2025 BERLIN

24 - 26 September, 2025 / Berlin, Germany

BOOSTING URL DETECTION WITH SYNTACTIC FEATURES IN SPAM EMAILS

Antonia Scherz

Net at Work, Germany

antonia.scherz@netatwork.de

www.virusbulletin.com

ABSTRACT

Traditional URL blocklisting methods often overlook recurring patterns from automated URL generation algorithms that are prevalent in high-volume spam campaigns. Attackers register new domains highly frequently, reusing shared path structures or query parameters that vary only slightly, for example by domain name or embedded random strings. Superficial variations and fast domain switching evades detection fairly often when systems rely solely on domain-level or character-based features. In this paper we demonstrate the potential of recurring syntactic patterns and derived features that capture URL similarities across spam campaigns. The approach offers a more robust detection strategy and classifies related malicious domains immediately for URLs with sufficient string lengths. If applicable, it increases detection coverage and improves resilience against obfuscation techniques commonly used in large-scale spam and phishing operations. Our syntax-based detection complements existing detection methods that focus on character-level (e.g. frequencies of special characters) or content-based features (e.g. suspicious domain names, email body text).

We propose a four-step procedure to include syntactic pattern recognition:

1. Extract syntactic patterns from the URLs using regex representation.
2. Identify similarity in n-gram usage within pre-filter syntactic patterns.
3. Refine classification labels for URLs and domains based on n-grams from detected domains.
4. Classify URLs in each cluster as malicious or benign based on cluster information and additional (traditional) email metadata.

This multi-step approach links seemingly dissimilar URLs to the same spam campaign origin, even when traditional detection techniques fail.

We apply this technique to a large corpus of URLs extracted from email bodies, demonstrating where syntactic features excel and where they fail. Our results show that combining URL structural analysis with contextual email metadata – such as sender reputation and email headers – improves the robustness of our current spam email detection systems, as validated on a dataset of over one million emails. This approach also provides additional signals to identify suspicious campaigns that evade traditional methods.

INTRODUCTION

Spam and phishing campaigns continue to be highly effective, generating substantial financial gains for malicious actors. This profitability incentivizes the use of automation to scale such operations. A common strategy involves domain generation algorithms (DGAs), which produce short-lived domains, directing users to harmful websites. The brief lifespan of these domains reduces the likelihood of them being blacklisted before they can cause harm. Another prevalent technique – central to this study – involves generating randomized paths and parameters within URLs, resulting in a continuous stream of unique indicators of compromise (IOCs). These unique URLs hinder detection and render conventional blacklist-based filtering less effective.

Despite the high variability of algorithmically generated URL strings, automation often leaves behind structural signatures. Identification and standardized representation of such signatures facilitates spam domain detection and allows relations to be drawn between domains describing spam campaigns. We explore the stand-alone value of aggregated structural pattern representation for URLs and signature detection for spam campaigns via n-gram matching.

To detect and block malicious URLs in real time, both URL and domain-based detection mechanisms have been well explored. Domain-based detection is computationally efficient, getting most of the work done analysing only domain-level attributes. URL-level detection achieves higher performance with more information based on the entire URL structure; however, at the cost of explainability of errors and model complexity. As explainability is key when blocking emails on behalf of customers, we explore the potential of extracted domain-level similarities of URL structures to improve our domain-based detection approach.

LITERATURE REVIEW

Our feature extraction approach falls broadly within three fields: URL level and domain-level detection algorithms, lexical analysis of URLs, and logistic regression-based spam detection. Performant token-based URL detection proves that certain tokens frequently recur in both benign and malicious URLs. As shown by [1], URL decomposition into tokens with assigned phishing likelihood score allows for accurate classification of URLs. In contrast, our method avoids detailed token-level analysis in the initial stage. Instead, we adopt a regular expression-based pre-filtering approach that first groups related URLs, followed by a more directed and efficient extraction of shared character strings. Our approach accounts for the dynamic characteristics of modern URL generation.

An extensive body of literature analyses feature extraction from URLs and proves their fit for URL detection in various different settings and model compositions. [2] demonstrate the predictive value of lexical features such as path segments, arguments and filenames. Alongside the contributions of [3], [4] and [5], these works provide a valuable foundation for designing features based on character distribution and entropy in URLs. Future work may integrate these insights to further enhance detection accuracy. Aggregated URL data at domain level is less well researched, especially in combination with email metadata, as demonstrated subsequently.

N-gram frequencies and similarity analysis count as a widespread string comparison method. [6] introduced the use of n-gram models to capture linguistic patterns in URLs for phishing detection. We adopt a similar approach by leveraging n-gram features to identify distinctive character sequences, but not necessarily meaningful words.

Recent work by [4] highlights the effectiveness of logistic regression models in spam detection. Given the importance of explainability in email filtering systems, especially for justifying blocked messages, we evaluate our approach using a logistic regression framework to better understand how individual features influence detection accuracy.

