under study were held constant.

1. **Context Window Length:** We compared maximum sequence lengths of 512 and 2,048 tokens for the URL + content task. The 512-token configuration

achieved 97.40% validation accuracy versus 97.35% for 2,048 tokens, indicating that the first few hundred tokens are sufficient for the model to determine intent and that excessively long contexts introduce noise without improving classification. All subsequent experiments used a 512-token context window.

2. **Markdown vs. Plain Text:** The Markdown variant achieved 97.40% validation accuracy compared to 96.95% for plain text (with all Markdown formatting stripped). The 0.45 percentage point improvement suggests that lightweight structural cues retained by Markdown, such as headings, bold text, and list markers, provide useful signals for distinguishing safe from malicious content.
3. **URL-Only vs. Content-Only:** The URL-only model achieved 94.48% accuracy, while the content-only model (excluding the URL) achieved 94.45%. The combined URL + content model reached 97.40%, a gain of approximately 3 percentage points over either individual modality (Figure 7). The near-identical performance of URL-only and content-only models indicates that these two sources carry largely complementary information: a malicious site can replicate the content of a legitimate page but cannot fake its URL, and conversely, an attacker may use a clean-looking URL while hosting deceptive content.

Table 4 summarizes all the results of the ablation studies.

Table 4: Ablation study results

Category	Configuration	Accuracy (%)
Context Length	2,048 tokens	97.35
	512 tokens	**97.40**
Input Format	Plain Text	96.95
	Markdown	**97.40**
Input Modality	URL only	94.48
	Content only	94.45
	URL + Content	**97.40**

4.3 Case Studies and Error Analysis

To complement the aggregate metrics, we examine individual predictions from the best-performing Hybrid Ensemble Large on the test set with several case studies. The content field is truncated for brevity.

4.3.1 Content as a Discriminative Signal

A key motivation for content-aware detection is that URLs alone can be insufficient. The following example illustrates this clearly:
URL: `https://sidime.com/realestate-prolux-items-58439`
Content:

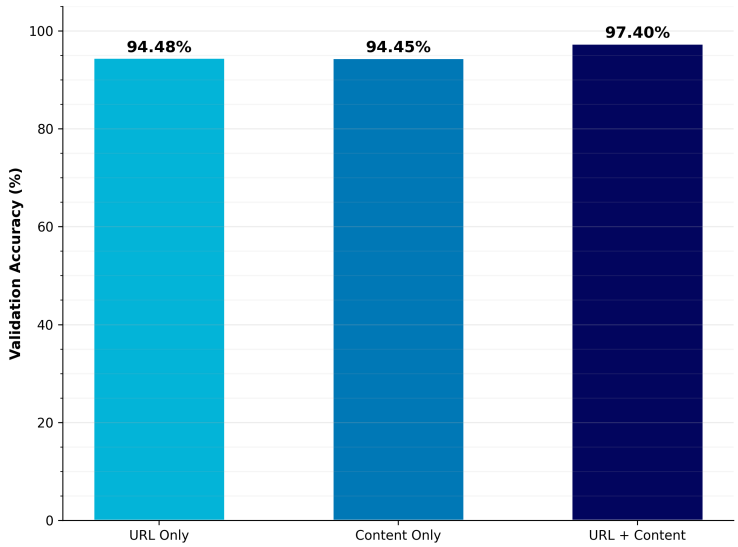

Fig. 7: Impact of input modality on detection accuracy.

```
title: Facebook | Confirm identity
---
#  Facebook

To protect the privacy of our community please confirm
your identity using your **Facebook** account with **Login** .

Email or Phone:

Password:

Log In

[Forgot Password?]

[Create New Account]
```

- **ELECTRA Small (URL only):** $P(\text{malicious}) = 0.0004$
- **RoBERTa Base (URL only):** $P(\text{malicious}) = 0.0002$
- **XLM-RoBERTa Base (URL + content):** $P(\text{malicious}) = 1.0000$
- **Hybrid Ensemble Large:** $P(\text{malicious}) = 0.9483$

The domain `sidime.com` uses a clean `.com` TLD with a path resembling a real estate listing, offering no obvious lexical red flags. Both URL-only models assign extremely low malicious probabilities (RoBERTa Base at 0.02% and ELECTRA Small at 0.04%),

confirming that the URL alone provides no signal of malicious intent. However, the page content reveals a Facebook login clone, a classic credential-harvesting phishing page hosted on an unrelated domain. XLM-RoBERTa Base, with access to both URL and content, identifies this as malicious with near-certainty ($P = 1.0000$), while the hybrid ensemble assigns 94.8% confidence.

