Universiteit Leiden

Master Computer Science

GitHub PoC Early Attention Trend as a Signal for Vulnerability Triage

Name:	Xinzhe Tian
Student ID:	s3335488
Date:	21/08/2025
Specialisation:	Advanced Computing and Systems
1st supervisor:	Dr. O. Gadyatskaya
2nd supervisor:	Dr. K.F.D. Rietveld

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Contents

Abstract

Software vulnerabilities represent a systemic security risk. Each year, tens of thousands of new Common Vulnerabilities and Exposures (CVEs) are published, yet only a small fraction are widely exploited. Conventional triage signals such as the Common Vulnerability Scoring System (CVSS) and the Exploit Prediction Scoring System (EPSS) are widely adopted for prioritisation, but remain constrained by their reliance on centralised metadata and delayed enrichment, a fragility exposed by the 2024 National Vulnerability Database (NVD) backlog.

This thesis explores whether early ecosystem attention on GitHub-hosted proof-of-concept (GHP) can serve as a forecasting signal for vulnerability prioritisation. Using gradient-boosted regression (XGBoost) combined with density-based clustering (HDBSCAN), the study demonstrates that early engagement trends, capturing both interaction volume and behavioural archetypes, are effective predictors of long-term GHP attention. This prediction target, in turn, could function as a real-time indicator of risk significance. The proposed framework departs from static scoring by leveraging decentralised ecosystem engagement, is sensitive to emerging momentum, and is capable of surfacing high-risk CVEs, positioning early GitHub PoC activity as an effective signal for vulnerability triage.

Acknowledgement

I would first like to thank my supervisor, Olga Gadyatskaya, and my mentor, Jafar Akhoundali, for guiding me through my first steps into research, and for their patience and encouragement as I learned to find my way.

To Olga: thank you for your unwavering support from beginning to end, for helping me search for a research topic, introducing me to researchers working in related fields, and consistently providing invaluable material, advice, and guidance throughout this thesis. Above all, thank you for being such a kind, understanding, and inspiring presence. I feel fortunate to have you as my supervisor.

To Jafar: thank you for placing your trust in me with this topic and inviting me into a larger study. I am grateful for your advice at every stage, the knowledge you shared so generously, and the help you never hesitated to offer. I have learned so much from our discussions. Your steady reassurance that I was on the right track is what gave me the confidence to keep moving forward.

I also wish to thank Dr. Kristian Rietveld for taking the time to serve as my second reader and for the valuable advice he provided.

To my family, and especially my mum: I cherish the glimpses of life you always shared with me, the photographs you took, the films you were watching. They were a quiet yet vital support, a spark that helped me greet each new day. To my friends: Chloe, thank you for your unconditional support and encouragement; Jiaqi, for the songs you added to our playlist over time, each carrying meaning and strength; and Nuo and Huizi, for staying in touch even across countries and continents. To the friends I met in the library: I will remember the midday chats over coffee, the laughter we shared when caught in the morning rain, and the librarians and familiar faces whose daily smiles and greetings made the workdays lighter.

To the plants I cared for alongside this thesis: thank you for being steadfast and radiant, growing strong and lively, you are the best home buddies. To Vivaldi, Ravel, Thelonious Monk, and all the other great voices of music whose work has endured, to my piano, to literature, and to art in all its forms: thank you for keeping me company in the quiet mornings and the long nights alike.

"Solvitur ambulando" — it is solved by walking. This has long been my favourite motto, and it is what carried me through to the completion of this research. To all those who walked beside me on this journey, I hold a lasting and heartfelt gratitude.

Chapter 1

Introduction

In our increasingly digital society, software vulnerabilities represent a structural risk. A single vulnerability in a widely deployed component can send ripples across entire industries, while others can trigger severe and long-lasting consequences. For example, Log4Shell (`CVE-2021-44228`), a remote code execution (RCE) flaw in the widely used Java logging library Log4j, exposed numerous organisations ranging from cloud platforms to enterprise applications. Similarly, the recent Microsoft SharePoint vulnerability (`CVE-2025-49706` and related chains) was exploited as a zero-day in widespread campaigns, allowing attackers to achieve RCE, establish persistent access, and extract cryptographic keys to forge authentication tokens. The deliberate and capable nature of this campaign forced organisations to take urgent action and mitigate widespread risk across internal networks.

However, the complexity of modern systems makes vulnerabilities both inevitable, and rapidly increasing in volume. As of mid-August 2025, over 30,500 CVEs (Common Vulnerabilities and Exposures) had been published, marking a 43% increase compared to the same point in 2024[1]. Given limited resources and operational constraints, it is infeasible for organisations to patch every vulnerability. As a result, prioritising remediation has become an essential part of modern cybersecurity practice, requiring defenders to determine which vulnerabilities demand immediate attention and which can be safely deferred.

Historically, vulnerability triage has relied on centralised metadata sources and static risk scores, most notably the Common Vulnerability Scoring System (CVSS). These systems offer a structured, theoretically grounded view of severity but often fall short in capturing which vulnerabilities are likely to be targeted in practice. The ecosystem's dependency on the National Vulnerability Database (NVD) further introduces a bottleneck: the 2024 enrichment backlog at NVD, during which over 93% of newly published CVEs remained un-analyzed for months, exposed the systemic fragility of metadata-driven triage pipelines.

In response, recent research has moved toward more dynamic models. Some approaches enrich traditional scoring with additional metadata, incorporating features such as exploit availability, patch timing, and publication delay. Others, such as the Exploit Prediction Scoring System (EPSS), moves a step further, using probabilistic forecasting to estimate short-term exploitation likelihood based on a broad and frequently updated feature set. While these enriched models offer measurable improvements over static baselines, they remain fundamentally dependent on structured metadata and post-disclosure

[1] `https://cvefeed.io/vulnerability-cve-metrics/`

artefacts that may be delayed, incomplete, or unavailable in the early stages of a CVE's lifecycle.

This thesis proposes a complementary approach: forecasting CVE risk based on early engagement trends with GitHub-hosted Proof-of-Concept (GHP) repositories. By leveraging behavioural signals within the first days following GHP activation, the proposed model aims to identify high-attention CVEs in real time, independently of static scores or curated metadata. In contrast to traditional systems, which rely on post-disclosure scoring and curation before operation, our model focuses on decentralised ecosystem engagement as an early indicator of risk significance.

Figure 1.1: Overview of the proposed framework.

Figure 1.1 provides an overview of the approach. It begins with a global prediction model to set the baseline, proceeds with trend-shape clustering to identify early attention archetypes, and concludes with cluster-wise models that refine predictions where the global view falls short. Together, these steps form a triage framework that is robust to metadata delays, responsive to real-world momentum, and capable of supporting early, evidence-driven prioritisation.

Chapter 2

Background and Related Work

2.1 NVD and CVSS: Foundations for Data-Driven Vulnerability Triage

Software vulnerability triage plays a central role in modern cyber defence. Given the continuous influx of newly disclosed vulnerabilities, many of which are never exploited, organisations must determine which CVEs warrant urgent attention and which can be safely deprioritised. Effective triage enables timely and resource-efficient remediation, reducing potential exposure by focusing on vulnerabilities most likely to be weaponised or actively targeted.

The National Vulnerability Database (NVD) remains the primary source of curated vulnerability metadata. Each CVE entry typically includes a natural language description, a CVSS Base score, and structured enrichment such as CWE (Common Weakness Enumeration) and CPE (Common Platform Enumeration). CVSS, the Common Vulnerability Scoring System, provides a standardised measure of technical severity and has become the default input for triage tools in both enterprise and academic contexts.

Although the CVSS specification includes optional temporal and environmental vectors, these are rarely populated or maintained. In practice, most public CVEs report only the fixed base score, and this is what most security vendors and researchers rely on for risk assessment. However, a consistent body of research demonstrates that the CVSS base score alone is insufficient for predicting real-world exploitation or supporting actionable prioritisation.

Both [Allodi and Massacci, 2014] and [Younis and Malaiya, 2015] evaluate the effectiveness of CVSS v2 scores as indicators of real-world exploitability, and both arrive at the same conclusion: CVSS provides poor predictive value. Allodi and Massacci show that CVSS-based prioritisation performs close to random selection, while incorporating PoC presence improves accuracy to around 45%. Younis and Malaiya further demonstrate that converting CVSS scores into a binary exploitability classifier yields extremely low precision , as low as 7% for Internet Explorer, indicating that high CVSS scores frequently overestimate exploitation risk.

