Towards Accurate and Optimized Booter Website Classification: Evaluating AI Models for Law Enforcement

NARENDRA SETTY, University of Twente, The Netherlands

The proliferation of booter websites offering DDoS-for-hire services poses a significant cybersecurity threat by enabling malicious actors to disrupt online services with minimal technical expertise. This problem is compelling due to the increasing sophistication of these platforms and the limitations of existing detection methods, which rely on predefined feature sets and often overlook complex semantic patterns or visual elements. My solution evaluates text-based and multimodal large language models (LLMs), achieving high accuracy (F1-score and accuracy of 100%) through optimized prompt engineering and semantic text analysis, surpassing traditional approaches. Additionally, the study leverages LLMs to analyze archived snapshots of booter websites that have been taken down, expanding the dataset and improving detection robustness. These findings provide law enforcement with practical, cost-effective guidelines for implementing scalable and reliable booter detection systems, enhancing their ability to combat DDoS threats efficiently. [1].

1 INTRODUCTION

The proliferation of "booter" website platforms offering Distributed Denial-of-Service (DDoS) attacks for hire has emerged as a critical cybersecurity threat, enabling malicious actors to disrupt online services, inflict economic damage, and compromise digital infrastructure [1]. DDoS attacks overwhelm target systems with excessive traffic, rendering them inaccessible to legitimate users, while booter services democratize access to such attacks by leasing botnets (networks of compromised devices) to individuals with limited technical expertise [2]. A notorious example is the 2016 case of vDOS, a prominent DDoS-for-hire service, which orchestrated over 2 million DDoS attacks between 2012 and 2016, knocking countless websites and internet users offline. In just four months from April to July 2016, vDOS launched attacks totaling 277 million seconds of downtime, equivalent to approximately 8.81 years of cumulative disruption. These attacks, capable of generating up to 50 gigabits per second (Gbps) of traffic, crippled websites not equipped with robust anti-DDoS protections, affecting businesses, gaming platforms, and critical services worldwide [8].

The growing sophistication of booters, including automated attack workflows and evasion tactics, underscores the urgency of addressing this challenge [2]. Manual detection methods, such as creating accounts to test each suspected site, are highly labor-intensive, as evidenced by recent FBI operations targeting DDoS-for-hire services [9]. Despite law enforcement takedowns, many of the seized booter sites were found to return within a median of just one day, highlighting how quickly these services can re-emerge [3].

Current solutions for detecting booter websites, while innovative, exhibit significant limitations in their approaches to identifying these malicious platforms. Santanna et al. (2016) developed a methodology that relies on analyzing a limited set of structural and metadata characteristics, such as domain age, WHOIS privacy, and the presence of DDoS protection services, without examining the textual content of the websites themselves [4]. In contrast, Zhang et al. (2018) focus exclusively on the textual content of webpages using a bag-of-words model and feature selection algorithms, yet is unable to to capture nuanced semantic patterns or contextual relationships within the text [5]. Neither approach employs LLMs to analyze textual content, which could enhance efficiency by identifying complex linguistic patterns [6]. Neither method accounts for the time taken for the classification process, which is critical for real-time detection and response. Additionally, neither method analyzes visual elements of webpages, such as images and buttons containing text that cannot be captured through standard web scraping techniques, which could provide critical indicators of malicious intent, as demonstrated by Lee et al. (2024) [7]. These gaps highlight the need for more comprehensive and advanced AI-driven methods to improve the accuracy and robustness of booter detection for law enforcement.

To address the limitations of existing booter website detection methods, this study proposes a comprehensive evaluation of AI-driven techniques to identify the most accurate and robust approach for classifying booter websites. The study will investigate various types of Large Language Models (LLMs) for semantic text analysis and visual element analysis. The models will be evaluated using a dataset of known booter and benign websites, with performance metrics including precision, recall, F1-score, false positive rate and processing time to quantify their effectiveness and minimize misclassification of legitimate platforms. To guide this research, the following research questions are proposed:

- RQ1: What are the critical features of booter websites (like text content, visual elements, URL structure) and the AI models best suited for leveraging these features in detection tasks?
- RQ2: How can various types of LLMs, including those for semantic text analysis and visual element analysis, be designed and compared to identify the most accurate approach for classifying booter websites?
- RQ3: How accurate are the proposed AI techniques in classifying booter websites compared to existing methods?

[1] sourcecode available at: https://github.com/Night-Swan/Research-Project

Author's address: Narendra Setty, n.setty@student.utwente.nl, University of Twente, P.O. Box 217, Enschede, The Netherlands, 7500AE.

TScIT 43, July 4, 2025, Enschede, The Netherlands

This study aims to make the following contributions to the field of booter website detection:

- Identify critical features of booter websites, including textual content, URL structure, and metadata, and evaluate which machine learning models are best suited to leverage these features for accurate detection.
- Design and implement a range of AI-driven detection methods, including various types of Large Language Models (LLMs) for semantic text analysis (extracted from website content) and visual element analysis (via screenshots of webpages). Analyze the strengths and weaknesses of each method in the context of booter website detection.
- Conduct a quantitative evaluation of all proposed methods, measuring precision, recall, F1-score, and false positive rate. This evaluation provides a clear benchmark for future research and tools in this domain.

2 RELATED WORK

2.1 Methodology

To identify relevant research on booter website detection, a systematic search was conducted for studies specifically addressing booters using keywords such as “booter classification” and “booter website identification” on Google Scholar and ChatGPT. This initial search identified foundational works including Santanna et al. (2017) and Zhang et al. (2019), which focus on booter detection but employ traditional machine learning approaches. Given the limited availability of recent literature dedicated exclusively to booter detection, the scope was broadened to include general website classification techniques. Keywords such as ’LLM website classification’ and ’AI website classification’ were used to identify advancements in LLM-based website classification that could be applicable to detecting booter websites.