4.3.2 Multilingual Detection

Our dataset spans multiple languages, and the hybrid ensemble demonstrates strong cross-lingual capability. Below, we present four correctly classified non-English examples from the test set.

1. **Vietnamese investment scam** (malicious, $P = 0.983$):
   URL: `https://quickstar.vip`
   Content:

   title: QuickStar
   —
   [logo]
   Đăng ký
   Đăng nhập
   Hãy tự tin đầu tư với QuickStar rồi theo dõi lợi nhuận của bạn tăng lên.
   QuickStar cung cấp các công cụ tài chính chuyên nghiệp để hỗ trợ bạn đưa ra các quyết định đầu tư hiệu quả khi giao dịch.
   Bắt đầu
   ...

   (Translation: "Sign up. Sign in. Invest confidently with QuickStar and watch your profits grow. QuickStar provides professional financial tools to help you make effective investment decisions when trading. Get started.")

   The website uses the '.vip' domain, which is suspicious in itself. Additionally, the language used ("invest confidently" and "watch your profits grow") is overly aggressive and seems out of place for legitimate financial services.
2. **Chinese fake Telegram download** (malicious, $P = 0.997$):
   URL: `http://telegramrc.com`
   Content:

   title: 电报Telegram下载|电报Telegram中文版下载官网
   —
   - [首页]
   - [FAQ]
   - [应用下载]
   ...
   Telegram可在多种设备上使用
   [windows中文版下载] [Mac中文版下载]
   [安卓手机中文版下载]
   [苹果手机中文版下载]

   (Translation: "Telegram Download | Telegram Chinese Version Official Download. Home. FAQ. App Download. ... Telegram is available on multiple devices. Windows/Mac/Android/iPhone Chinese download.")

The decisive signal is a non-official domain hosting an app download page for a major, well-known platform, combined with the HTTP protocol and absence of any organizational metadata.

3. **French travel blog** (safe, $P = 0.011$):
   URL: `https://avant-de-partir.fr`
   Content:

```
title: Avant de partir en voyage...Nos conseils
---
[Skip to content]

- [Dans le monde]
- [En France]
- [Voyages et découvertes]
- [Blog et conseils]
- [Contact]

Rechercher :

# Avant de partir en voyage...Nos conseils

Croisière tour du monde : pour un voyage inoubliable !
...
```

   (Translation: "Before going on a trip... Our advice. Around the world. In France. Travel and discoveries. Blog and tips. Contact. ... World cruise: for an unforgettable trip!")

   The model correctly identifies this as safe based on the coherent hierarchical navigation structure, topically consistent content, and absence of credential-capture or urgency patterns.

4. **Japanese Excel tutorial** (safe, $P = 0.007$):
   URL: `https://excel-fighter.net/excel-copy`
   Content:

```
title: Excelでセルを簡単にコピーする3つの方法
—
- [ホーム]
- [Excel]
# Excelでセルを簡単にコピーする3つの方法
...
エクセルを操作する中で、頻繁に利用する機能の一つが「コピー」です。
```

   (Translation: "3 Easy Ways to Copy Cells in Excel. Home. Excel. ... One of the most frequently used functions when working with Excel is 'Copy'.")

   The model correctly identifies this as safe based on the tutorial-style heading, consistent educational content body, and complete absence of any form elements or call-to-action patterns associated with phishing.

In all four cases, the model assigns confident and correct predictions despite the non-English text, demonstrating that XLM-RoBERTa's multilingual pretraining transfers effectively to this task.

4.3.3 Error Analysis

We analyze representative false positives and false negatives to characterize the model's failure modes. The confusion matrix is shown in Figure 8.

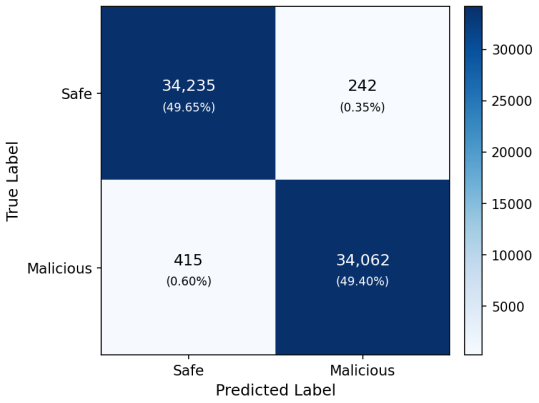

Fig. 8: Confusion matrix of the Hybrid Ensemble Large on the test set.