METHODOLOGY

First we introduce our main methodology for detecting phishing attempts based on syntactic patterns. Then we describe how to integrate this approach into an end-to-end pipeline for phishing detection.

Syntactic pattern extraction

Our approach explores a regular expression (regex) formulation of path, query and fragment parts of URLs for a systematic syntactic representation. To generate a high-level description of the lateral URL parts we divide each URL by domain, top-level domain, path, query and fragment parts. The domain part contains all domains and subdomains preceding the top-level domain. Together with the top-level domain (further referred to as DomainTld), this part will not be included in the syntactic representation using regex.

To build an automated signature extraction process for the remaining path, query and fragment parts, we leverage regex syntax over other common signature identification techniques to describe and extract patterns. This technique is highly suitable because it can handle varying degrees of randomness in different parts of a URL. In comparison, similarity in n-gram usage is computationally expensive for a large body of strings, and string-based similarity measures face challenges when length and character usage in string sequences differ. N-gram extraction will be applied later to identify signature strings indicative of spam clusters.

We further process the different URL components as follows:

1. Path:
    a. Extract common file endings.
    b. Divide the string by ‘/’ and extract the string length of each part.
    c. For all character strings, check for digits, special characters, upper- and lower-case letters, and replace the string with the corresponding regex, appending the string length.
    d. Concatenate all parts.

Note: Special characters are all represented by a dash (‘–’) at the end of the regex block. Further experiments might explore the potential for a more detailed representation.

2. Query:
    a. Divide the string by ‘&’ and divide the resulting parameter value pairs by ‘=’.
    b. Extract the regex representation describing the length of the string and use of digits, special characters, upper- and lower-case letters.
    c. Concatenate all parameters and values to pairs, and all pairs together for the full query regex.
3. Fragment:
    a. Extract emails and replace them with a structural regex representation of an email.
    b. Replace the rest of the string with the regex depending on the length and use of digits, special characters, upper- and lower-case letters, and concatenate the resulting regex blocks.

All parts, path, query and fragment blocks combined form the complete syntactic representation of each URL, as illustrated in Figures 1 and 2.

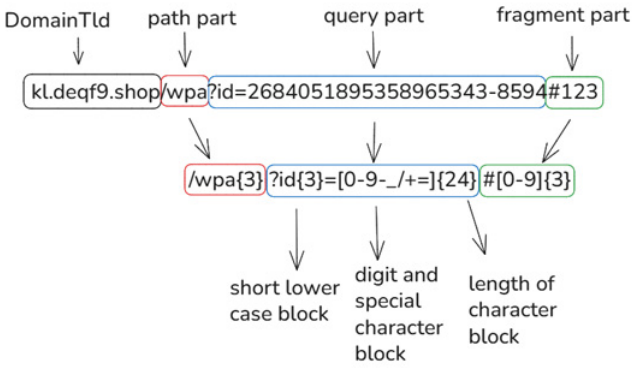

Figure 1: Mapping of URL to its syntactic patterns.

domainstld --- str	url --- str	full_regex_url --- str
netzkompass.eu	netzkompass.eu/ga/open/2-29560287-17-11484-20327-9…	/[a-z]{2}/[a-z]{4}/[a-z0-9-]{36}/
netzkompass.eu	netzkompass.eu/ga/open/2-23686206-17-11457-20322-0…	/[a-z]{2}/[a-z]{4}/[a-z0-9-]{36}/
netzkompass.eu	netzkompass.eu/ga/open/2-23686206-17-11449-20288-0…	/[a-z]{2}/[a-z]{4}/[a-z0-9-]{36}/

Figure 2: Example of URLs mapping to the same regex pattern.

All URLs in Figure 2 map to the following syntactic pattern, where each regex block (in red) describes the character types used, and each length (in blue) marks the length of the represented block:

```
/[a-z]{2}/[a-z]{4}/[a-z0-9-]{36}/
```

With this syntax extraction it can be difficult to represent all URLs of a spam cluster with one unique pattern. Random strings might include or exclude character groups by chance and differ in their string length. This makes the resulting pattern description differ in either regex blocks or length indicators (described as the red and blue parts of a regex pattern above). In our empirical analysis, we found that most variation within related groups of syntactic patterns lies in the length of regex blocks. As an alternative approach, we therefore find alternative regex pattern groups based on groups of regex representations. This eliminates length descriptions to test whether a more general description is a better input for the prediction model. An example is shown in Figure 3.

url --- str	domainstld --- str	full_regex_url --- str	full_regex_url_nodig… --- str
guettlerbau-gmbh.de/…	guettlerbau-gmbh.de	/[a-z]{10}/[a-z]{17}…	/[a-z]{x}/[a-z]{x}/
doq.org/aerzteschaft…	doq.org	/[a-z]{12}/[a-z]{14}…	/[a-z]{x}/[a-z]{x}/
doyuk.de/bekleidung/…	doyuk.de	/[a-z]{10}/[a-z]{12}…	/[a-z]{x}/[a-z]{x}/

Figure 3: Different length descriptions in regex patterns of the same domains.