In later work, [Suciu et al., 2022] offer further critique of CVSS v3 exploitability metrics, showing that they perform poorly as predictors of real-world exploit development. In a large-scale evaluation, they demonstrate that CVSS v3 exploitability scores yield a maximum precision of just 0.19 , meaning over 80% of vulnerabilities flagged as "likely to be exploited" never are. Through concrete examples, such as CVE-2018-8174 and CVE-

2018-8440, both exploited in the wild yet assigned exploitability scores below the 10th percentile, the authors show that CVSS systematically misses active threats. They argue that this imprecision stems from the design of CVSS itself, which relies on pre-disclosure technical analysis and evaluates vulnerabilities statically and in isolation.

These empirical failures stem from CVSS's structural limitations. As the CVSS v3.1 user guide itself concedes, "CVSS is designed to measure the severity of a vulnerability and should not be used alone to assess riskassess risk"[1]. By design, the base score omits temporal, environmental, and contextual factors, making it ill-suited for dynamic risk assessment. While CVSS v3 introduced more granular base metrics, it remains a snapshot system, lacking mechanisms for updating risk as new exploit artefacts or attack signals emerge.

While CVSS's static design limits its ability to reflect evolving threat conditions, an even broader vulnerability lies in the ecosystem's dependence on centralised metadata sources, NVD's role as the primary enrichment pipeline makes it a single point of failure. This structural fragility became plainly visible in early 2024, when a severe backlog in NVD start to take place. Beginning on February 12, NVD drastically slowed its processing of new CVEs. Between February and late May, 12,720 vulnerabilities were published, but 93.4% remained unanalyzed[2]. Among them were 50.8% of Known Exploited Vulnerabilities (KEVs) and 82% of CVEs with public PoCs, leaving security teams blind to critical signals of active risk.

The backlog's roots were structural: limited public-funding, surging CVE volumes (over 33,000 in 2023), and a centralised, labour-intensive enrichment pipeline[3]. While CVSS score coverage shows good recovery in late 2024, aided by external sources like CISA's *Vulnrichment* GitHub feed, a large number of CVEs still lacked full analysis[4].

This systemic slowdown fractured the long-standing notion of NVD as a "single source of truth" for vulnerability metadata. Many enterprise systems that depend on NVD for CVSS scores and CPE tags, including SIEMs, scanners, and risk dashboards, were left with placeholders, delayed updates, or missing entries altogether. Industry voices, including IBM and VulnCheck, have since argued for a paradigm shift toward more decentralised and flexible triage approaches[5]

2.2 Enriched Vulnerability Triage Models

There has been a growing shift in recent research toward enriching traditional vulnerability scoring models with broader lifecycle and ecosystem-aware features. More recent triage systems attempt to improve upon static scoring by incorporating additional signals that better reflect real-world risk. These enhancements typically include metadata-derived features, exploit artefact indicators, and machine learning and heuristic models that infer exploit likelihood.

An increasingly prominent scoring standard is the Exploit Prediction Scoring System (EPSS, [Jacobs et al., 2021]). EPSS is explicitly designed to estimate the likelihood of exploitation in the near term (e.g., 30 days), using a probabilistic model updated daily. It draws on a large feature set, reportedly over 1,400 attributes, including CVSS vectors,

[1] `https://www.first.org/cvss/`

[2] `https://www.vulncheck.com/blog/nvd-backlog-exploitation`

[3] See ReversingLabs blog post on April 2, 2025.

[4] `https://www.vulncheck.com/blog/nvd-backlog-exploitation-lurking`

[5] See IBM Article: CNVD backlog update: Attackers change tactics as analysis slows.

public exploit availability, and social media mentions. This focus on dynamic threat signals makes EPSS a more adaptable alternative for short-term triage.

As previously noted, [Suciu et al., 2022] critique the static nature of CVSS v3, arguing that it fails to capture real-world exploitability due to its reliance on isolated, pre-disclosure technical analysis. In contrast, they emphasise that post-disclosure artefacts , such as proof-of-concept code, documentation, and social media discussion, provide more actionable signals for forecasting exploit development. Building on this observation, they introduce Expected Exploitability (EE), a supervised learning framework designed to predict whether a functional exploit will emerge for a given CVE. Their feature set incorporates PoC artefacts extracted from ExploitDB, BugTraq, and Vulners, including programming language, code complexity metrics, reserved keyword usage, and textual n-grams from both code and commentary. When combined with contextual features such as CVSS scores, CWE types, and vendor/product data, these models outperformed all baselines, suggesting that PoC artefacts meaningfully enhance exploit prediction beyond what static metadata alone can offer.

[Zhang and Li, 2020] approach exploit prediction as a temporal classification task, estimating the probability that a CVE will be exploited on each individual day following disclosure. Input features include CVSS base metrics, CWE categories, software identifiers, and tf-idf embeddings of CVE descriptions. Notably, they also extract timeline features, including the difference between the modified date and the original published date for the vulnerability, and the difference between the last seen date and the original publish date. While this design brings useful temporal context to post-disclosure risk modelling, the inclusion of retrospectively observable features, such as `last_seen_date`, raises concerns of data leakage, limiting its applicability for real-time vulnerability prioritisation.

[Costa and Tymburibá, 2022] propose V-REx, a neural network-based system for predicting vulnerability exploitability. The framework integrates time-aware features such as vulnerability age, patch-to-exploit delay, and structured risk windows into a supervised learning setup. V-REx uses three neural network variants (standard, enhanced, and interconnected-enhanced) to classify vulnerabilities, drawing on CVE/NVD-derived metadata, CVSS scores, and NLP-processed textual features, and then incorporates an enhanced genetic algorithm for hyperparameter tuning. By embedding temporal context directly into the learning architecture, the authors report improved performance relative to both CVSS and EPSS, suggesting that dynamic modelling yields stronger predictive signals than static scoring alone.

A more recent study ([Khanmohammadi et al., 2025]) introduces ExploitabilityBirthMark, a static classifier that predicts whether a CVE will eventually be exploited using only information available at day zero. Unlike EPSS, it bypasses enrichment fields that are often missing during disclosure backlogs, such as CVSS or CPE, and instead derives NVD report features from summary-level cues, vendor and product mentions, and lightweight external signals such as vendor size or open-source status. Implemented with an XGBoost model tuned via grid search, BirthMark markedly outperforms a stripped-down Day-1 EPSS baseline: prioritising the top 30% of CVEs by its score captures about 70% of those later exploited, compared to only 40% with Day-1 EPSS. While this static approach sacrifices temporal nuance, it offers a counterpoint to EPSS's reliance on progressively accumulated data, pushing towards early-stage prediction tools that are less dependent on centralised metadata flows.

2.3 Social Media as Emerging Risk Signal Platforms

The vulnerability risk landscape continues to evolve through years. While prior work has relied on structured sources like CVE metadata, exploit datasets, and Symantec threat feeds, enriching NVD-based scores and metrics with wider temporal and contextual features, a growing body of research is turning their main focus to social platforms as alternative signal sources. Platforms like Twitter, GitHub and Reddit are gaining attention for their ability to reflect early, decentralised indicators of exploit activity, including researcher interest, PoC publication, and threat actor engagement.

Multiple recent surveys on vulnerability triage studies echo this shift in thinking.[Jiang et al., 2025]., when discussing on data source for vulnerability prioritisation research, specifically call for dynamic, time-sensitive indicators, noting that *"emerging sources, such as social media (Twitter, GitHub), offer real-time insights into exploit announcements."* Similarly,[Le et al., 2023] highlight that GitHub remains an underutilised yet promising source of early vulnerability signals, and advocate for improved integration of behavioural indicators into triage frameworks.

A line of research has explored the viability of Twitter as an early warning system for vulnerability exploitation. [Sabottke et al., 2015] (2015) first demonstrated that vulnerability discussions on Twitter could be leveraged to predict real-world exploits, using support vector machines trained on tweet text and metadata to flag CVEs of interest. Building on this foundation, [Chen et al., 2019] proposed an ensemble-based approach to forecast when a CVE would be exploited, linking tweet activity patterns to both PoC and real-world exploitation timelines. Their graph-based model showed that the velocity and structure of early tweet dissemination can serve as temporal cues for exploit likelihood.