2.2 Traditional Machine Learning-Based Detection

Santanna et al. (2017) and Zhang et al. (2019) employed traditional machine learning techniques to classify booter websites, focusing on structured feature extraction and statistical classification. Santanna et al. developed a three-step process: crawling 928 suspect URLs from sources like Google, YouTube, and hacker forums using keywords (“booter,” “stresser,” “ddoser”), scraping features such as domain age, WHOIS privacy, number of pages, and presence of terms of service, and classifying them using eight methods. The Cosine distance method, optimized with a supervised machine learning algorithm, achieved 95.5% accuracy on a dataset of 465 URLs (140 booters, 325 non-booters), with feature relevance assessed via the odds ratio metric. Zhang et al. proposed a two-component system with a crawler collecting 718 URLs (51 confirmed booters) via targeted Google searches and a classifier processing text through a 66,448-dimensional bag-of-words model. After preprocessing (stop word removal) and feature selection using global, type-based, and combined inter-category distance measures to select 800 terms, Multinomial Naive Bayes achieved 98.74% accuracy, outperforming Linear SVM. These methods excel in leveraging predefined features but may struggle with capturing complex semantic or visual patterns critical for detecting evolving booter websites.

2.3 Large Language Model-Based Text Analysis

Sava (2024) and Sasazawa and Sogawa (2025) highlight the potential of large language models (LLMs) for advanced website classification and crawling, with implications for booter website detection. Sava explored self-hosted, open-source LLMs like LLaMA, Gemma, Phi, and command-r-plus (104 billion parameters) using the Ollama framework to categorize 5,000 websites from Cloudflare Radar, achieving 75.67% accuracy with command-r-plus by processing raw HTML and accessibility tree data without labeled training data, demonstrating LLMs’ strength in extracting semantic patterns like DDoS-related keywords or sentiment, offering a scalable alternative to traditional feature engineering [10]. Conversely, Sasazawa and Sogawa proposed an LLM-based method using GPT-4o and GPT-4o-mini to classify web pages as "Index Pages" (with hyperlinks) or "Content Pages" (like articles), achieving a high F1 score of 0.894 on a dataset of 10,000 pages per news website, enhancing crawling efficiency by prioritizing index pages to discover new content [12]. While not specific to booters, their approach is relevant for law enforcement, as adapting LLM-based classification could be applied to identify booter websites, improving the discovery of suspect URLs in dynamic ecosystems like hacker forums or dark web marketplaces, complementing traditional keyword-based crawling methods.

2.4 Multimodal and Autonomous LLM-Based Detection

Lee et al. (2024) and Nakano et al. (2025) utilized advanced LLMs to integrate multiple data modalities or autonomous reasoning for malicious website detection, applicable to booter identification. Lee et al. proposed a two-phase phishing detection system using multimodal LLMs (GPT-4, Claude3, Gemini Pro-Vision) to analyze webpage screenshots, HTML content, or both. The first phase identifies brands (like WhatsApp) based on visual elements (logos, themes) and text, while the second verifies domain-brand consistency, classifying mismatches as phishing. Evaluated on 2,981 benign and 1,499 phishing webpages (October–December 2023), the system achieved F1-scores of 0.90–0.92, with Claude3 excelling on screenshots and GPT-4 benefiting from combined inputs. Nakano et al.’s ScamFerret system used GPT-4 for autonomous scam detection, analyzing URLs through multi-step reasoning and collecting external data (such as WHOIS, DNS records, social media) [11]. It achieved 0.972 accuracy for English scams and 0.993 for multilingual shopping scams without labelled datasets, though it faces challenges with cost and image-based scams. These methods, combining visual, textual, and external data or autonomous analysis, offer robust frameworks for booter detection by capturing diverse features like webpage layouts, buttons, and metadata.

Table 1. Presence/Absence of Features in Website Classification Studies

Paper/Author(s)	Structural/Metadata	Textual	Visual	LLM-Based	Booters	Open Source Models
Santanna et al. (2017)	Yes	No	No	No	Yes	Yes
Zhang et al. (2019)	No	Yes	No	No	Yes	Yes
Sava (2024)	Yes	Yes	No	Yes	No	Yes
Lee et al. (2024)	Yes	Yes	Yes	Yes	No	No
Nakano et al. (2025)	Yes	Yes	Yes	Yes	No	No
Sasazawa and Sogawa (2025)	No	Yes	No	Yes	No	No

2.5 Critical Features and Suitable AI Models

To address RQ1, which is identifying the critical features of booter websites and the AI models best suited for leveraging these features in detection tasks, the related work highlights a range of features and corresponding models. Santanna et al. (2017) emphasize structural and metadata features, such as domain age, number of pages, and presence of terms of service, using a classifier with the Cosine distance method augmented by machine learning, achieving 95.5% accuracy. Zhang et al. (2019) focus on textual content, extracting features via a bag-of-words model and employing Multinomial Naive Bayes with a global distance-based feature selection, yielding 98.74% accuracy. Sava (2024) demonstrates the potential of LLMs like command-r-plus for processing raw HTML and accessibility tree data, achieving 75.67% accuracy in website categorization, suggesting LLMs' suitability for semantic text analysis in booter detection. Lee et al. (2024) and Nakano et al. (2025) extend this by incorporating visual elements (like screenshots for logos, layouts) and external metadata (WHOIS, DNS records), using multimodal LLMs like GPT-4 and Claude3, which achieve high F1-scores (0.90–0.92) in phishing detection and 0.972–0.993 accuracy in scam detection. These studies collectively explore textual content, visual elements, and metadata as features for website classification, with traditional classifiers and LLMs applied to these tasks.