N-gram extraction from pre-filtered syntactic patterns

Character-level n-grams from the lateral path components of URLs identify related URLs within groups. For each pattern group, we iterate through a decreasing sequence of n-gram sizes (starting from 20, 15, then 10, and finally 7) to identify substrings that consistently occur across the majority of URLs. We compute n-gram frequencies for each input group and select n-grams passing a significant threshold based on the fraction of total URL numbers identical to the pattern detection score (capped at 0.95). We retain only those n-grams whose frequency exceeds this threshold as likely campaign-specific signatures.

Integration of syntactic pattern information into a spam detection system

In our spam detection system, we integrate syntactic patterns into two detection subprocesses. First, we store syntactic patterns identified as malicious and include this information in our lookup process to quickly grade URLs found in email bodies. We refer to this as syntactic pattern matching. Second, we build syntactic features to include in our domain classification model, referred to as syntactic feature generation.

Syntactic pattern matching approach

Our syntactic patterns database stores patterns along with a detection score that indicates how many URLs in this pattern are malicious. This allows us to classify new URLs in ongoing spam campaigns immediately, even if a new domain is used. We also store identifying n-grams for lookup, that support immediate detection of newly generated domains. We illustrate the end-to-end evaluation process of email metadata, from arrival to classification, in Figure 4.

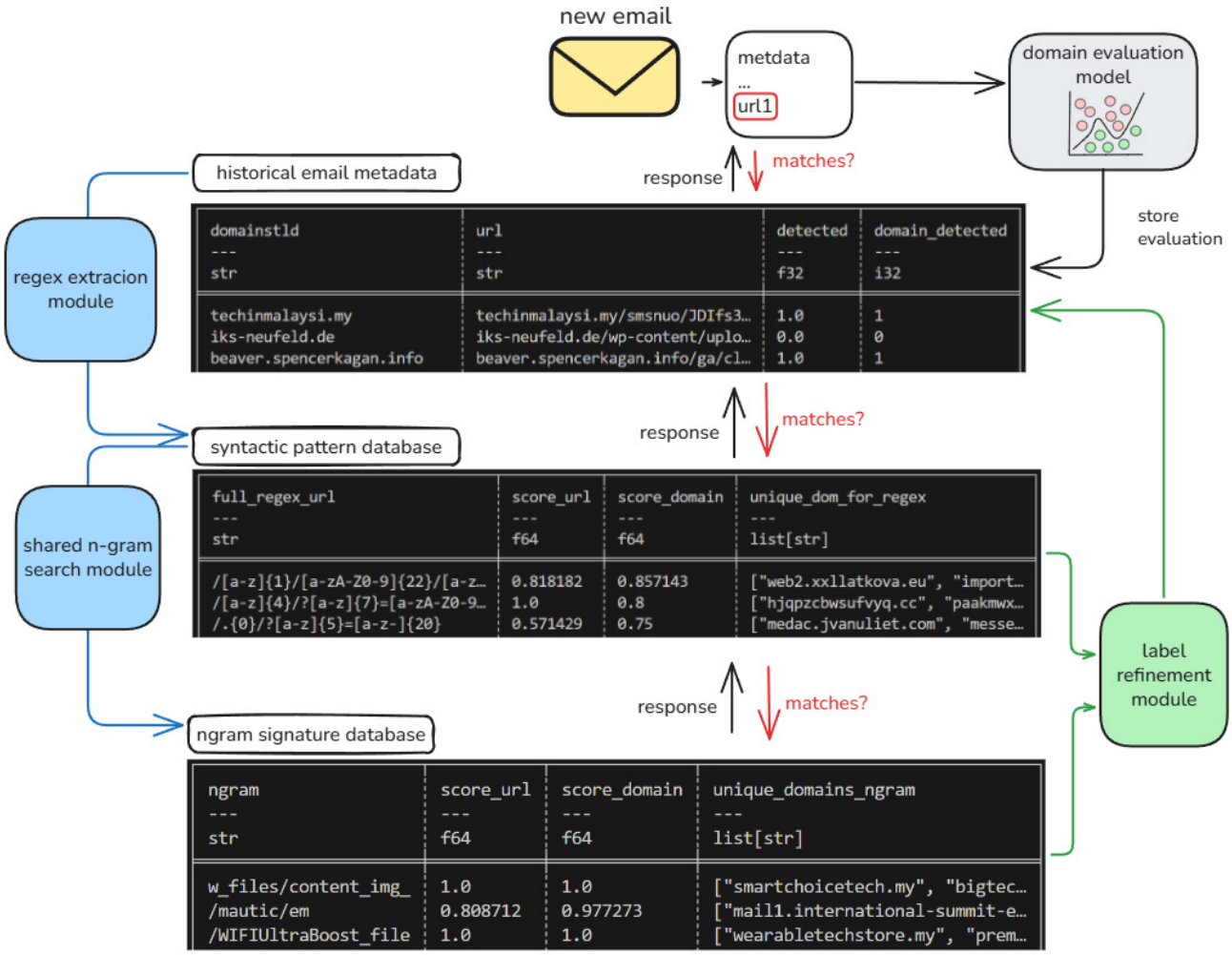

Figure 4: End-to-end evaluation of email metadata.