More recent efforts have advanced both data extraction fidelity and predictive robustness. [Du et al., 2023] developed ExpSeeker, a deep-learning pipeline that automatically identifies tweets containing public exploit code and extracts relevant metadata such as CVE identifiers and vulnerability types. Their method often detects exploits ahead of ExploitDB, highlighting Twitter's decentralised and real-time signal advantage. In parallel, [de Sousa et al., 2020] evaluated multiple classifiers on a five-year Twitter dataset, and found that Twitter metadata and user statistics (e.g., follower counts, retweet activity) generally outperformed tweet content for exploit detection, highlighting the value of engagement signals in social-media–based triage. Collectively, these studies showcase Twitter's potential as a high-tempo signal source for triage, while also reveal noise and temporal drift challenges that constrain long-term reliability.

Alongside Twitter, GitHub has also become an emerging platform for deriving vulnerability triage signals. EPSS v3 has already included GitHub as a source of publicly available exploit code besides Exploit-DB and MetaSploit[6]. In addition, several recent studies have specifically investigated GitHub's potential as a risk-signalling platform, recognising its role in reflecting emerging exploit activity and early ecosystem attention.

[Shrestha et al., 2020] presented an in-depth contrastive analysis of discussion spread about software vulnerabilities in three social platforms,GitHub, Twitter, and Reddit. They highlight GitHub as a primary surface for vulnerability-related discussion, finding that in 46% of cases, conversation begins on GitHub, and in over 16% of cases, even before CVEs are published to the NVD. This positions GitHub not merely as a code-hosting site, but as a real-time situational awareness platform for developers, security engineers, and potentially adversaries. Notably, they also find that discussions about CVEs later linked

[6] `https://www.first.org/epss/model)`

to state-sponsored APT campaigns, including Russian, Chinese, and Iranian operations, frequently begin on GitHub.

A recent risk scoring framework, XVRS, proposed by [Seker and Meng, 2023], directly incorporates GitHub-derived metrics as part of its dynamic threat intelligence model. Specifically, they include a composite score based on the number of GitHub repositories, stars, and forks associated with a vulnerability, aggregated over the most recent three-month period. While XVRS acknowledges the correlation between GitHub activity and emerging risk, it uses these features purely as volume-based augmentations.

[Kita et al., 2025] further underscores GitHub's growing role in the vulnerability lifecycle, recognising that GitHub has grown into a major repository of exploit code, often exceeding ExploitDB and Metasploit in coverage and timeliness. They propose a prioritisation framework for exploit codes published on GitHub to support more effective vulnerability triage. Specifically, they adpoted a graph-based scheme for prioritising trustworthy exploit codes on GitHub. Their approach constructs a trust graph among users based on repository stars and follower links, and seeds this graph using authors referenced in NVD, ExploitDB, or Metasploit. By applying TrustRank, they rank exploit codes by inferred credibility, framing GitHub as a viable, though noisy, source of real-world exploitation signal.

These findings reinforce the premise of this thesis: that GitHub PoC activity and engagement patterns are not merely passive or retrospective, but reflect a decentralised, actor-diverse form of ecosystem attention that may precede formal recognition by scoring systems or curated threat databases. It can be a signal platform for emerging vulnerability risk, and is in need for automated triage mechanisms.

2.4 Positioning and Methodological Framing

Prior research has made significant strides in vulnerability scoring, exploit prediction, and the use of GitHub data. This section situates the present work within that landscape. We begin by framing GitHub interactions as meaningful indicators of early interest. We then compare this thesis's scope and assumptions with two adjacent studies that also analyse GitHub-hosted PoC repositories, but from notably different perspectives. Finally, we outline the method framing and design considerations of this work, particularly around temporal modelling and interpretability, to distinguish it from existing vulnerability triage approaches.

2.4.1 GitHub and User Interaction Mining

In practice, GitHub operates as a platform for collaborative software development, structured around repositories that host source code and related artefacts. User interactions form the basis of observable activity on GitHub. Accounts engage with repositories in several means: starring is a way to bookmark or endorse a project, signalling attention and visibility; watching subscribes a user to updates and notifications; and forking creates a personal copy to experiment or prepare contributions without affecting the upstream repository. Because these actions are logged with timestamps, they make project activity directly traceable over time, enabling large-scale, behaviour-centric measurement.

Empirical studies of GitHub interaction factors driving project popularity ([Borges et al., 2016], [Borges and Tulio Valente, 2018]) show that developers often treat stars as a decision heuristic: in a survey of 791 developers, roughly 73% reported considering star

counts when deciding whether to try or contribute to a project. This establishes stars as a practical proxy for community attention. Stars and forks are also strongly correlated ($\rho \approx 0.55$), with forks reflecting a higher-effort form of engagement. Importantly, starring behaviour exhibits recognisable temporal patterns around release cycles and version updates, giving rise to distinct growth regimes: slow, moderate, fast, or viral. These patterns support the modelling of early star/fork trajectories as predictive signals, rather than relying solely on raw totals.

Prior studies routinely mine user-interaction traces at scale to answer broader software engineering questions. For example, [Bissyandé et al., 2013] analyse watchers, forks, issues, contributors, and team size across 100,000 repositories to study language ecosystems and project outcomes. Similarly, [Parekh, 2024] links GitHub stars with PyPI downloads for 3,182 Python libraries, incorporating interaction and maintainership features. These exemplars demonstrate that stars, forks, and related interactions are established, queryable signals for empirical analysis, supporting their use here as early attention and engagement indicators.

2.4.2 Adjacent study on GitHub Proof-of-Concept (PoC) repositories

As discussed in Section 2.3, [Kita et al., 2025] propose a prioritisation framework for GitHub-hosted exploit code, focusing on code trustworthiness and social credibility. Their approach combines clone de-duplication with a graph-based trust scoring scheme, resulting in a curated set of PoCs presumed to be credible and exploit-worthy, making the framework post-hoc, trust-filtered, and contributor-centric.

In parallel, [Yadmani et al., 2022] present a large-scale empirical investigation into the maliciousness of CVE-linked PoC repositories hosted on GitHub. Their work frames PoCs as potential threats to analysts and the community, applying static malware heuristics (e.g., IP addresses, obfuscation patterns, VirusTotal scores) to detect repositories that embed or disguise harmful content. While their findings confirm that GitHub activity is not inherently benign, their approach frames PoCs primarily as *objects of threat containment*.

In contrast, this thesis adopts a broader and signal-oriented perspective. Rather than filtering for functionality or intent, we treat all CVE-tagged repositories as potential sources of behavioural signal. Interaction dynamics, such as star, and fork events, are used not to validate the correctness of the PoC, but to infer ecosystem attention patterns, in which actors could be user patching third-party sources [Schiappa et al., 2019], reputable exploit code contributors [Kita et al., 2025], opportunistic attackers, or momentary passive observers drawn in by trending vulnerabilities.

Since any form of decentralised coordination or early interest may signal urgency, we argue that GitHub PoC timelines can offer early, actor-agnostic insight into which vulnerabilities are likely to matter. This thesis thus shifts the analytical lens from post-hoc PoC trustworthiness to early attention trajectories, foregrounding behavioural patterns over static code attributes.

2.4.3 Design considerations for vulnerability triage models

Temporal Feature Design

Many prior models incorporate time as a static input feature. For example, [Zhang and Li, 2020] extract attributes such as the difference between a vulnerability's last seen date and its original disclosure date, using the Vulners database. Similarly, [Costa and Tymburibá, 2022] include measures like vulnerability age and patch-to-exploit delay to define risk windows. EPSS also uses a timeline indicator, the number of days since CVE publication, which ranks among its top 30 predictive features.

In contrast, this thesis treats time not as a fixed attribute but as a dynamic structure: it models how engagement unfolds within a defined early observation window. The model leverages daily interaction signals (e.g. stars, forks) to capture behavioural growth characteristics over time, offering a more granular and temporally expressive foundation for forecasting vulnerability risk.

Temporal Prediction Targets

When it comes to the temporal framing of prediction targets, existing models take varied approaches.

[Suciu et al., 2022] define *expected exploitability* as the probability that a functional exploit will be developed for a CVE over time. While the prediction itself is static, the model is evaluated at multiple post-disclosure checkpoints (e.g. Day 0, 10, 30, 365), demonstrating that the earliest post-disclosure artefacts tend to carry the most predictive value. Whereas [Zhang and Li, 2020] embed time directly into the prediction target by training a sequence of neural networks to estimate the likelihood that a CVE will be exploited by day n after disclosure. Their model learns day-by-day risk progression, with particular focus on the first 30 days post-disclosure.