2.6 Limitations of Current Methods

While Santanna et al. (2017) and Zhang et al. (2019) provide robust frameworks for booter website detection, their reliance on traditional machine learning with structural features (domain age, page count) and text-based features (bag-of-words models) faces challenges in adapting to the evolving nature of booter websites. These methods depend on predefined feature sets that may struggle to capture complex semantic patterns, sentiment, or context-specific terminology used by modern booter sites to evade detection. Additionally, their focus on textual and structural data overlooks visual elements, such as website layouts, design motifs, or graphical components, which are often consistent across booter platforms but difficult to quantify using traditional feature engineering. Furthermore, neither study addresses the computational efficiency of their approaches, omitting details on the time required for model training or prediction, which is critical for practical deployment in dynamic, real-world settings. The rapidly changing DDoS-for-hire landscape, where many identified booter sites have been taken down and new ones have emerged with altered characteristics, underscores the need for more flexible and adaptive detection methods. Large language models (LLMs) offer a promising solution to these challenges, as they can process raw textual content, extract nuanced semantic and contextual features, and incorporate multimodal data, such as visual elements, to enhance the accuracy, adaptability, and potentially the efficiency of booter website detection.

2.7 Feature Comparison Across Studies

Table 1 summarizes the presence or absence of features used in the discussed studies for booter website detection or related website classification tasks. The table highlights the diversity of approaches and the increasing adoption of LLMs and multimodal methods in recent works.

3 METHODOLOGY

3.1 Dataset

The dataset for this study was compiled to create a robust and diverse collection of both booter and non-booter websites, enabling effective training and evaluation of AI-driven detection models. It consists of two primary categories: booter websites and benign websites. The booter website collection is further divided into two subsections (online booters and offline booters) to account for both currently operational sites and historical snapshots of sites that may no longer be active. This approach ensures the dataset captures a wide range of booter characteristics, reflecting both contemporary and past trends in DDoS-for-hire services. The dataset includes a total of 274 websites: 126 booter websites and 148 benign websites.

3.1.1 Online Booters. The online booters dataset includes active booter websites, collected through an automated web scraping process designed to find and confirm these sites. The process used a Python script with the Selenium library, employing a headless web browser to gather URLs from DuckDuckGo searches, complemented by a Bing crawler using requests and BeautifulSoup to extract additional URLs from Bing search results, including decoding Bing's redirect URLs to obtain direct website addresses. Search terms like "booter," "stresser," and "DDoS service" were used across both search engines to locate relevant websites. The scripts collected unique website addresses, using random pauses and changing identifiers

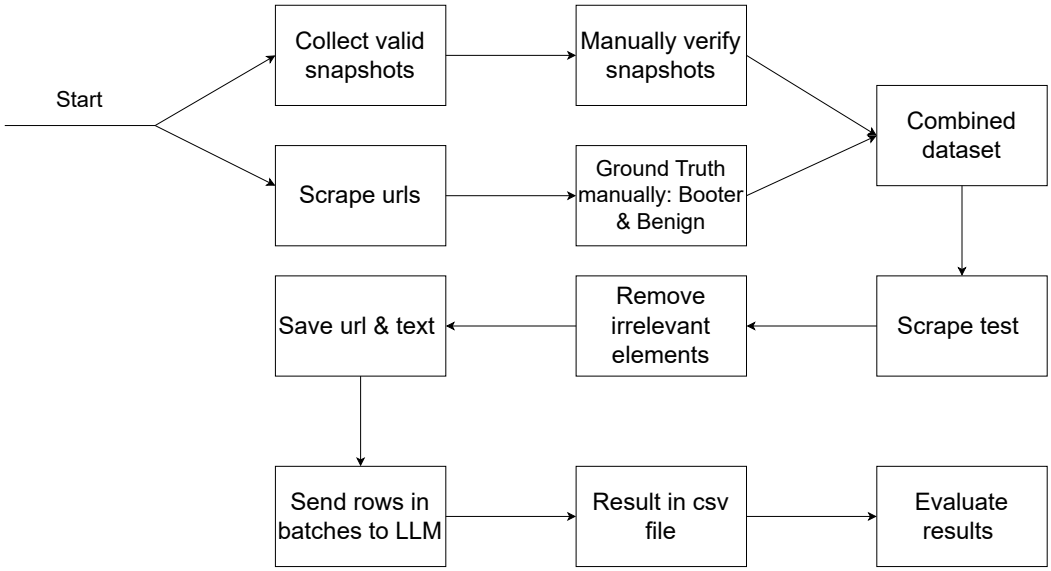


Fig. 1. Flowchart of the Booter Website Detection Process

to avoid detection, while filtering out unrelated links (like those from "duckduckgo.com" or "bing.com"). After gathering the URLs, a manual review was conducted to distinguish booter websites from non-booter ones, establishing ground truth for the dataset, with the non-booter websites included as benign URLs.

3.1.2 Offline Booters. Historical snapshots of booter websites were sourced from the Wayback Machine using a curated blacklist from Santanna et al. (2017), providing a trusted foundation for identifying known DDoS-for-hire services. An asynchronous Python script queried the Wayback Machine's CDX API to retrieve recent snapshots with valid status codes, enabling efficient collection of archived web content. The HTML content was parsed using BeautifulSoup to extract meaningful text, with non-content elements like scripts, styles, and navigation bars removed to focus on core website data. To ensure snapshot quality, various language models, including Llama and Phi, were tested for validation. However, many models flagged snapshots as invalid due to their sensitive content, potentially limiting the dataset. Including explicit mentions of booters in the validation prompts could have addressed this issue but risked influencing the types of snapshots retrieved, which could bias further research.