False positives (242 cases, 0.70% of safe sites):

False positives arise when legitimate websites have unusual-looking URLs combined with sensitive contents, such as login forms or payment flows, which are surface-level patterns the model has learned to associate with phishing.

1. LogMeIn login page ($P = 0.997$):
   URL:
   `https://wpwgteqdaauzkuclsqut.lmi-app25-18.logmein.com/openclient.html`
   Content:

```
title: LogMeIn Accounts
---
# Log in

or [sign up]

LogMeIn ID:

[Forgot your password?]

I trust this device. Keep me logged in.
```

```
Log in [Back]

Copyright © 2003-2025 LogMeIn, Inc. All rights reserved.
[Legal] [Privacy] [Sales] Questions: +1.866.478.1805
```

A legitimate remote access login page, but the machine-generated subdomain (`wpwgteqdaauzkuclsqut`) combined with the login form closely matches credential-harvesting patterns.

2. teamLab Borderless ticket page ($P = 0.999$):
   URL: `https://borderless-azabudai.ticket.teamlab.art`
   Content:

```
title: 公式チケットサイト | 森ビル デジタルアート ミュージアム：エプソン チーム ラボボーダレス
—
購入内容選択
お客様情報
お支払い情報
購入内容確認
...
- エントランスパス
- フレキシブルパス
- 大人 ¥3,600～¥4,400
```

(Translation: "Official Ticket Site | teamLab Borderless. Select purchase. Customer information. Payment information. . . . Entrance pass. Flexible pass. Adult ¥3,600–¥4,400.") A legitimate art museum ticketing page, but the uncommon TLD (`.art`), nested subdomains, and payment-related content trigger the same patterns.

False negatives (415 cases, 1.20% of malicious sites):

False negatives occur when both the URL and content appear legitimate, and the malicious nature of the service cannot be determined from the inputs alone.

1. Fake Best Buy clone ($P = 0.153$):
   URL: `https://web.bestbuytopsales.com`
   Content:

```
title: Shop Best Buy for electronics, computers, appliances,
  cell phones, video games & more new tech. In-store pickup
  & free 2-day shipping on thousands of items.
description: Web site created using create-react-app
---
![logo]

### Log in

Phone
```

```
password

Log in

or

Register

[Forgot password]
```

The title, description, and layout are copied verbatim from the real Best Buy website. The model has no knowledge that "Best Buy" is a real brand or that `bestbuy.com` is its official domain, so it cannot recognize `bestbuytopsales.com` as an impersonation.

2. Fraudulent cosmetics store ($P = 0.002$):
   URL: `https://glossvita.com/cart/checkout`
   Content:

```
title: Cart - Glossvita.com
---
Please Wait...

- [Login]
- [Home]
- [All Collection]
  - [All]
  - [Makeup]
  - [Makeup and correctors]
  - [Compact powders]
...
###### SHOPPING CART
###### CHECK OUT DETAIL
###### ORDER COMPLETE

No products in shopping cart
```

The URL is clean and the content features a detailed product catalog, navigation, and checkout flow indistinguishable from a real online store. The fraudulent nature of this site lies in its business practices, not its presentation.

Among the 415 false negatives, three patterns recur most frequently: high-quality brand impersonation pages where content is copied verbatim from a legitimate site (as in the Best Buy example), fraudulent storefronts with complete but fictitious product catalogs indistinguishable from real e-commerce pages, and content-sparse redirect pages where insufficient rendered text is available to form a confident judgment.

Reducing false negatives likely requires external knowledge such as brand-to-domain mappings derived from search engine results or domain reputation databases, enabling the model to recognize impersonation attempts that are indistinguishable from the URL and content alone.

4.4 Discussion

The most striking finding is that a simple, GPU-free TF-IDF + Logistic Regression model with content access (97.82%) surpassed every URL-only transformer, including models with over 300M parameters. Its dual word-level and character-level BPE tokenizers, each with 100K vocabulary terms and (1,2)-gram TF-IDF, create a 200K-dimensional feature space that captures both phishing-specific phrases (e.g., "Verify your", "PayPal account", "be locked") and structural URL patterns from the byte-level tokenizer (e.g., "amaz 0", ". tk", "/ signin" from a spoofed Amazon URL). Notably, this is the only model where recall (98.00%) exceeds precision (97.66%), suggesting that content-level term patterns are especially effective at catching malicious pages that URL analysis alone would miss.