Rather than focusing solely on the disclosure-to-exploit interval, this thesis models ecosystem attention to GitHub-hosted PoCs (GHPs), spanning a broader scale of the vulnerability lifecycle. The proposed model anchors each CVE at the point of first GitHub PoC activation and predicting future engagement trajectories over 30, 60, and 90 days. This approach captures a richer behavioural context, aligning actor engagement whether they occur before, during, or long after formal CVE publication.

Temporal Data Splitting

[Le et al., 2023] observe that among their reviewed studies, k-fold cross-validation , which allows each data point to appear in both training and validation roles, remains a dominant evaluation strategy. However, they pointed out that such methods lack a truly held-out test set and fail to reflect the conditions of real-world deployment.

To address this, the present study adopts a temporally aware data split. The dataset is divided into a training set, an additional validation set for hyperparameter tuning, and a strictly time-separated test set, all ordered chronologically based on each CVE's publication date. This approach better reflects the evolving nature of the ecosystem, ensuring that the model learns from past trends and is evaluated on future, previously unseen CVEs.

In addition, this thesis's temporal split is designed to align with the 2024 NVD backlog: the model is trained on CVEs disclosed prior to March 2024, and evaluated on those published during the backlog period, when NVD enrichment slowed or stalled. This setup serves as a real-world stress test, allowing us to evaluate whether temporal GitHub PoC attention trends can provide meaningful risk signals amid broader ecosystem shifts. A model that performs well under these conditions would demonstrate potential for resilient, decentralised triage, acting as a buffer for identifying emerging threats before formal

scores become available.

Interpretability

Beyond calls for time-aware, dynamic, and alternative risk signals, [Le et al., 2023] and [Jiang et al., 2025] have also identified the lack of model interpretability as a major barrier to realistic adoption of predictive models. As detailed in later sections, our study addresses this challenge using a gradient-boosted regression model (XGBoost), paired with SHAP-based post-hoc analysis to provide interpretability. This approach enables inspection of why certain CVEs are prioritised, offering deeper diagnostic insight into model behaviour.

In summary, this thesis propose a behavioural forecasting framework for CVE triage, grounded in early GitHub Proof-of-Concept (PoC) activity patterns. It contributes to the literature in the following dimensions:

- **Dynamic & Alternative Risk Modelling:** Unlike traditional approaches that rely on static CVE metadata or treat time as a fixed attribute (e.g., delay since disclosure), this work adopts a dynamic lens, forecasting CVE popularity based on *how GHP engagement unfolds over time*. By treating user interactions (stars, forks, etc.) as a behavioural time series within defined observation windows, the model captures temporal momentum rather than just static volume. GitHub attention trends thus serve as a decentralised and real-time signal source, offering predictive value even when conventional metadata (e.g., CVSS, EPSS) is incomplete, delayed, or absent.
- **Forecastability from Early Attention Shape:** By constructing a GitHub PoC prediction model and conducting comprehensive experiments across multiple input–output horizons, this thesis demonstrates that short observation windows (5–7 days) are sufficient to support accurate long-range forecasts of PoC popularity (30–90 days). SHAP-based interpretability analysis further confirm that early GHP interactions, both in volume and in temporal shape, carry strong predictive signal for downstream engagement.
- **Behavioural Clustering & Subpopulation Modelling:** Recognising the limitations of global models in the face of attention-pattern heterogeneity, this thesis introduces a trend-based clustering pipeline to segment CVEs by early PoC engagement dynamics. Subsequent cluster-wise analysis and targeted prediction enhancements, including extended observation windows, shape-based / contextual feature enrichment, and skew-aware regression models, are conducted to improve performance across distinct temporal archetypes.
- **Alignment with Real-World Exploitation:** Beyond numerical accuracy, the thesis evaluates the model in a ranking scenario and validates its predictions against external threat signals, including Metasploit modules, KEV listings, and ransomware leak datasets. The results show that high-risk CVEs often exhibit measurable GitHub PoC activity *before* these signals emerge, and the model captures the top-priority CVEs well, underscoring the its potential for vulnerability triage and early warning in real-world defence workflows.

Together, these contributions position the present work as a lightweight, transparent, and temporally grounded alternative to static scoring systems, offering a practical path toward decentralised vulnerability triage.

Chapter 3

Database and Feature Foundations

3.1 Dataset Structure

This study draws on a prepared dataset compiled by prior researchers, integrating NVD CVE metadata with GHP repositories and user-level interaction events. The observation window extends through 18 March 2025, and no repo creation or interaction beyond this date is included. For NVD coverage, a CVE is considered in scope if it was (a)published on or before 18 March 2025 or (b)linked to at least one PoC repository created on or before that date, thereby preserving cases of pre-disclosure PoCs. All timing analyses that reference a CVE "publish date" require a non-null NVD publication timestamp.

The dataset also incorporates external exploitation sources, including CISA & VulnCheck KEV (Know Exploited Vulnerabilities), ExploitDB, Metasploit, and ransomware-related CVE lists which will be used later in the thesis to provide context and downstream evaluation. Records from ExploitDB and Metasploit are gathered and maintained by the present author. All data entries with a timestamp are are truncated at the cut-off date to preserve temporal consistency. The detailed integration methodology and underlying rationale will be presented in Section 7.2.

3.1.1 GHP-Specific Data

To capture early GHP activity, the dataset identifies GitHub repositories that explicitly reference individual CVE identifiers in visible metadata fields, namely repository titles, descriptions, README files, and topic tags. This ensures that the reference is both deliberate and publicly documented. In order to maintain analytical clarity, this research exclude GHP repositories referencing multiple CVEs, enforcing a one-to-one mapping between each CVE and its associated GHP repository.

Furthermore, the dataset is structured to support time-series analysis of how interest in a vulnerability emerges and evolves, often within days of initial disclosure or GHP activation. In addition to repository-level metadata such as repo_created_at, the dataset records individual user interactions (stars or forks) with timestamp and user ID. These fine-grained traces allow us to reconstruct per-CVE engagement timelines and behavioural patterns.

The final portion of Table 3.1 outlines CVE-level aggregates derived from user interaction traces. These metrics form the foundation for feature engineering introduced in Section 4.1.3.

Table 3.1: Database schema of the prepared dataset, including CVE metadata, PoC repositories, User interactions and CVE-level aggregates.

Object / Table	Field	Type	Description / Semantics
NVD CVE	id	text	CVE identifier.
	status	text	CVE status code (e.g., P, PU, R, U. D-rejected entries are excluded).
	date_published	date	NVD publication date (may be NULL for R/U or incomplete entries).
	CVSS_score, version, base_severity, vector_string	int, text	CVSS version, base score, severity, and CVSS vector.
GHP Repository	id	int	Internal GHP repository identifier.
	name, repo_link, github_id	text	Repository metadata and URL.
	about, readme	text	Snapshot textual metadata used in PoC discovery.
	repo_created_at	timestamptz	GitHub repository creation time (GHP "publication").
	cve_count	int	Number of CVEs referenced by this repository.
	stars, forks	int	Repo-level counters used for summaries. (Watches not recorded—see limitation.)
CVE-PoC Mapping	cve_id, poc_id	text, int	Many-to-many mapping from CVEs to PoC repositories.
User Interaction Events	cve_id	text	CVE receiving the event.
	user_id	text	GitHub user id; -1 denotes repository creation (sentinel, not a user).
	interacted_time	timestamptz	Event timestamp (repo creation, star, or fork).
CVE-level Aggregates	cve_id	text	CVE key.
	total_interactions	int	Count of user events (stars/forks) aggregated across that CVE's repositories.
	active_days	int	Distinct days with $\geq$1 interaction.
	first_interaction_time, last_interaction_time	timestamptz	Earliest and latest user interaction times.
	first_repo_creation	timestamptz	The earliest repo_created_at timestamp among mapped GHP repositories (GHP "publication" time).
	days_to_1, 3, 5, 10, 15, 20, 50, 100	int	Days from GHP publication to first time cumulative interactions reach k.