Consequently, the Mistral language model was selected for its balanced performance. It reliably filtered out irrelevant pages, such as error messages, seized site notices, parked domains, or pages indicating the domain was for sale. At the same time, it preserved snapshots with operational website content, such as details about DDoS services or pricing. These validated snapshots were manually verified to confirm their relevance and added to the dataset as offline booter records. This approach was valuable because it captured historical booter websites that are no longer active, enriching the dataset for analyzing trends and patterns in DDoS-for-hire services over time. Additionally, by including archived snapshots, the dataset was expanded to include a wider variety of websites for testing and analysis, offering a more comprehensive view of booter operations that real-time crawling alone might miss, while maintaining neutrality in model validation to support unbiased research.

3.2 Pre-processing

For real-world law enforcement applications and research purposes, specific pipelines were developed to collect and process data from websites, as detailed in the following subsections.

3.2.1 Real-Time Processing for Law Enforcement. For real-world law enforcement applications, a dedicated pipeline was developed to identify and classify booter websites in real time. This approach involves a Python script that searches for potential booter websites using keywords such as "booter," "stresser," and "DDoS service" on search engines like DuckDuckGo. The script collects URLs, stores them in a CSV file, and then processes each URL in real time by accessing the live website, extracting the first 1000 characters of cleaned text (after removing non-content HTML elements), and performing classification using the selected AI models. This real-time method ensures that law enforcement can detect and respond to active booter websites promptly, reflecting their current state on the internet.

3.2.2 Pre-Processed Dataset for Research. For research purposes, to ensure efficiency and consistency in evaluating the classification models, a pre-processing script was developed to collect and standardize text content from both online and offline websites in advance. The script extracts the first 1000 characters of cleaned text from each website in the dataset (274 websites: 126 booter and 148 benign) by removing non-content HTML elements. This approach was designed to save time, reduce redundant scraping during repeated classification experiments, and ensure uniform input data for the AI models. The pre-processed data, along with screenshot file paths for multimodal analysis, was stored in a CSV file. By pre-processing the data, variations due to network connection speeds or website accessibility issues are minimized, providing a stable foundation for comparative analysis of model performance.

3.3 Classification Methodology

To systematically evaluate the effectiveness of AI-driven techniques for booter website detection, I developed two distinct but complementary classification pipelines: (1) a text-based classification approach utilizing large language models (LLMs) for semantic analysis, and (2) a multimodal classification approach for visual elements through screenshot analysis. Both pipelines were designed to process the curated dataset of 274 websites (comprising 126 confirmed booter sites and 148 benign control sites) stored in standardized CSV format, containing pre-processed text content and, for the multimodal approach, file paths to corresponding webpage screenshots.

The classification systems output structured JSON results containing several key metrics:

- Binary classification (`booter: true/false`) indicating the model's determination
- Confidence level (`high/low`) reflecting the model's certainty in its classification
- Explanatory rationale (`reason`) providing human-interpretable justification for the decision
- Processing time: quantifying computational efficiency

This dual-pipeline architecture enables comprehensive evaluation of both unimodal (text-only) and multimodal (visual) approaches, allowing direct comparison of their relative strengths and limitations in booter website detection. The following subsections detail the implementation and model selection for each approach.

3.3.1 Model Selection and Architecture. For the text-based classification pipeline, I evaluated several large language models (LLMs) available through the Ollama framework, chosen for its zero-cost accessibility, privacy-preserving local execution, and support for models compatible with standard laptop hardware. This selection criteria was particularly important given the sensitive nature of law enforcement applications and the need for reproducible, cost-effective solutions. The evaluated models represent diverse architectural approaches and parameter scales:

- **Mistral (7B)**: A 7-billion parameter model known for its efficient performance in text classification tasks.
- **Llama3.1 (7B)**: Meta's open-weight model optimized for balanced performance across various NLP tasks.
- **Mistral-Nemo**: A 12B model built by Mistral in collaboration with NVIDIA.
- **Phi3:mini**: A compact yet capable model from Microsoft's Phi series, designed for resource-constrained environments
- **Qwen2.5**: Alibaba's multilingual model with strong semantic understanding capabilities.

The multimodal classification pipeline presented unique technical challenges, particularly in image processing and cross-modal understanding. Initial experiments with full-page screenshots revealed limitations in model performance, as the visual complexity often exceeded the models' ability to extract relevant features. I therefore refined the approach to focus on landing page screenshots (capturing the critical first impression and primary interface elements) paired with the first 1000 characters of textual content - a compromise that maintained information richness while respecting model input limitations. Screenshots were converted to base64 encoding for model compatibility. The evaluated vision-language models included:

- **LLaVA (7B and 13B)**: Open-source visual language models with varying capacity for image-text understanding
- **Gemma3 (4B and 12B)**: Google's lightweight multimodal models with strong visual grounding capabilities
- **Bakllava**: A specialized vision-language model optimized for web content analysis
- **Qwen2.5-vl**: Alibaba's multimodal model designed for robust image and text integration, suitable for webpage analysis.

This systematic evaluation across model architectures and modalities allows us to address RQ2 by identifying the most effective combinations of model type and input features for booter website detection. The diversity in model sizes (from 4B to 13B parameters) also provides insights into the trade-offs between computational requirements and classification accuracy - a critical consideration for real-world law enforcement applications.