At the same time, model scale shows diminishing returns. URL-only models are fundamentally limited by the information in a short string that attackers deliberately design to look benign. TinyURLBERT (3.7M parameters) trails RoBERTa Base (124.6M) by just 0.90 percentage points despite being 33× smaller, achieving 96.48% with a domain-specific vocabulary of only approximately 400 tokens and knowledge distillation from a larger teacher. In the URL + content setting, mmBERT Small (140.6M) matches mmBERT Base (307.5M) at 98.94%. Beyond a certain capacity threshold, additional parameters yield no measurable benefit.

Notably, URL-specific pretrained models did not outperform their general-purpose counterparts. URLBERT (96.04%) and DomURLs_BERT (97.31%), both pretrained on URL corpora, fell short of BERT Base (97.37%) and RoBERTa Base (97.38%). URLBERT's five pretraining objectives, including adversarial augmentations (FGSM-based perturbations and KL-divergence regularization) designed for robustness against noisy URL components, provide diminishing returns on our preprocessed and normalized URLs, where such noise has already been removed. URLBERT's tokenizer was trained exclusively on English-language URLs, while DomURLs_BERT's tokenizer, though trained on a corpus that includes multilingual sources (mC4), remains dominated by English-centric web crawls. In both cases, the tokenizers are optimized for URL subword patterns rather than the diverse character sets found in internationalized domain names. In contrast, general-purpose models like XLM-RoBERTa, pretrained on 100 languages, handle these naturally. DomURLs_BERT fared better thanks to its structural tokens (`[DOMAIN]`, `[PATH]`), but its strongest published results were on tasks like DGA botnet detection and DNS tunneling, where domain names carry strong statistical patterns that differ markedly from the subtler lexical cues in phishing URLs.

These results align with the ablation study (Section 4.2), which shows that URL and content are comparably strong indicators of malicious intent, each capturing distinct aspects: the URL reveals structural artifacts that are always available and fast to process but easily spoofed, while content captures the semantic intent of what users actually see, which is harder to fake convincingly across diverse attack types. Combining them yields a nearly 3 percentage point improvement, as each compensates for the other's blind spots. Handcrafted features further augment this with explicit, interpretable signals (metadata completeness, link structure, CAPTCHA presence) that neither neural approach encodes directly.

Together, these findings carry a practical implication for how malicious website detection systems should be designed: the bottleneck is not model capacity but input scope. A system that fetches and preprocesses page content at classification time, even using a lightweight TF-IDF pipeline, will outperform arbitrarily large models that examine only the URL. For practitioners, this suggests that the engineering investment in content retrieval and preprocessing yields greater security value than scaling up the model. For researchers, it suggests that future work should focus on handling the failure modes identified in Section 4.3.3, particularly brand-to-domain grounding and sparse-content pages, rather than on architectural refinements to URL encoders.

4.5 Limitations

While our approach achieves strong results, its limitations should be acknowledged:

1. **Off-site malicious activity:** Some malicious websites direct users to complete harmful actions through external channels (email, phone, social media). Detection based solely on URL and page content is insufficient in such cases.
2. **Ambiguous labeling boundaries:** The distinction between safe and malicious is not always clear-cut. Gambling and adult content websites are legal in some jurisdictions but illegal in others, introducing label noise that can affect both training and evaluation. This ambiguity is particularly acute for datasets like ChongLuaDao, where community-driven labeling may reflect regional legal norms rather than universal definitions of malicious intent.
3. **Selection bias in datasets:** The malicious websites in our datasets were previously detected and reported to platforms such as PhishTank and OpenPhish. The training data may therefore underrepresent more sophisticated or novel attacks that have evaded detection.
4. **Auto-generated subdomain ambiguity**: Legitimate services increasingly use machine-generated subdomains for session routing and load balancing, producing URL patterns that are superficially indistinguishable from attacker-registered domains. Without access to domain registration metadata or organizational subdomain ownership records, content-aware models are structurally prone to false positives.