Formally, let $R(c)$ denote the set of GHP repositories linked to CVE c, and let $E(r)$

represent the set of user interactions (stars or forks) on repository r, where each event satisfies the time constraint `interacted_time` $\leq$ 18 March 2025. The total interaction count for CVE c is defined as:

$$\text{total\_interactions}(c) = \sum_{r \in R(c)} |E(r)|$$

The quantity days_to_$k(c)$ is defined as the smallest integer d such that the cumulative number of interactions across all $r \in R(c)$, within the interval $[0, d]$ days from the earliest GHP publication, reaches or exceeds a threshold $k \in \{1, 3, 5, 10, 15, 20, 50, 100\}$. By selecting different values of the activation threshold k, the model can adaptively filter out dormant vulnerabilities that fail to gain sufficient early traction. These offset dates are later used to define the anchor point `day_0` for each CVE during downstream feature engineering and trend clustering. This will be further discussed in Section 4.1.

Limitation (Watch events): the interaction event log in this dataset records only stars and forks. Due to restrictions in the GitHub API, watch/subscribe events cannot be captured in a reliable, timestamped form. As watch events are generally considered less indicative of engagement compared to stars or forks, their omission is unlikely to materially affect the analysis. Accordingly, watch-related columns are set to zero in repository-level summaries and excluded from all per-event analytics. Throughout this thesis, we therefore define user-level interaction strictly as stars or forks.

3.2 Dataset Analysis

Up until 18 March 2025, the dataset covers 285,849 NVD CVEs, of which 7,081 (2.48%) are linked to at least one mapped GHP repository. Out of all CVEs, 3,803 (1.33%) received recorded GitHub user interactions such as stars or forks. Year-stratified metrics for the 3,803 CVEs reveal that the latency from CVE publication to GHP creation has fallen steadily: average days-to-GHP drop from $\tilde{4}91$ (2018) to $\tilde{1}1$ (2025), indicating faster GHP availability and dissemination over time.

The GHPs for the 3803 CVEs are distributed across 14,710 repositories, with publication volume exhibiting clear phases of growth (Figure 3.1, left). This momentum culminated in a peak between 2022 and 2024, during which more than 2,500 repositories were created annually, reaching 3,385 in 2024 alone. In the first months of 2025, before the dataset cutoff, a further 709 repositories had already been published.

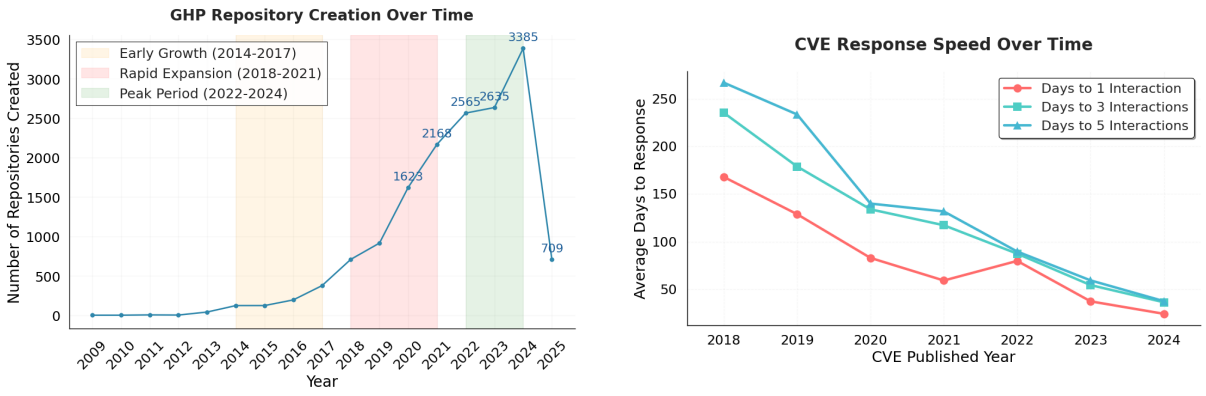


Figure 3.1: Temporal evolution of the GHP ecosystem. Left: annual volume of new GHP repositories. Right: decreasing delay between GHP publication and initial user interactions.

The GHP-CVE relationship is predominantly one-to-one. Of 14,710 GHP repositories in the dataset, 14,240 (96.80%) target a single CVE, while 470 (3.20%) reference multiple CVEs. As mentioned in Section 3.1.1, this research exclude GHP repositories referencing multiple CVEs, enforcing a one-to-one mapping for analytical clarity. From the CVE's perspective, most vulnerabilities are associated with few GHPs: 74.18% have exactly one GHP, 15.80% have 2–3, and only 0.38% have 50+ GHPs (max 478), indicating an extreme long-tail of ecosystem attention.

Total interaction threshold and response speed analysis: within the CVE subset that received at least one user interaction (n=3,803), 67.34% of CVEs reach a total of 3 interactions, 55.35% reach 5, and 42.15% reach 10. In contrast, only 19.88% and 12.96% of CVEs cross the higher thresholds of 50 and 100 interactions, respectively. Figure 3.1 (right) shows that the time required to reach early attention thresholds has accelerated notably in recent years. In 2018, the average delays from PoC repository creation to the first, third, and fifth interaction were 168, 236, and 267 days. By 2024, these delays had dropped sharply to 24, 36, and 37 days. This shift reflects both faster discovery and increased social engagement within the ecosystem.

3.2.1 90-Day Window Analysis

In a fixed 90-day observation window, anchored at the day each CVE reaches its first interaction (`days_to_1`), GHP attention is strongly front-loaded yet highly unequal. Let I_3 and I_{90} denote the cumulative number of user interactions (stars or forks) a CVE receives within the first 3 and 90 days, respectively. We define the **early-share ratio** as $s = I_3/I_{90}$, which captures the concentration of attention in the early phase. All 90-day totals are right-censored at 18 March 2025 to preserve temporal integrity.

At the CVE level, the median early-share is $s = 0.67$, with 70.8% of CVEs receiving at least half of their 90-day interactions within the first three days, and 43.3% receiving at least 80%. At the dataset level, however, only 31% of all recorded interactions occur in the first three days, indicating that a small subset of CVEs continues to attract substantial attention well beyond the initial burst.

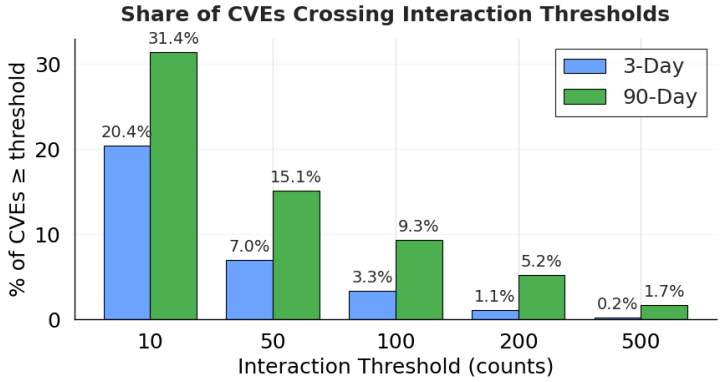


Figure 3.2: *Share of CVEs crossing interaction thresholds (3-day vs 90-day): Percent of CVEs with totals ≥{10, 50, 100, 200, 500}. Same cohort used for both windows.*

Consistent with this skew, Figure 3.2 shows that 3.3% of CVEs surpass 100 interactions within the first three days, compared to 9.3% by day 90. This gap widens at higher thresholds: only 0.2% of CVEs reach 500 interactions within 3 days, while 1.7%

do so within 90 days. This further indicates that while early surges are more common, sustained attention is rarer and concentrated in a small subset of highly engaging CVEs.

In summary, the 90-day GHP interaction window captures a spectrum of behavioural trajectories, from CVEs that attract no meaningful engagement, to those that trigger intense early bursts, and rarer cases that accumulate attention gradually over time. These patterns underscore both the steep inequality and the temporal diversity of GHP user engagement, reinforcing the need for predictive models that account for variation in shape and timing. This motivates the adoption of trend-based clustering and shape-aware modelling, presented in Chapter 5 and Section 6.2, respectively.

3.2.2 Relative Disclosure Timing

Beyond analysing GHP interaction timelines, we also want to understand where GHP activation falls within the broader lifecycle of each CVE. To capture the temporal relationship between PoC emergence and official disclosure, we introduce a categorical variable called activation timing type. For each CVE, we compare the GHP activation date (the day it first received user interaction) with its official publica- tion date in the NVD. Based on this comparison, CVEs are grouped into three mutually exclusive categories:

- Pre-published: activation occurred before NVD disclosure (suggesting insider access, leaks, or early PoC emergence);
- Same-day: activation occurred on the same day as NVD publication;
- Post-published: activation occurred after disclosure (representing common release trajectory).