The historical email metadata database holds domain-level information on previously detected and not detected domains. The syntactic pattern and n-gram databases hold patterns alongside a score, on domain and URL level, of how many domains and URLs were previously associated with this pattern. We match domains, extracted syntactic patterns, and URL strings extracted from new URLs seen in email bodies, and define a URL IOC as detected if we find a match, with one exception: if a match in syntactic patterns does not contain any n-gram associated with the respective syntactic pattern, we do not detect the URL.

Syntactic feature generation and domain classification model

We extract features from regex pattern characteristics – such as the frequency of email-related expressions and the length of regex segments – and integrate them into our domain-based detection model to improve classification of newly observed domains. Our model uses aggregate email metadata for each domain, with features like the number of emails containing URLs with that domain and the time elapsed since first observation. Classification of new domains runs at fixed time intervals. Instead of quarantining emails, we temporarily reject those containing novel domains. Mail servers automatically retry delivery, typically within 15 minutes. During this interval, we collect additional data and evaluate the domains. An email including a URL with a domain not previously observed will be added to a batch for evaluation and assessed by our system within the 15-minute window, enabling classification before the second delivery attempt. New information added to our detection system must possess domain-level granularity, therefore syntactic information extracted at URL level requires aggregation to describe domain-level patterns.

EXPERIMENTS

Data

To evaluate the effectiveness of our approach, we use a dataset that represents a part of the daily stream of emails from the DACH (Germany, Austria and Switzerland) region we analyse regularly. We use email metadata for the months of April and May 2025, as shown in Table 1.

Dataset time span	Number of data points extracted	Purpose of data
Data on domain appearance for one spam campaign in April and May 2025	1.8 million URLs / 286 unique domains	Descriptive illustration
12.05 - 18.05.2025	4.8 million emails / 67k unique domains	Initial lookup of syntactic patterns with detection scores
19.05 - 25.05.2025	5 million emails / 71k unique domains	Evaluating simple syntactic pattern matching / Train and evaluate domain classification model
26.05 - 01.06.2025	4 million emails / 51k unique domains	Testing trained model

Table 1: Dataset time span, data points and analysis purposes.

Our data stems from all URLs found in a subset of emails spanning three weeks of inbound and detected traffic. To illustrate spamming behaviour interesting to our extraction approach, we use data on a spam cluster active in April and May 2025. For demonstrating the impact of syntactic feature extraction and matching, we use data for the week of 19 to 25 May. To evaluate the potential of integrating syntactic features into domain classification modelling we use one week of data to establish initial syntactic features (12 to 18 May), one week of data to train and develop our model (19 to 25 May), and one week of data only for testing performance (26 May to 1 June).

To improve our domain detection model, we include additional extracted features from syntactic patterns. These summarize information at URL level with length of syntactic patterns, number of domains related to the same pattern, and the mean number of detections for a group. We perform a second experiment in which we group these new features by patterns that become identical when ignoring the length values within patterns as described in the Methodology section. In this feature set, we take the mean of the previous features and include in addition the number of regex patterns present in the group. Other features were tested but excluded due to low predictive power. These additional features are included on top of the domain-level features described previously.

Descriptive statistics and prevalent sending patterns

We compute detection scores at both the URL and domain levels. Figures 5 and 6 present these scores as the distribution of the number of unique domains associated with each syntactic pattern. Specifically, Figure 5 shows the detection status of each URL at the time the corresponding email was received, including URLs from emails that had not yet been fully classified. For the purpose of this analysis, any URL not detected at that moment is labelled as undetected, even if it might be identified as malicious later through delayed evaluation of newly observed domains. URL-level detection scores simulate a real-time detection setting, where limited prior knowledge restricts the ability to consistently detect threats from new domains. In contrast, Figure 6 aggregates detection scores at the domain level. This removes inconsistencies between URLs belonging to the same domain, offering a clearer assessment of how effective the regex-based grouping is. Many syntactic pattern groups show a detection rate of 1, indicating consistent identification of malicious domain clusters. Groups remaining with mixed detection performance suggest that they contain a blend of unrelated benign and malicious domains.