3.3.2 Preliminary Evaluation of Vision-Language Models. To determine the most effective vision-language model for the multimodal classification pipeline, a preliminary evaluation was conducted using a subset of five booter website URLs from the dataset. The test focused exclusively on image-based classification, utilizing landing page screenshots to assess the models' ability to identify booter websites based on visual elements, such as text embedded in images, which text-based LLMs might miss. Four multimodal models were evaluated: LLaVA (7B), Bakllava, Gemma3 (4B), and Qwen2.5VL. Screenshots were converted to base64 encoding to ensure compatibility with the models, and each model classified the websites as either booter or benign.

The evaluation revealed significant performance differences. Qwen2.5VL achieved perfect accuracy, correctly classifying all five websites. In

contrast, LLaVA (7B), Bakllava, and Gemma3 (4B) correctly classified only one, zero, and zero websites, respectively. Analysis showed that LLaVA, Bakllava, and Gemma3 struggled with optical character recognition (OCR), frequently failing to extract text from screenshots or misinterpreting key terms like “DDoS” or “stresser.” These models often hallucinated letters or phrases, resulting in incorrect classifications. For example, the term “stresser” was sometimes misread as unrelated words, which compromised their ability to detect booter-specific visual cues. Qwen2.5VL, however, demonstrated robust OCR and contextual understanding, accurately extracting and interpreting text within images, such as promotional content or attack service labels, making it highly effective at identifying booter websites based on visual features.

Despite Qwen2.5VL’s superior performance, its processing time was prohibitively long, requiring approximately 45,000 seconds to process the dataset. This excessive computational cost made it impractical for large-scale applications, particularly for law enforcement scenarios requiring rapid detection. Consequently, to prioritize efficiency and meet time constraints, subsequent multimodal experiments were not pursued with Qwen2.5-VL or other vision-language models. Instead, the study focused on text-based classification pipelines, which offered a better balance of accuracy and computational efficiency for the full dataset of 274 websites.

3.3.3 Prompt Engineering Strategy. Recognizing that Large Language Model performance is highly sensitive to prompt design, I conducted systematic experiments with three distinct prompt architectures to evaluate their impact on classification accuracy and robustness. This prompt variation study addresses a critical aspect of AI-driven detection systems - how instruction framing affects model performance in security applications. The prompt engineering strategy was designed to balance specificity with flexibility, enabling comparative analysis across different model architectures and sizes.

Descriptive Prompt (Structured Definition). The baseline prompt employs a comprehensive, definition-based approach that explicitly enumerates booter website characteristics and classification criteria. This verbose prompt serves multiple functions:

- Provides detailed feature descriptors (e.g., "Layer 4/7 attacks", "crypto-only payments")
- Establishes clear decision boundaries between booter and benign websites
- Specifies exact JSON output formatting requirements
- Includes confidence-level qualification criteria

This structured approach mirrors law enforcement operational guidelines, ensuring classifications meet evidentiary standards for potential investigative follow-up.

Concise Prompt (Minimalist Instruction). The condensed prompt variant tests model performance under constrained instructional conditions, simulating scenarios where:

- Computational overhead must be minimized for high-volume processing
- Rapid deployment is prioritized over optimal accuracy
- Model inference speed is critical

The concise prompt reduces the complexity and length of instructions compared to the descriptive prompt, while maintaining the essential logic for effective classification.

Few-Shot Prompt. The demonstration-based prompt incorporates a few carefully curated examples representing:

(1) Clear booter case (explicit attack service offering)
(2) Definitively benign case

This approach tests models’ ability to generalize from limited examples, which is particularly valuable given the rapidly evolving tactics of booter sites that may not match predefined feature lists. The few-shot design also helps mitigate "hallucination" tendencies in smaller models by grounding responses in concrete examples.

4 RESULTS

This section presents the evaluation results of the text-based classification pipeline for booter website detection, using a dataset of 274 websites (126 booter and 148 benign). Six large language models: Mistral (7B), Phi3:mini, Llama3.1 (7B), Mistral-nemo, Qwen2.5 and Mistral-nemo were tested with three prompt architectures: Descriptive, Concise and Few-shot. Performance metrics include false negatives (FN), false positives (FP), precision, recall, F1-score, and total processing time (in seconds). The results are summarized in three tables, one for each prompt type, to facilitate comparison across models.

4.1 Descriptive Prompt Results

The Descriptive prompt provided detailed feature descriptors and classification criteria to ensure robust and interpretable classifications. Table 2 summarizes the performance of the three LLMs.

Table 2. Performance Metrics for Descriptive Prompt (274 Websites)

Model	FN	FP	Precision	Recall	F1-Score	Accuracy
Mistral (7B)	1	2	0.984	0.992	0.988	0.989
Phi3:mini	3	2	0.984	0.976	0.980	0.982
Llama3.1 (7B)	1	0	1.000	0.992	0.996	0.996
Qwen2.5	9	0	1.000	0.929	0.963	0.967
Mistral-nemo	0	0	1.000	1.000	1.000	1.000

As seen in table 2, Mistral-nemo achieved the highest performance with a perfect F1-score (1.000), precision (1.000), and recall (1.000), indicating no false positives or false negatives. Llama3.1 followed closely with an F1-score of 0.996, perfect precision (1.000), and high recall (0.992), making it highly reliable for law enforcement applications.

4.2 Concise Prompt Results

The Concise prompt minimized token consumption to prioritize processing speed, suitable for high-volume scenarios. As seen in table 3, Llama3.1 achieved the highest F1-score (0.992), with balanced precision (0.992) and recall (0.992), demonstrating robustness to prompt brevity. Mistral (7B) followed closely with an F1-score of 0.988 and high recall (0.992). Phi3:mini had the lowest F1-score (0.929), with 15 false positives and lower recall (0.881), indicating reduced sensitivity to concise prompts.