Because some repositories may have been created for unrelated purposes and only later adapted to host PoCs, we exclude any repository whose creation date precedes the associated CVE's publication by more than 270 days. This conservative threshold helps reduce false early signals. In total, 40 such repositories were excluded from the analysis.

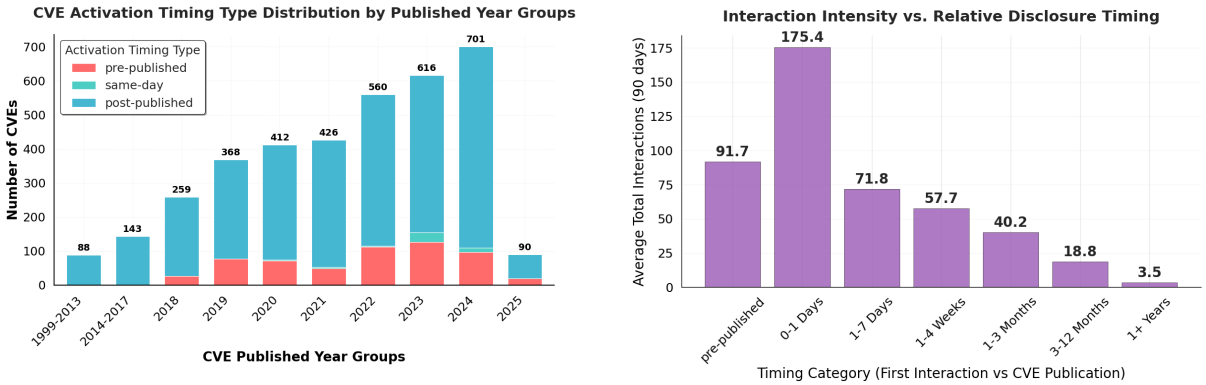


Figure 3.3: **Left:** Distribution of activation timing types by CVE publication year. **Right:** Average total interactions over 90 days since GHP activation, stratified by the delay between CVE publication and first user interaction.

.

Figure 3.3 (left) shows the distribution of these activation types over time, grouped by CVE publication year. From 2018 onward, the majority of CVEs activated after NVD

disclosure, but a persistent subset (between 15% and 25% in recent years) activated before or on the disclosure day. This observation is operationally significant: pre-published and same-day cases may indicate limited patching windows or attacker pre-positioning.

Figure 3.3 (right) shows how interaction intensity varies with the disclosure timing. GHPs released within a day of disclosure received by far the most attention (approx. 175 interactions on average), nearly 50× higher than those emerging a year or more later (approx. 3). Pre-disclosure GHPs also drew substantially higher engagement (approx. 92 interactions) than GHPs appearing weeks or months after disclosure, underscoring that earlier visibility strongly amplifies downstream traction. Together, these results confirm that disclosure tempo is not only a structural marker of ecosystem shifts but also a determinant of GHP relevance.

In later sections (see Section 4.3), we incorporate relative disclosure timing as a contextual feature, encoding both the CVE's publish year and its relative position on the GitHub timeline. These features capture disclosure tempo, ecosystem shifts, and potential attacker anticipation, extending beyond GHP interaction volumes.

Chapter 4

Global Prediction

4.1 Global Prediction Methodology

This section introduces our framework for forecasting CVE popularity using early GHP interaction patterns. The methodology is structured around five core components: (1) input–output window design; (2) modelling approach; (3) feature engineering; (4) multi-metric evaluation protocol; and (5) post-hoc model interpretation using SHAP (SHapley Additive exPlanations).

4.1.1 Activation Threshold and Input–Output Window Design

To filter out completely dormant CVEs that never gain meaningful engagement, we introduce an *activation threshold*: a CVE is considered *activated* once its cumulative GHP interaction count reaches a threshold value k (e.g., 3, 5, 10). As described in Section 3.1.1, for all CVEs that meet this threshold, the corresponding `day_to_k` timestamp is designated as the anchor point (`day_0`). This timestamp serves as the temporal starting point for downstream tasks, including feature engineering, time series construction, and early-trend clustering. Model performance is tested on various activation thresholds.

We formulate the forecasting task as a regression problem over temporally aligned windows. For each CVE, the input window captures early GHP interaction activity, anchored at the point of initial repository activation. The output window corresponds to a fixed forecast horizon, specifically 30, 60, or 90 days, representing the cumulative future interaction volume the model aims to predict.

To systematically evaluate early signal quality and long-range forecastability, we consider a grid of input–output (I/O) window combinations. Specifically, we use input lengths of 3, 5, 7, 14 days and output lengths of 30, 60, 90 days. This design reflects the real-world tempo of security operations. A 3-day input window represents an early forecast opportunity, while the 5- and 7-day inputs align with weekly patch review cycles, making them particularly relevant for operational integration. We also include a 14-day input window as a safety-net reference, enabling us to explore how predictive performance improves when more early signals are available, even at the cost of reduced lead time.

This design ensures that our evaluation framework not only provides fine-grained technical comparison across window lengths, but also captures realistic trade-offs between forecast window length and prediction accuracy.

4.1.2 Modelling Approach

Prediction objective Our goal is to estimate future GHP popularity, as expressed by the repository interaction count, based on early interaction signals. For each retained input–output (I/O) pair, we predict the total GHP interaction count at day 30, 60, or 90. The learning target is defined as:

$$y_{t+h} = \log(1 + \mathrm{GHP}(t + h)),$$

where t denotes the activation day and $h \in \{30, 60, 90\}$ corresponds to the forecast horizon. This log-transformed target stabilises variance and supports comparison across CVEs with vastly different popularity scales.

Time-aware train–test split To preserve temporal causality, we apply a chronological split between training and test data:

Table 4.1: Chronological train–test split for global CVE forecasting.

Phase	**Date range**	**# CVEs**
Train	2018-01-01 – 2024-03-17	2,833
Test	2024-03-18 – 2025-03-18	823 (30-day), 767 (60-day), 705 (90-day)

CVEs are included in the test set only if their lifespan, starting from the activation date, fully covers the selected output window. For example, a CVE activated on 2025-01-15 is eligible for the 30-day task but excluded from the 90-day task. This filtering step ensures evaluation consistency across forecast horizons.

Modelling choice We adopt XGBoost(gradient-boosted decision trees, [Chen and Guestrin, 2016]) as our primary modelling method due to the following reasons:

1. **Effective with modest sample sizes.** XGBoost is well suited for our dataset scale (approx. 3,400 CVEs), its tree-based models maintain strong performance even with small-to-medium tabular data. This property is especially valuable in our cluster-wise analysis (Section 6–7), where some CVE clusters contain less than 100 examples. In these settings, XGBoost provides stable learning dynamics, avoids overfitting through early stopping and regularisation.
2. **Robustness to skewed targets.** GHP popularity is highly skewed, with interaction counts spanning several orders of magnitude. While we apply a `log1p` transformation to reduce this extreme variance, the resulting target distribution remains asymmetric,dominated by low-activity CVEs with a long right tail. XGBoost can this residual skew robustly: its tree-based loss functions remain stable and performant even under such conditions, in contrast to linear models which often misrepresent high-variance targets and are vulnerable to outlier distortion.
3. **Built-in feature selection.** When giving a rich set of features, there will naturally be correlated and overlapping signals. XGBoost's greedy splitting and regularisation (e.g., via η, λ, and γ) naturally filter out spurious or low-utility features. This reduces overfitting risk and enables fast, low-maintenance iteration across multiple feature sets.

4. **Scalability and interpretability.** XGBoost offers efficient training, making it suitable for repeated runs across multiple input–output configurations and ablation scenarios. Moreover, its compatibility with SHAP (SHapley Additive exPlanations, [Lundberg and Lee, 2017]) allows us to decompose predictions into per-feature contributions, enabling detailed interpretability at both global and CVE-specific levels.

This combination of robustness, automation, and transparency makes XGBoost a suitable choice for our forecasting framework, particularly when paired with our temporally aligned feature engineering and risk-aware evaluation protocol. To provide a comparison benchmark, we include Linear Regression as a baseline estimator. Although limited in its ability to capture non-linear growth patterns or delayed bursts, Linear Regression provides a transparent and interpretable point of reference.