5 Conclusion and Future Work

Malicious website detection has traditionally focused on URL-level signals, which struggle to reliably capture malicious intent. To address this, we presented a content-aware approach to malicious website detection that pairs URL analysis with rendered webpage text using multilingual BERT-based models and hybrid ensembles. On a dataset of 689,556 webpages, incorporating content improved individual transformer accuracy by up to 1.83 percentage points, and case study analysis confirmed that content catches malicious sites that URL-only models miss entirely. The best-performing configuration, a hybrid ensemble combining XLM-RoBERTa Base with ELECTRA Small via Random Forest stacking, achieved 99.06% accuracy on our test set. This approach also outperformed prior methods on four external datasets by 0.78 to

7.70 percentage points, demonstrating strong generalizability. These results suggest that content-aware detection can meaningfully combat malicious websites in practical deployments such as browser extensions and enterprise security gateways.

Several directions could extend this work:

- **Visual analysis:** Incorporating screenshot-based models could help detect knock-off websites that mimic content but not design quality.
- **External signals:** Integrating data such as user reviews, website traffic, or domain reputation data could improve coverage and detection accuracy.
- **Explainability:** The case studies demonstrated that probability scores alongside representative content excerpts already provide a basic form of human-interpretable evidence. Extending this with attention visualization or generative explanation models could surface the specific phrases, structural patterns, or metadata inconsistencies driving each classification, increasing user trust and helping security analysts understand emerging threat patterns.

Funding Declaration

No, this research did not receive funding.

References

[1] IBM Security and Ponemon Institute: Cost of a Data Breach Report 2025. Technical report, IBM Corporation (July 2025). https://www.ibm.com/reports/data-breach

[2] Anti-Phishing Working Group: Phishing Activity Trends Report: 1st Quarter 2025. Technical report, APWG (July 2025). https://docs.apwg.org/reports/apwg_trends_report_q1_2025.pdf

[3] Muthazhagu, V.H., Surendiran, B.: Exploring the Role of AI in Web Design and Development: A Voyage through Automated Code Generation. In: Proceedings of the International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE), pp. 1–8 (2024). https://doi.org/10.1109/IITCEE59897.2024.10467409

[4] Lee, K., Lim, K., Kim, H., Kwon, Y., Kim, D.: 7 Days Later: Analyzing Phishing-Site Lifespan After Detection. In: Proceedings of the ACM Web Conference (WWW '25), pp. 945–956 (2025). https://doi.org/10.1145/3696410.3714678

[5] Fernando, M., Mahmood, A.N., Chowdhury, M.J.M.: PhishLex: A Proactive Zero-Day Phishing Defence Mechanism using URL Lexical Features. In: Proceedings of the NDSS Symposium (2022). https://doi.org/10.2139/ssrn.5205908

[6] Elsheh, M.M., Swayeb, K.: Phishing Website Detection Using a Hybrid Approach

Based on Support Vector Machine and Ant Colony Optimization. In: IEEE MI-STA, pp. 402–406 (2023). https://doi.org/10.1109/MI-STA57575.2023.10169464

[7] Abad, S., Gholamy, H., Aslani, M.: Classification of Malicious URLs Using Machine Learning. Sensors **23**(18), 7760 (2023) https://doi.org/10.3390/s23187760

[8] Ashok, A., Rathis, D., Raghavendra, R., Umadevi, V.: A Comparative Analysis of Traditional Machine Learning, Deep Learning and Boosting Algorithms on Phishing URL Detection. In: IEEE CVMI (2024). https://doi.org/10.1109/CVMI61877.2024.10782525

[9] Sankaranarayanan, S., Sivachandran, A.T., Anis, Hasikin, K., Rahman, A.: An Ensemble Classification Method Based on Machine Learning Models for Malicious URLs. PLoS ONE **19** (2024) https://doi.org/10.1371/journal.pone.0302196

[10] Jyothi, B.S., Akshaya, M., Anjum, K., Bhavana, A., Sreemukha, K.: URL Based Phishing Detection using Machine Learning. In: Proceedings of the 4th International Conference on Information Technology, Civil Innovation, Science, and Management (ICITSM) (2025). https://doi.org/10.4108/eai.28-4-2025.2358166

[11] Sultana, R., Rahman, M.A., Khan, M.I.: Hybrid Model Based Phishing Websites Detection Using Deep Learning Technique. In: IEEE ICCIT (2023). https://doi.org/10.1109/ICCIT60459.2023.10441639