Table 3. Performance Metrics for Concise Prompt (274 Websites)

Model	FN	FP	Precision	Recall	F1-Score	Accuracy
Mistral (7B)	1	2	0.984	0.992	0.988	0.989
Phi3:mini	15	2	0.982	0.881	0.929	0.938
Llama3.1 (7B)	1	1	0.992	0.992	0.992	0.993
qwen2.5	13	1	0.991	0.897	0.941	0.952
Mistral-nemo	5	1	0.991	0.960	0.976	0.978

4.3 Few-shot Prompt Results

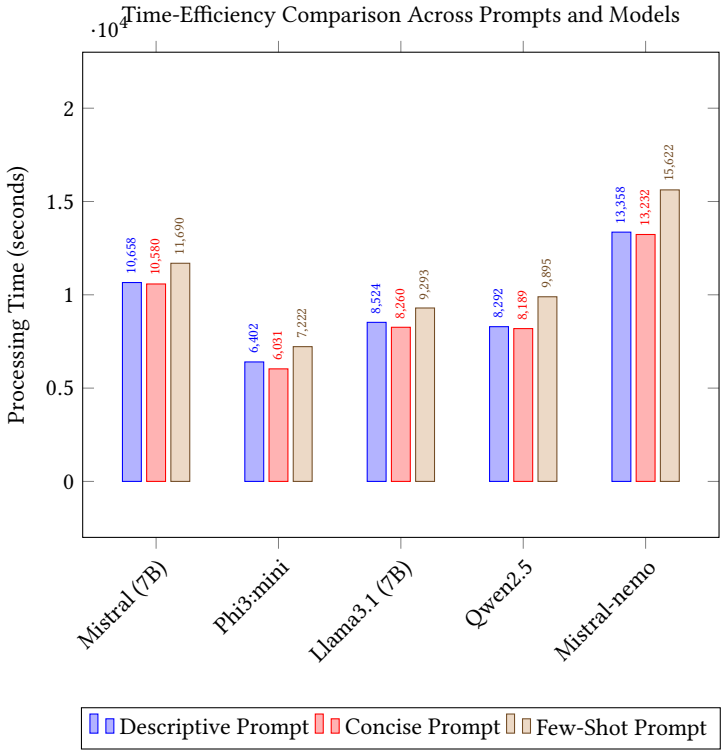
In Table 4, mistral-nemo led with the highest F1-score (0.992) and perfect recall (1.000), alongside high precision (0.984), indicating robust detection with minimal errors. Llama3.1 (7B) and Qwen2.5 both achieved strong F1-scores (0.988), with Llama3.1 showing slightly higher precision (0.992) and Qwen2.5 matching its recall (0.992). Mistral (7B) also performed well with an F1-score of 0.980. Phi3:mini had the lowest F1-score (0.893) and recall (0.833) due to 21 false negatives, indicating limitations in detecting booter websites.

Table 4. Performance Metrics for Concise Prompt (274 Websites)

Model	FN	FP	Precision	Recall	F1-Score	Accuracy
Mistral (7B)	1	4	0.969	0.992	0.980	0.982
Phi3:mini	21	4	0.963	0.833	0.893	0.909
Llama3.1 (7B)	2	1	0.992	0.984	0.988	0.989
Qwen2.5	1	2	0.984	0.992	0.988	0.989
Mistral-nemo	0	2	0.984	1.000	0.992	0.993

4.4 Comparing time-efficiency of each combination

As observed in the bar graph, Mistral-nemo takes the greatest amount of time while phi3:mini takes the least amount of time. Additionally, the few-shot prompt seems to be the most computationally intensive prompt while the concise prompt as expected being the least.



4.5 Answering RQ2

Addressing RQ2 (How can various types of LLMs be designed and compared for booter classification?), my evaluation shows that text-based LLMs (Mistral-nemo, Llama3.1) with Descriptive prompts outperform other models. The high F1-scores (1.000 for Mistral-nemo) and perfect precision (1.000 for Llama3.1) demonstrate their suitability for law enforcement applications, while the multimodal model Qwen2.5-VL offers limited advantages due to their extensive computational costs despite its ability to parse screenshots more effectively than other multimodal.

4.6 Answering RQ3

For RQ3 (How accurate are the proposed AI techniques in classifying booter websites compared to existing methods?), the proposed LLM-based methods significantly outperform traditional approaches. Santanna et al. (2017) achieved 95.5% accuracy using structural and metadata features with a Cosine distance classifier, while Zhang et al. (2019) reported 98.74% accuracy with a bag-of-words model and Multinomial Naive Bayes. In contrast, the text-based LLMs, particularly Mistral-nemo with the Descriptive prompt, achieved a perfect F1-score (1.000) and accuracy (1.000), with Llama3.1 close behind at an F1-score of 0.996 and accuracy of 0.996. These improvements stem from LLMs' ability to capture complex semantic patterns in raw text, reducing reliance on predefined feature sets that may miss evolving booter characteristics. The low false positive

rates (0 for Mistral-nemo and Llama3.1) are particularly valuable for law enforcement, ensuring minimal misclassification of legitimate websites. While traditional methods are effective, their static feature engineering limits adaptability compared to the dynamic, context-aware analysis of LLMs, making my approach more robust and accurate for real-world deployment.

5 CONCLUSION

This study evaluated the effectiveness of large language models (LLMs), in detecting booter websites for law enforcement applications. By comparing text-based and multimodal classification approaches across multiple models and prompt architectures, the research provides critical insights into optimizing accuracy, efficiency, and practicality in real-world scenarios. To enhance dataset robustness, the study also developed an LLM-based pipeline to retrieve and validate archived snapshots of defunct booter websites from Archive.org, ensuring comprehensive coverage for analysis.