4.1.3 Feature Space

To predict the future popularity of each CVE, we constructed features directly from early GitHub Proof-of-Concept (GHP) interaction timelines. These features fall into three main groups: raw count features, which capture the absolute volume of attention; shape-based features, which describe how that attention evolved during the input window; and contextual features, which incorporate high-level timing information such as disclosure year and delay between CVE publication and PoC activity. Together, these features provide the model with a comprehensive representation of both what happened, how it unfolded, and when it occurred in the vulnerability lifecycle.

Raw Count Features. This feature group captures the early volume of user engagement and accompanying repository-level metadata. The calculation follows a similar methodology to the aggregate attributes introduced in Section3.1.1 and listed in Table 3.1 (e.g., `total_interactions`, `active_days`), but is restricted to a fixed early-engagement window (e.g., the first 7 days after activation).

Formally, let $R(c)$ denote the set of GHP repositories linked to CVE c, and let $E(r)$ represent the set of user interaction events (stars or forks) associated with repository r. For a fixed offset t, measured in days since GHP activation, the **cumulative interaction count** up to day t (where t = 1, 3, 5, 7, 30, 60, 90) is defined as:

$$\text{total\_interactions}_t(c) = \sum_{r \in R(c)} |\{e \in E(r) : 0 \leq \texttt{days\_since\_GHP\_activation}(e) \leq t\}|$$

This expression yields feature values such as `total_interactions_3`, `total_interactions_5`, etc., which capture the cumulative number of user interactions received by CVE c within the first t days after PoC publication.

We also compute:

- **Active days** within the input window, defined as the number of distinct days with at least one recorded interaction (e.g., `active_days_5`, `active_days_7`).
- **Repository metadata aggregates**, including total number of linked GHP repositories to each CVE, and the cumulative fork and star counts within fixed time slices (e.g., `repo_count_day_3`, `stars_day_3`, `forks_day_5`).

Together, these features form the foundational signal for capturing early interest volume and initial popularity growth.

Shape-based Features. Based on raw count features, we engineered shape features that describe how the GHP activity evolved over time. These include:

Table 4.2: Overview of shape-based features derived from early GitHub PoC interaction timelines.

Feature Type	Feature Name	Computation
Log-scaled Volume	`log_count_{k}`	$\log(1 + \texttt{total\_interactions\_\{k\}})$
Interaction Gain	`delta_{i}_{j}`	$\texttt{total\_interactions\_\{j\}} - \texttt{total\_interactions\_\{i\}}$
Slope (Growth Rate)	`slope_{i}_{j}`	$\frac{\texttt{total\_interactions\_\{j\}} - \texttt{total\_interactions\_\{i\}}}{j-i}$
Burstiness Ratio	`burstiness_{k}`	$\frac{\texttt{active\_days\_\{k\}}}{k}$
Early Share Ratio	`early_share_{i}_{j}`	$\frac{\texttt{total\_interactions\_\{i\}}}{\texttt{total\_interactions\_\{j\}}}$

- **Log-scaled volumes** (e.g., `log_count_3`): stabilise variance and reduce scale distortion;
- **Deltas and slopes** across time spans (e.g., `delta_3_5`, `slope_3_7`): capture acceleration or decay in attention;
- **Burstiness ratios** (e.g., `burstiness_7`): quantify how unevenly activity is distributed within the input window;
- **Early share ratios** (e.g., `early_share_3_7`): express what proportion of the total attention occurred in the first few days within the input window.

Table 4.2 shows how shape-based features are computed, based on the raw count features. Shape-based features allow the model to distinguish among different types of early engagement patterns, such as slow-burn CVEs that accumulate attention gradually over time, versus early-burst CVEs that peak rapidly at the start of the input window.

For each input–output (I/O) configuration, the feature set is pruned to align with the length of the input window. Features that require more temporal context than the input window provides (e.g., `delta_5_14` for a 3-day input) will be excluded. This enforces temporal consistency and avoids information leakage from future observations.

Contextual Features. In addition to GHP popularity dynamics captured by raw and shape-based features, this next group of features provide a broader temporal context for each CVE in its vulnerability lifecycle.

Table 4.3: Contextual feature definitions.

Variable	Definition
`cve_year`	NVD publish calendar year (2018–2025)
`publish_delay`	Days between CVE publication and GHP activation
`delay_type`	Categorical: *pre-published*, *same-day*, *post-published*

To prevent future data leakage, we masked CVEs with GHP activation date preceding their NVD publish date: during preprocessing, `publish_delay` is filled with zero; whereas `cve_year` and `delay_type` are one-hot encoded as category `"Unknown"`, which explicitly represent masked cases.

4.1.4 Evaluation Protocol

We assess model performance using a combination of standard regression metrics, log-transformed variants, and residual skewness. This protocol is designed not only to measure predictive accuracy, but also to reflect the real-world interpretability and operational relevance of each score,particularly in the context of early risk triage and vulnerability prioritisation.

For each input–output (I/O) configuration, we evaluate models using both absolute-scale and log-transformed variants of common regression metrics:

- **R^2 score** (coefficient of determination), computed on:
  - the raw target (untransformed counts) to measure real-world prediction usefulness, and
  - the log target (`log1p(total_interaction)`), to evaluate the model's ability to capture order-of-magnitude patterns across CVEs, particularly important in ranking and prioritisation contexts;
- **Mean Absolute Error (MAE)**, for both log-scaled target and in absolute value, to quantify the average of prediction errors;
- **Root Mean Squared Error (RMSE)**, which penalises large errors more heavily than MAE, making it useful for stress-testing predictions on high-variance or outlier CVEs;
- **Residual skewness (log-scale)** of prediction errors, used to diagnose systematic bias.

Table 4.4: Error metric formulas used in model evaluation. Here, y_i denotes the true value and $\hat{y}_i$ the predicted value for sample i.

Metric	Formula
MAE (Mean Absolute Error)	$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\lvert\hat{y}_i - y_i\rvert$
R^2 (Coefficient of Determination)	$R^2 = 1 - \frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$
Log-MAE / Log-R^2	Same formulas applied to $\log(1+y_i)$ and $\log(1+\hat{y}_i)$
Residual Skewness	$\text{Skew} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{r_i - \bar{r}}{s}\right)^3$, where $r_i = \hat{y}_i - y_i$
RMSE (Root Mean Squared Error)	$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}$

All metrics are computed on the fixed test set (from 2024-03-18 to 2025-03-18) using only CVEs whose lifespan fully covers the output window. This ensures that all predictions are evaluated against a fully observed ground truth.

4.1.5 Post-hoc Model Interpretation (SHAP Analysis)

To interpret the internal logic of the fitted XGBoost models, we apply SHAP (SHapley Additive exPlanations, [Lundberg and Lee, 2017]). SHAP assigns a local contribution score to each feature for every CVE, by decomposing the model's prediction into a sum of feature attributions plus a base value (the expected output). Grounded in cooperative game theory, this method quantifies how much each feature increases or decreases the forecasted popularity for a given CVE, relative to the baseline. For example, features

such as early fork volume or the time between NVD publish date and GHP activation date may have strong positive or negative influence depending on their specific values.

SHAP also supports global analysis in addition to per-sample explanation. This enables three key diagnostic capabilities: (1) identifying globally influential features; (2) comparing feature impact distributions across input windows, feature sets, and PoC trend shape clusters; and (3) diagnosing systematic prediction errors by analysing the SHAP profiles of high-residual CVEs. SHAP thus serves not only as a transparency mechanism, but also as a practical diagnostic tool, offering a principled way to interpret and evaluate model behaviour.

4.2 Global Prediction Results

4.2.1 Baseline Model vs. XGBoost Regression

To provide a benchmark for model performance, we conducted a comparative experiment between Linear Regression and XGBoost Regression, using a minimal feature set composed of the total interaction count and number of active days within the early observation window. This experiment focuses on two input configurations: 5-day and 7-day. The task is to predict the cumulative number GHP interactions at day 30. Each model was trained on a dataset of 2,833 CVEs published before 2024-03-18 with full input–output coverage, and evaluated on a held-out test set of 823 CVEs (post-2024-03-18) that satisfy the 30-day output horizon constraint. All models were trained in log-transformed space, using `log1p()` on the target variable. This transformation improves learning stability and emphasises order-of-magnitude differences, ensuring fairer comparison between models, particularly for Linear Regression, which is otherwise highly sensitive to outliers. XGBoost models were trained with default regularisation; Linear models were fitted via ordinary least squares.