[12] Xiao, X., Zhang, D., Hu, G., Jiang, Y., Xia, S.: CNN-MHSA: A Convolutional Neural Network and Multi-Head Self-Attention Combined Approach for Detecting Phishing Websites. Neural Networks **125**, 303–312 (2020) https://doi.org/10.1016/j.neunet.2020.02.013

[13] Porcu, V., Havlínová, A.: Past vs. Present: Key Differences Between Conventional Machine Learning and Transformer Architectures. Advances in Nonlinear Variational Inequalities **28**(2s), 2537 (2025) https://doi.org/10.52783/anvi.v28.2537

[14] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Proceedings of NAACL-HLT 2019, pp. 4171–4186 (2019). https://doi.org/10.18653/v1/N19-1423

[15] Su, M.-Y., Su, K.-L.: BERT-Based Approaches to Identifying Malicious URLs. Sensors **23**(20), 8499 (2023) https://doi.org/10.3390/s23208499

[16] Patra, C., Giri, D., Maitra, T., Kundu, B.: A Comparative Study on Detecting Phishing URLs Leveraging Pre-trained BERT Variants. In: 2024 6th International Conference on Computational Intelligence and Networks (CINE), pp. 5–13 (2025). https://doi.org/10.1109/CINE63708.2024.10881521

[17] Do, N.Q., Selamat, A., Fujita, H., Krejcar, O.: An Integrated Model Based on Deep Learning Classifiers and Pre-trained Transformer for Phishing URL Detection. Future Generation Computer Systems **161**, 269–285 (2024) https://doi.org/10.1016/j.future.2024.06.031

[18] Yu, B., Tang, F., Ergu, D., Zeng, R., Ma, B., Liu, F.: Efficient Classification of Malicious URLs: M-BERT—A Modified BERT Variant for Enhanced Semantic Understanding. IEEE Access **12**, 13453–13468 (2024) https://doi.org/10.1109/ACCESS.2024.3357095

[19] Opara, C., Chen, Y., Wei, B.: Look before you leap: Detecting phishing web pages by exploiting raw URL and HTML characteristics. Expert Systems with Applications **236**, 121183 (2024) https://doi.org/10.1016/j.eswa.2023.121183

[20] Tian, Y., Yumin, Z., Jia, Y., Sun, J., Wang, Y.: WebGuard++:Interpretable Malicious URL Detection via Bidirectional Fusion of HTML Subgraphs and Multi-Scale Convolutional BERT. ArXiv **abs/2506.19356** (2025)

[21] Qasim, M.-A.A.A.H., Flayh, N.A.: Enhancing Phishing Website Detection via Feature Selection in URL-Based Analysis. Informatica **47**(9) (2023) https://doi.org/10.31449/inf.v47i9.5177

[22] Yeasmin, M.N., Refat, M.A.R., Singh, B.C., Alom, Z., Aung, Z., Azim, M.: EnLeM: ensemble learning-based model to detect phishing websites. EURASIP Journal on Information Security **2026**(1), 1 (2025)

[23] Liu, R., Wang, Y., Guo, Z., Xu, H., Qin, Z., Ma, W., Zhang, F.: TransURL: Improving malicious URL detection with multi-layer Transformer encoding and multi-scale pyramid features. Computer Networks **253**, 110707 (2024) https://doi.org/10.1016/j.comnet.2024.110707

[24] Li, Y., Liu, Y., Li, P., Jia, Y., Wang, Y.: Continuous Multi-Task Pre-training for Malicious URL Detection and Webpage Classification (2025). https://arxiv.org/abs/2402.11495

[25] El Mahdaouy, A., Lamsiyah, S., Janati Idrissi, M., Alami, H., Yartaoui, Z., Berrada, I.: DomURLs_BERT: Pre-trained BERT-based Model for Malicious Domains and URLs Detection and Classification. Journal of Network and Systems Management **34**(2), 36 (2026) https://doi.org/10.1007/s10922-025-10010-9

[26] Feng, J., Qiao, Y., Ye, O., Zhang, Y.: Detecting phishing webpages via homology analysis of webpage structure. PeerJ Comput. Sci. **8**(e868), 868 (2022)

[27] Bilot, T., Geis, G., Hammi, B.: PhishGNN: A Phishing Website Detection Framework using Graph Neural Networks. In: Proceedings of the 19th International Conference on Security and Cryptography - Volume 1: SECRYPT,, pp. 428–435. SciTePress, ??? (2022). https://doi.org/10.5220/0011328600003283 . INSTICC