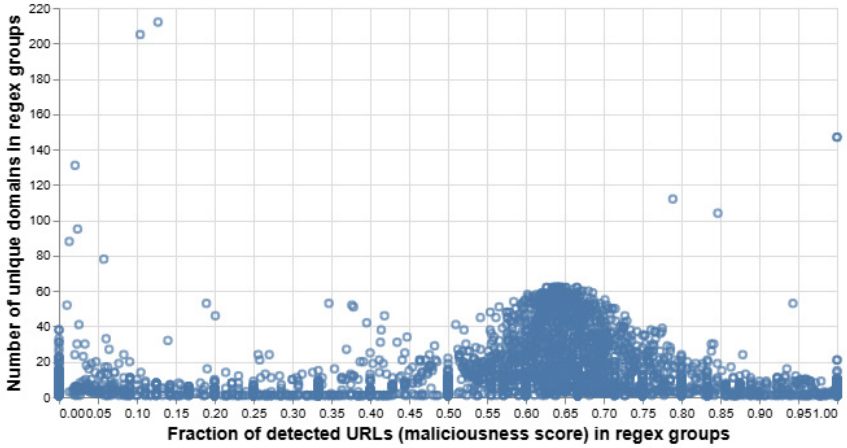

Figure 5: Fraction of detected URLs and number of unique domains in each regex group.

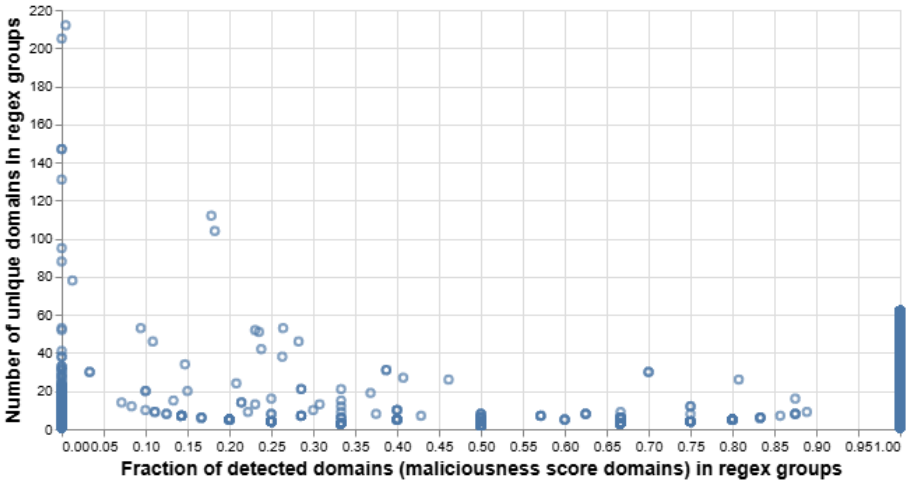

Figure 6: Fraction of detected domains and number of unique domains in regex groups.

Pure domain-based detection relies heavily on frequency-based features. A high spike in the number of URLs associated with a domain within a short time frame is highly suspicious and likely belongs to a malicious domain. Based on such peaks, one can detect most of the spam. However, the approach misses a significant portion of the initial mail volume for each new domain and is vulnerable to 'warm up' tactics: send a low volume of emails with a new domain before turning up the volume of URLs later. Warming up a domain aims to establish a good reputation for a domain before being used on a large scale to avoid potentially stricter checks for new domains. Sometimes cold emailing in marketing leverages the same strategy. Spam or phishing actors often combine the warm-up tactic with a delayed switch online of malicious content to avoid detection by crawlers used in pre-delivery protection. This technique further complicates immediate detection as initially there is no malicious content to find. Consequently, low-volume malicious domains are harder to detect. Syntactic patterns of new domains matched to already seen ones improve the early detection of domains in warm-up or stealth phase. Figure 7 illustrates a sending pattern marking the first time a new domain appears.

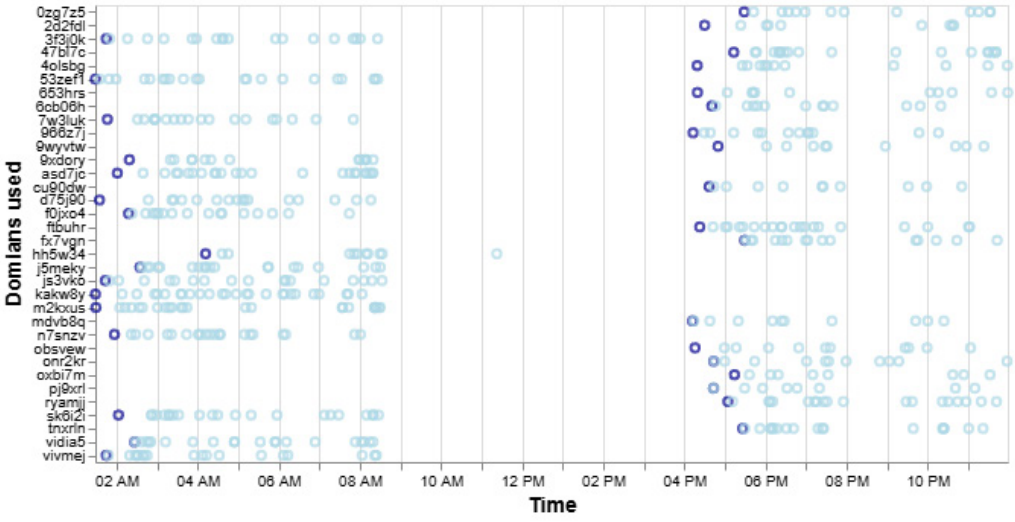

Figure 7: Recurring use of domains across a time span.