5.1 Key Findings

This study evaluated AI-driven methods for detecting booter websites, focusing on text-based and multimodal large language models (LLMs). Text-based LLMs, particularly Mistral-nemo with a Descriptive prompt, achieved a perfect F1-score (1.000) and accuracy (1.000), surpassing traditional methods like Santanna et al. (95.5% accuracy) and Zhang et al. (98.74% accuracy). Llama3.1 also performed strongly (F1: 0.996), with perfect precision, minimizing false positives critical for law enforcement. Concise prompts reduced processing time but slightly lowered accuracy (like Mistral's F1: 0.988), while Few-shot prompts improved generalization for some models. Multimodal models like Qwen2.5-VL showed promise in screenshot analysis but were impractical due to high computational costs (approximately 45000 seconds for 274 websites). For law enforcement, text-based LLMs with Descriptive prompts are recommended for high accuracy, with Phi3:mini offering faster processing for high-volume screening. Consequently, the combination of mistral-nemo with the descriptive prompt was chosen to be the best combination to classify booter websites despite it being slightly more computationally intensive than the other models. The comparison with pre-existing methods can be seen in Table 5. Finally, the paper also presents a valid way to retrieve valid snapshots from archive.org using validation by mistral llm.

The findings answered the following research questions:

RQ1: Critical features of booter websites used for classification include text from webpage and whois data. Cosine distance method and Bag-of-words model were used to classify booters. However for other studies llms were used to analyze both textual and visual content in websites for classification.

RQ2: Text-based LLMs, particularly Mistral-nemo with Descriptive prompts, achieved optimal performance (F1: 1.000) by capturing semantic patterns, outperforming multimodal models due to lower computational costs.

RQ3: The proposed LLM-based methods achieved up to 100% accuracy, surpassing traditional methods (95.5%–98.74%), due to their ability to adaptively analyze complex semantic and contextual features.

Table 5. Accuracy Comparison with Previous Works

Study / Model	**Accuracy**
Santanna et al. (2017)	95.5%
Zhang et al. (2019)	98.74%
Mistral-nemo (Descriptive Prompt, This Study)	**100%**

5.2 Implications on Law Enforcement

This study underscores the practical applicability of lightweight large language models (LLMs) in cybersecurity contexts, particularly for law enforcement agencies combating the proliferation of booter services. The findings demonstrate that models such as Mistral-nemo and Llama3.1, when used with well-crafted descriptive prompts, achieve exceptionally high accuracy and reliability. These results suggest that LLM-driven classification systems can be effectively integrated into real-time monitoring and enforcement pipelines, offering scalable and cost-efficient solutions for detecting and responding to evolving cyber threats.

5.3 Ethical Concerns

While the proposed methodology offers significant potential, it also raises important ethical concerns. Misclassification of legitimate websites as booter platforms, particularly in automated law enforcement workflows, could result in unwarranted investigations, reputational damage, or legal consequences for innocent parties. Therefore, it is essential to incorporate robust human verification, transparency in classification criteria, and clear avenues for redress in any operational deployment. Ensuring fairness, accountability, and proportionality in the use of AI-driven detection systems must remain a guiding principle in their adoption.

5.4 Limitations

This study faced two main limitations. First, dataset collection was limited to DuckDuckGo and Bing with a narrow set of keywords, potentially overlooking booter sites indexed by other platforms such as Tor. Second, hardware constraints (31GB RAM) restricted model evaluation to lightweight LLMs (Mistral 7B, Phi3:mini), preventing testing of more advanced models (GPT-4, Claude3) that could enhance detection, especially in multimodal tasks. All experiments were run on a personal laptop, where background processes may have affected the accuracy of performance timing and resource measurements.

REFERENCES

[1] Central District of California, "Federal Authorities Seize 13 Internet Domains Associated with 'Booter' Websites that Offered DDoS Computer Attack Services," United States Department of Justice, 2023. Available: https://www.justice.gov/usao-cdca/pr/federal-authorities-seize-13-internet-domains-associated-booter-websites-offered-ddos.

[2] Cloudflare."What is an IP stresser? | DDoS booters," Cloudflare. Available: https://www.cloudflare.com/en-ca/learning/ddos/ddos-attack-tools/ddos-booter-ip-stresser/.

[3] Ben Collier, Anh V. Vu, Daniel R. Thomas. 2025. Assessing the Aftermath: the Effects of a Global Takedown against DDoS-for-hire Services. arXiv:2502.04753v1. Available: https://arxiv.org/html/2502.04753v1.

[4] José Jair Santanna, Ricardo O. De Schmidt, Daphne Tuncer, Joey de Vries, Lisandro Z. Granville, and Aiko Pras, 2017. Booter list generation: The basis for investigating DDoS-for-hire website. Available: https://doi.org/10.1002/nem.2008.

[5] Wand Zhang, Xu Bai, Chanjuan Chen and Zaolin Chen. 2019. Booter Blacklist Generation Based on Content Characteristics. pp. 529–542. Available: https://link.springer.com/chapter/10.1007/978-3-030-12981-1_37.

[6] Anna Lieb, Maneesh Arora and Eni Mustafaraj. 2025. Creating Targeted, Interpretable Topic Models with LLM-Generated Text Augmentation. arXiv:2504.17445. Retrieved from: https://arxiv.org/abs/2504.17445.

[7] Jehyun Lee, Peiyuan Lin, Bryan Hool and Dinil Mon Dikavakaran. 2024. Multimodal Large Language Models for Phishing Webpage Detection and Identification. arXiv:2408.05941. Retrievd from: https://arxiv.org/abs/2408.05941.

[8] Brian Krebs. 2020. Owners of DDoS-for-Hire Service vDOS Get 6 Months Community Service. Krebs on Security. Retrieved June 12 2025 from: https://krebsonsecurity.com/2020/06/owners-of-ddos-for-hire-service-vdos-get-6-months-community-service/.