[28] Hofmans, W., Wei, W., Vanneste, S., Mets, K.: Towards Efficient GNN-Based Phishing Detection from HTML Source Code. In: The 37th Benelux Conference on Artificial Intelligence and the 34th Belgian Dutch Conference on Machine Learning (2025). https://openreview.net/forum?id=tJA55axTtA

[29] Aljofey, A., Jiang, Q., Rasool, A., Chen, H., Liu, W., Qu, Q., Wang, Y.: An effective detection approach for phishing websites using URL and HTML features. Scientific Reports **12**(1), 8842 (2022) https://doi.org/10.1038/s41598-022-10841-5

[30] Hamadouche, S., Boudraa, O., Gasmi, M.: Combining Lexical, Host, and Content-based features for Phishing Websites detection using Machine Learning Models. EAI Endorsed Transactions on Scalable Information Systems **11**(6) (2024) https://doi.org/10.4108/eetsis.4421

[31] Yang, T., Sun, J.: A hybrid ensemble deep learning framework with novel meta-heuristic optimization for scalable malicious website detection. Scientific Reports **15**(1), 44630 (2025) https://doi.org/10.1038/s41598-025-33695-z

[32] Dalton, T., Gowda, H., Rao, G., Pargi, S., Hadj Khodabakhshi, A., Rombs, J., Jou, S., Marwah, M.: PhreshPhish: A Real-World, High-Quality, Large-Scale Phishing Website Dataset and Benchmark. arXiv preprint arXiv:2507.10854 (2025) https://doi.org/10.48550/arXiv.2507.10854

[33] Ariyadasa, S., Fernando, S., Fernando, S.: Phishing Websites Dataset. https://doi.org/10.17632/n96ncsr5g4.1

[34] Çolhak, F., Ecevit, M.I., Uçar, B.E., Creutzburg, R., Dağ, H.: Phishing Website Detection Through Multi-model Analysis of HTML Content. In: Proceedings of International Conference on Theoretical and Applied Computing, pp. 171–184 (2025)

[35] ChongLuaDao: ChongLuaDao Anti-Scam Platform. https://chongluadao.vn

[36] Kaufmann, J.: html-to-markdown: Convert HTML to Markdown. Version 2. MIT License. https://github.com/JohannesKaufmann/html-to-markdown

[37] CrabInHoney: URLBERT Tiny v4: Phishing URL Classifier. Hugging Face (2024). https://huggingface.co/CrabInHoney/urlbert-tiny-v4-phishing-classifier

[38] Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R.: ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. CoRR **abs/1909.11942** (2019) arXiv:1909.11942

[39] Clark, K., Luong, M., Le, Q.V., Manning, C.D.: ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. CoRR **abs/2003.10555** (2020) arXiv:2003.10555

[40] Mukherjee, S., Awadallah, A.H., Gao, J.: XtremeDistilTransformers: Task Transfer for Task-agnostic Distillation (2021)

[41] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: RoBERTa: A Robustly Optimized BERT Pretraining Approach. CoRR **abs/1907.11692** (2019) arXiv:1907.11692

[42] Marone, M., Weller, O., Fleshman, W., Yang, E., Lawrie, D., Durme, B.V.: mmBERT: A Modern Multilingual Encoder with Annealed Language Learning (2025). https://arxiv.org/abs/2509.06888

[43] Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., Stoyanov, V.: Unsupervised Cross-lingual Representation Learning at Scale (2020). https://arxiv.org/abs/1911.02116

[44] Chen, T., Guestrin, C.: XGBoost: A Scalable Tree Boosting System. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794 (2016). https://doi.org/10.1145/2939672.2939785

[45] Sennrich, R., Haddow, B., Birch, A.: Neural Machine Translation of Rare Words with Subword Units. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1715–1725. Association for Computational Linguistics, ??? (2016). https://doi.org/10.18653/v1/P16-1162

[46] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.-Y.: LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In: Advances in Neural Information Processing Systems, vol. 30. Curran Associates, Inc., ??? (2017)

[47] Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A.V., Gulin, A.: CatBoost: Unbiased Boosting with Categorical Features (2019). https://arxiv.org/abs/1706.09516

[48] Loshchilov, I., Hutter, F.: Decoupled Weight Decay Regularization (2019). https://arxiv.org/abs/1711.05101