Experiment 1: Syntactic pattern matching

As outlined in the Methodology section, we first evaluate improvements of pre-filtering domains belonging to syntactic patterns with a high detection score before refining and classifying the resulting labels in our domain evaluation process. We define a minimum pattern length of 20, as this empirically produced highly uniform clusters. Empirical results indicate that filtering by already seen patterns increases detection rates by approximately 5-10% on average, while maintaining a comparably low false positive rate. This improvement is expected, as domain-level models – particularly those relying on aggregated or high-level features – typically achieve relatively low performance scores compared to models operating on full URL structures. In the context of large-scale spam operations, even a small number of missed domains can have a significant impact if they appear across a large number of URLs. Integrating syntactic pattern similarity early into the detection pipeline increases the likelihood of identifying all associated domains soon after the campaign begins.

One challenge of our approach lies in the small-scale structural variability of related URLs. Due to simplifications during pattern extraction, a single campaign may be fragmented into several pattern groups (Figures 5 and 6). Not all of these groups consistently capture the full set of associated domains. A second limitation relates to overly generic patterns composed of only a few structural blocks producing synthetic links between unrelated URLs. Many homepages, for example, direct to their German homepage by adding a '/de' path to their domain. Our approach would group all these homepages together, disregarding missing relations between all cases. Cases that match both benign and malicious link structures might misclassify domains, as illustrated in Figures 8 and 9. To mitigate this fragmentation, n-gram extraction consolidates pre-grouped related patterns and domains into fewer clusters per campaign and refines domain labels.

```
Analysing pattern: /ga/open/2
detected urls sum:138030
total urls:138870
domains detected by regex not in ngram: {'pizzeriaromaffm.de', 'swisslymph.ch'}
detected urls and domains:
shape: (34, 3)
```

domainstld --- str	url_example --- str	nr_unique_regexpat --- u32
aurreola.datahosterz.info	aurreola.datahosterz.info/ga/open/2-38522530-3893-…	1
cr65.hhpdp.com	cr65.hhpdp.com/ga/open/2-434172-22-28-60-fddf46860…	1
starfish.gamesieuhay.info	starfish.gamesieuhay.info/ga/open/2-4296638-275-57…	1
enucleator.swordsetsportal.info	enucleator.swordsetsportal.info/ga/open/2-27585186…	2

Figure 8: Cluster with two misclassifications with simple syntactic pattern matching.

```
not detected urls and domains:
shape: (2, 3)
```

domainstld --- str	url_example --- str	nr_unique_regexpat --- u32
swisslymph.ch	swisslymph.ch/de/news/orthoconcept-swiss-lymph-kid…	1
pizzeriaromaffm.de	pizzeriaromaffm.de/de/menu/0d2a9a79-6fbd-486a-86bb…	2

Figure 9: Misclassification cases with simple syntactic pattern matching.

The uniqueness of extracted n-grams varies considerably. Some n-grams capture highly distinctive elements, such as randomized strings (e.g. 'ztzOjQ6InN0YXQiO3M6M'); others conform with generic URL strings, such as common path segments (e.g. '/tracking/click'), which are less effective for campaign-level identification. Future work could explore automated techniques to categorize n-grams with high discriminative value. In the demonstrated approach, we restrict analysis to clusters with high detection rates to ensure the reliability of subsequent matching processes.

The specification of clusters using signature n-grams additionally reveals relevant patterns within groups initially excluded from detection due to low detection rates. As discussed in subsequent sections, temporal factors play a critical role in the detection of domains. The previously illustrated example contains instances of missed detections, which can often be attributed to URLs that had not yet undergone domain-based evaluation at the time of analysis. As a result, clusters with such URLs exhibit relatively low detection scores. In the Appendix we present a representative case, demonstrating that our approach can detect uniformly malicious clusters more rapidly than methods relying solely on domain-based detection.

It should also be noted that there are types of spam that involve only domains, for which this model provides little to no benefit. For instance, in 'prescription spam', emails often contain domains without lateral URL parts that change frequently and redirect to a more stable destination domain that hosts the spam content. Similarly, certain 'adult dating' spam messages make use of domain names that mimic URL shorteners, having shifted away from using legitimate URL-shortening services. However, these URLs often contain paths that are too short to allow for the construction of valid and reliable syntactic patterns, limiting the effectiveness of pattern-based detection methods.

Experiment 2: Syntactic feature generation for domain classification algorithms

To test informational value for machine-learning-based detection approaches, we select the highly interpretable logistic regression model to demonstrate the potential of extracted features from regex pattern matching. We apply custom sampling and transformation to improve model performance.