[9] Sergiu Gatlan. 2023. FBI seizes 13 more domains linked to DDoS-for-hire services. BleepingComputer, Retrieved June 5 2025 from: https://www.bleepingcomputer.com/news/security/fbi-seizes-13-more-domains-linked-to-ddos-for-hire-services/.

[10] Dragos-Mihail Sava. 2024. *Text-based classification of websites using self-hosted Large Language Models: An accuracy and efficiency analysis*. Bachelor's thesis, University of Twente. Enschede, The Netherlands. Available at: https://essay.utwente.nl/101155/1/Sava_BA_EEMCS.pdf

[11] Hiroki Nakano, Takashi Koide and Daiki Chiba. 2025. *ScamFerret: Detecting Scam Websites Autonomously with Large Language Models*. arXiv:2502.10110. Retrieved from: https://arxiv.org/abs/2502.10110

[12] Yuichi Sasazawa, Yasuhiro Sogawa. 2025. *Web Page Classification using LLMs for Crawling Support*. arXiv:2505.06972. Retrieved from: https://doi.org/10.48550/arXiv.2505.06972

A APPENDIX

A.1 AI Statement

During the preparation of this work the author used ChatGPT in order to rephrase and enhancing some of the text in the paper and used it to generate and debug some of the code. After using this tool/service, the author reviewed and edited the content as needed and take(s) full responsibility for the content of the work.

A.2 Prompts

A.2.1 Descriptive Prompt. You are an AI tasked with classifying webpages as booter or non-booter based on their content.

Booter Traits (classify as booter): - Offers "Booter", "Stresser", or "DDoS" services, e.g., "IP Stresser", "Launch Stress Test". - Mentions Layer 4/7 attacks, e.g., "UDP Flood", "TCP SYN", "HTTP/2 SLOWLORIS". - Lists pricing/subscriptions, e.g., "$5/month", "VIP plan", "crypto payments". - Highlights power, e.g., "50-180 Gbps", "20K rps", or anonymity, e.g., "spoofed attacks", "no logs". - Promotes bypassing security, e.g., "bypass Cloudflare", "private scripts". - Includes login/attack hubs, e.g., "Send Attack", "Free Trial".

Non-Booter Traits (classify as non-booter): - Educational or informational content explaining DDoS/stressers without offering services.

Output JSON: "booter": true | false, "confidence": "high" | "low", "reason": "Explain citing specific phrases."

Task: Classify the webpage text as booter or non-booter using the traits above.

A.2.2 Concise Prompt. Determine if the webpage is a booter website, meaning it promotes or sells services to disrupt networks. Look for intent to offer attacks, such as mentions of attack power, payment options, or anonymity. If the text only informs about DDoS (e.g., academic content), it's non-booter. Output JSON: "booter": true|false, "confidence": "high"|"low", "reason": "why you made this choice"

A.2.3 Few-shot Prompt. You are an AI tasked with classifying webpages as booter or non-booter based on their content. Use the following examples to guide your classification:

Example 1 (Booter): Text: "NetworkStress.XYZ Welcome to NetworkStress.XYZ Login Register! WE PROVIDE A WIDE VARIETY OF SERVER STRESS TESTING METHODS SO YOU CAN PREPARE YOURSELF AGAINST HACKERS AND CYBER CRIMINALS. WE ALSO HELP IN FIXING VULNERABILITY IN YOUR SERVERS AND HARDEN YOUR SECURITY AND FIREWALL! Previous Next We provide the best Stress Testing experience. Information on our many features Guaranteed Power We guarantee a 10-80Gbps power per boot using our LDAP/NTP method. Advanced source Our high end dedicated servers can satisfy even the most power hungry customers.. Powerful boots We're using recursive DNS servers, LDAP and NTP amplifiers, for method testing 24/7 Customer Care Our support team is online 24/7, feel free to ask us anything that on your mind by submitting a ticket.. Bitcoin accepted here We believe in the huge potential of this new digital currency.. Paypal accepted here You may now use your credit or debit card to make membership payments through PayPal. We offer one of the best Stress Te" Classification: "booter": true, "confidence": "high", "reason": "Mentions 'Stress Testing', '10-80Gbps power', 'LDAP/NTP method', and payment options like Bitcoin and PayPal, indicating booter services."

Example 2 (Non-Booter): Text: "Scam Detector: Website Reviews and Online Scam Protection Skip to content Scam Detector Search Sub Domain Searching: Domain age, Alexa rank, HTTPS valid, Blacklisting, SSL certificates, Source code, Location, IP address, WOT Trustworthiness, Spam reports, Advanced technology, Privacy Policy, Terms of Use, Contact options Scam Detector Main Menu Home WEBSITE VALIDATOR Are you just about to make a purchase online? See below if the website you want to use is legit SEARCH Save this page on your device home screen to have it handy next time you buy stuff online Sub Domain Searching: Domain age, Alexa rank, HTTPS valid, Blacklisting, SSL certificates, Source code, Location, IP address, WOT Trustworthiness, Spam reports, Advanced technology, Privacy Policy, Terms of Use, Contact options Scam Detector, Website Reviews and Online Scam Protection Scam Detector provides original information, reporting, research, reviews, and analysis on websites, domain

names, and e-commerce platforms, advising r" Classification: "booter": false, "confidence": "high", "reason": "Content focuses on scam detection and website validation, with no mention of stress testing or DDoS services."

Task: Classify the provided webpage text as booter or non-booter based on the examples above.

Output JSON: "booter": true | false, "confidence": "high" | "low", "reason": "Explain citing specific phrases."