To construct robust training and testing datasets, we implement a stratified sampling strategy based on both structural and temporal characteristics of the data. The process uses the number of mails associated with each DomainTld as a weighting indicator to reduce sampling bias and improve generalization. We test the performance of our model on unseen data, from one week of unseen data. We delete any previously seen domains and report prediction performance on this test dataset in the section on model evaluation.

Before training the model, we use a custom feature engineering pipeline tailored for processing domain-related data. The transformation logic includes domain-specific pre-processing, such as computing derived statistical features from input fields (e.g. number of mails seen with domain and regex-based metrics), categorical encoding of top-level domains (TLDs) using predefined or dynamically inferred groupings, and binning temporal features. A keyword-matching mechanism extracts features based on matches against domain components using specified keyword groups, generating binary indicators for keyword presence. We apply log-scaling, threshold-based binarization, and one-hot encoding for various domain features, TLD, and behavioural features.

For our model we train a logistic regression classifier with an L1 regularization penalty to promote sparsity in the feature space, which helps in feature selection and enhances interpretability. Regularization strength is controlled to encourage stronger penalization of non-zero coefficients to avoid overfitting.

To assess the performance of our proposed URL detection method, we employ accuracy and F1 score as standard evaluation metrics derived from the confusion matrix. Accuracy measures the overall correctness of the model's predictions, calculated as the ratio of correctly predicted instances (true positives and true negatives) to the total number of instances:

Accuracy = (TP+TN) / (TP+TN+FP+FN)

Where: **TP** = True Positives; **TN** = True Negatives; **FP** = False Positives; **FN** = False Negatives

F1 score is the harmonic mean of precision and recall and is useful as a performance measure when dealing with imbalanced datasets, as it penalizes extreme values in either precision or recall:

F1 score = 2 * (precision * recall) / (precision + recall)

Where: **Precision** = TP / (TP + FP); **Recall** = TP / (TP + FN)

We test the modelling approach with two aggregation levels of features. The first experiment includes additional information based on the means of all regex patterns belonging to one domain. The second experiment further groups regex patterns by deleting the length of each regex block. This seems to more accurately attain fewer variations in syntactic patterns describing the same domain cluster. An additional feature included for the second experiment was the number of unique syntactic patterns belonging to the group. We compare the performance of a logistic regression model trained on the old feature set and the new feature set. As expected, the detection rate for each syntactic pattern group was key in improving model prediction performance.

Performance experiment with one regex pattern per group:

Model	**Accuracy**	**F1**
With new regex features	0.83	0.81
Without regex features	0.84	0.83

Performance experiment with grouped regex patterns disregarding length specifications:

Model	**Accuracy**	**F1**
With new regex features grouped	0.87	0.85
Without regex features	0.84	0.83

The results show that detection rates for shared regular expression (regex) patterns improve our model's performance with features on grouped regex patterns. As Figure 6 already showed, groups of regex patterns achieve a more pronounced separation between detection scores for malicious and benign patterns. Aggregated features on individual pattern levels encode information less specifically, and introduce ambiguity, as shown for detection scores in Figure 5. This reduces the model's prediction performance. Therefore, we highlight the importance of thoughtful feature aggregation when including pattern information in prediction models. More detailed features, such as specific character sequences, token lengths, or the presence of identifies n-grams, could offer an improved representation of structural characteristics of spam domains.

In the current detection system, we emphasize the importance of the initial matching process, which effectively filters out non-relevant examples before they reach the prediction model. This pre-processing step ensures that the initial logistic regression model operates on a cleaner dataset to maintain its efficiency and relevance. In conclusion, we would recommend optimizing the feature generation and selection process for standalone domain prediction algorithms in cases that do not have access to a continuous stream of data and that cannot use preceding pre-filtering and matching of domains.

DISCUSSION

We presented a novel approach to classify malicious URLs based on syntactic patterns. Our results on roughly a month's worth of real-world email metadata demonstrate that accounting for syntactic patterns of a URL improves detection rates of malicious domains. During our experiments, we also identified promising avenues for future work.

For example, one could design more efficient approaches to extract string patterns from URLs, such that they form more meaningful clusters. Especially when refining clusters by n-grams, automatic detection of non-generic patterns holds a high potential to improve cluster quality. Newsletter and other legitimate large-scale URL generation mechanisms make it difficult to distinguish clusters – especially when some are misused by spam actors. Excluding URLs with known newsletter patterns could further separate clusters. Other detection mechanisms might be required in these cases. Another important topic is how word detection can be employed to filter identifying string patterns and make regex patterns more unique.

REFERENCES

[1] Khonji, M.; Iraqi, Y.; Jones, A. Lexical URL analysis for discriminating phishing and legitimate websites. Proceedings of the 8th Annual Collaboration, Electronic Messaging, Anti-Abuse and Spam Conference (CEAS '11), 109–115. 2011. https://doi.org/10.1145/2030376.2030389.

[2] Mamun, M. S. I.; Rathore, M. A.; Lashkari, A. H.; Stakhanova, N.; Ghorbani, A. A. Detecting malicious URLs using lexical analysis. In Lecture Notes in Computer Science (Vol. 9955, pp. 467–482). Springer. 2016. https://doi.org/10.1007/978-3-319-46298-1_30.

[3] Butnaru, A.; Mylonas, A.; Pitropakis, N. Towards lightweight URL-based phishing detection. Future Internet, 13(6), 154. 2021. https://doi.org/10.3390/fi13060154.

[4] Pastika, P. B.; Alamsyah. Machine learning-based malicious website detection using logistic regression algorithm. EMACS Journal. 2024. Retrieved from https://journal.binus.ac.id/plugins/generic/pdfJsViewer/pdf.js/web/viewer.html?file=https%3A%2F%2Fjournal.binus.ac.id%2Findex.php%2FEMACS%2Farticle%2Fdownload%2F11844%2F5245%2F64136.

[5] Atrees, M.; Ahmad, A.; Alghanim, F. Enhancing detection of malicious URLs using boosting and lexical features. Intelligent Automation & Soft Computing, 31(3), 1405–1422. 2022. https://doi.org/10.32604/iasc.2022.020229.

[6] Darling, M.; Heileman, G.; Gressel, G.; Ashok, A.; Poornachandran, P. A lexical approach for classifying malicious URLs. In Proceedings of the 2015 International Conference on High Performance Computing & Simulation (HPCS) (pp. 195–202). 2015. IEEE. https://doi.org/10.1109/HPCSim.2015.7237040.

APPENDIX

Cluster with low detection rate, where detected and not detected domains are all spam-related.

```
Analysing pattern: splay.php?
detected urls sum:174
total urls:3210
domains detected by ngram: 16
domains detected by ngram not in regex: set()
domains detected by regex: 16
domains detected by regex not in ngram: set()
shape: (16, 4)
```

domainstld	url_example	domain_detected	nr_unique_regexpat
str	str	i32	u32
distinctneeds.ovh	distinctneeds.ovh/iem/display.php?M=738015&C=66736…	0	1
hiraschateautailoring.com	hiraschateautailoring.com/iem64/display.php?M=4343…	0	1
subscribermailing.info	subscribermailing.info/display.php?M=18193&C=6c61a…	0	2
hirasplatinumtailor.com	hirasplatinumtailor.com/iem64/display.php?M=259356…	1	1
hirasluxuryweave.com	hirasluxuryweave.com/iem64/display.php?M=44709463&…	0	1
…	…	…	…
hirassilkroadtailors.com	hirassilkroadtailors.com/iem64/display.php?M=45079…	1	1
hirasdiamondmeasure.com	hirasdiamondmeasure.com/iem64/display.php?M=437961…	1	1
online-advertising.website	online-advertising.website/iem/display.php?M=70954…	0	1
subscribermailing.online	subscribermailing.online/display.php?M=19151&C=f80…	0	1
hirasnoblecuts.com	hirasnoblecuts.com/iem64/display.php?M=515177&C=da…	1	2

unique regex patterns bad and good linked by ngram:
shape: (7, 4)

domainstld	url_example	domain_detected	nr_unique_regexpat
str	str	i32	u32
hirasplatinumtailor.com	hirasplatinumtailor.com/iem64/display.php?M=259356…	1	1
hirasregalbespoke.com	hirasregalbespoke.com/iem64/display.php?M=3221135&…	1	1
hirasexclusivesilhouette.com	hirasexclusivesilhouette.com/iem64/display.php?M=5…	1	1
app.angricky.com	app.angricky.com/display.php?M=888976&C=aea72ea4aa…	1	1
hirassilkroadtailors.com	hirassilkroadtailors.com/iem64/display.php?M=45079…	1	1
hirasdiamondmeasure.com	hirasdiamondmeasure.com/iem64/display.php?M=437961…	1	1
hirasnoblecuts.com	hirasnoblecuts.com/iem64/display.php?M=515177&C=da…	1	2

shape: (9, 4)

domainstld	url_example	domain_detected	nr_unique_regexpat
str	str	i32	u32
distinctneeds.ovh	distinctneeds.ovh/iem/display.php?M=738015&C=66736…	0	1
hiraschateautailoring.com	hiraschateautailoring.com/iem64/display.php?M=4343…	0	1
subscribermailing.info	subscribermailing.info/display.php?M=18193&C=6c61a…	0	2
hirasluxuryweave.com	hirasluxuryweave.com/iem64/display.php?M=44709463&…	0	1
email-blast.online	email-blast.online/display.php?M=494458&C=d68914f3…	0	3
7bdadccbc090.preparadoja.com.br	7bdadccbc090.preparadoja.com.br/painel/display.php…	0	1
hirasimperialmeasure.com	hirasimperialmeasure.com/iem64/display.php?M=45293…	0	1
online-advertising.website	online-advertising.website/iem/display.php?M=70954…	0	1
subscribermailing.online	subscribermailing.online/display.php?M=19151&C=f80…	0	1