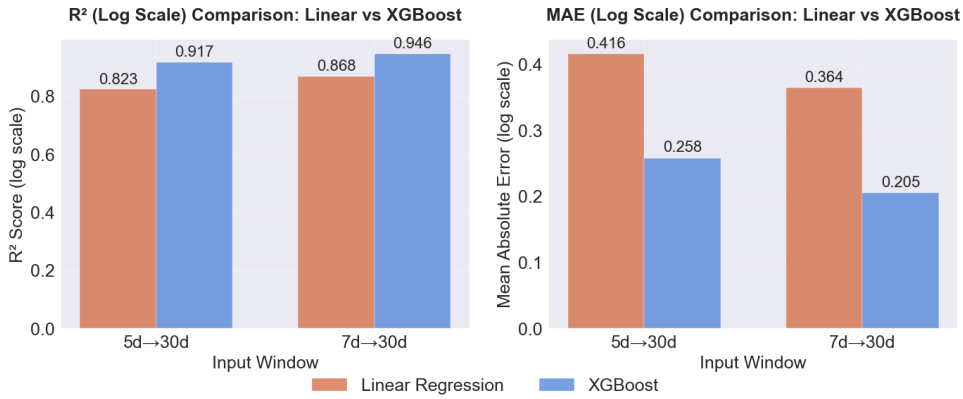


Figure 4.1: Performance comparison between Linear Regression and XGBoost using only raw count features (interaction count and active days) for 5-day and 7-day input windows. Left: R^2 score (log scale); Right: MAE (log scale). Bars show model fit and forecast accuracy over the 30-day output horizon.

As shown in Table 4.5 and Figure 4.1, XGBoost consistently outperforms Linear Regression across all metrics. On the log-scale R^2 axis (left subplot), XGBoost improves performance by +0.094 for 5-day inputs and +0.078 for 7-day inputs, indicating a stronger fit to CVE popularity growth. On the log-scale MAE axis (right subplot), error reductions

Table 4.5: Baseline Model Performance Comparison (Updated)

Experiment	Model	R^2 (Log)	MAE (Raw)	MAE (Log)
5d→30d	Linear	0.823	18.4	0.416
5d→30d	XGBoost	0.917	11.8	0.258
7d→30d	Linear	0.868	23.0	0.364
7d→30d	XGBoost	0.946	6.7	0.205

are substantial: MAE decreases by 38% (from 0.416 to 0.258) for 5-day inputs and 44% (from 0.364 to 0.205) for 7-day inputs. These results demonstrate the predictive benefit of non-linear decision boundaries, even when the input feature set is minimal.

4.2.2 Activation Threshold, Input–Output Window and Raw Count vs. Shape-based Feature

To assess the global prediction model on a broad scale, we first evaluate the performance across 96 experiment groups defined by activation thresholds (1, 3, 5, 10 interaction counts), input window lengths (3, 5, 7, 14 days), output windows (30, 60, 90 days), and two feature sets (raw count only; raw count + shape-based features). Rather than enumerating all results, we highlight representative cases that capture the main findings. The discussion is structured along three dimensions: (i) feature set comparison, (ii) input–output window trade-offs, and (iii) activation threshold effects. This analysis leads to the identification of a focused subset of configurations that serve as the basis for subsequent model tuning.

Feature Set Comparison

Section 4.2.1 shows that even with raw count features alone, the XGBoost model achieves solid performance on the log-scale metrics (R^2 and MAE). In contrast, performance on the absolute scale is markedly weaker across the full set of 96 experiments: R^2(Abs) values are frequently low or negative for short input windows (3, 5, 7 days). The only exception is the 14-day input, where R^2(Abs) remains above 0.7, indicating that longer observation periods can partially offset the limitations of raw count features.

Adding shape-based features alongside raw counts yields prominent gains on absolute-scale R^2, while also delivering consistent improvements across other metrics, including log-scale R^2, MAE, and RMSE. Representative cases with activation TH5 and input windows of 5 or 7 days illustrate this effect.

Table 4.6 shows how the raw-only feature set struggles on the absolute scale: R^2(Abs) values are often negative, and MAE ranges between 35–51 for 5-day inputs and 27–45 for 7-day inputs. When shape-based features are added, R^2(Abs) improves dramatically, reaching values as high as 0.820 for the 7d→90d configuration, while MAE decreases by 9–20 points across horizons. RMSE also drops substantially, with reductions of 55–63% relative to the raw-only baseline, reflecting a markedly lower influence from large errors.

Table 4.6: Performance comparison between raw-only and raw+shape feature sets (TH5).

Input	Output	Feature Set	R^2(Abs)	MAE	RMSE	ΔR^2(Abs)	ΔMAE	ΔRMSE
5d	30d	Raw	–0.003	35.11	257.34			
		Both	0.783	22.17	119.67	+0.786	–12.94	–137.67
7d	30d	Raw	0.425	26.82	194.92			
		Both	0.763	18.18	124.99	+0.338	–8.64	–69.93
5d	60d	Raw	–0.336	43.08	317.78			
		Both	0.650	30.11	162.57	+0.986	–12.97	–155.21
7d	60d	Raw	0.126	34.91	257.04			
		Both	0.777	23.38	129.94	+0.651	–11.53	–127.10
5d	90d	Raw	–0.623	50.99	376.54			
		Both	0.595	35.32	187.98	+1.218	–15.67	–188.56
7d	90d	Raw	–0.294	44.76	336.17			
		Both	0.820	24.36	125.34	+1.114	–20.40	–210.83

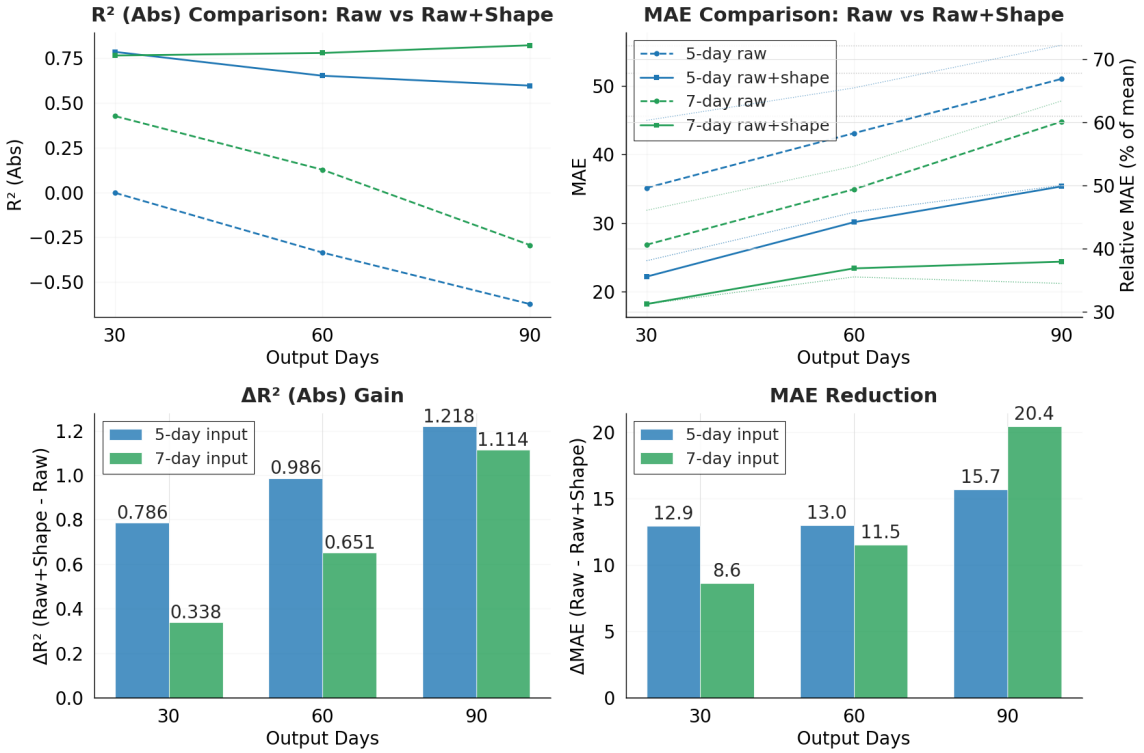


Figure 4.2: Impact of adding shape-based features to raw-only models for 5-day and 7-day inputs across all output horizons. **Top left**: R^2 (Abs) comparison. **Top right**: MAE comparison (dashed lines show raw-only models, solid lines show raw+shape). Secondary Y-axis shows MAE as a percentage of the target mean. **Bottom left**: ΔR^2 (Abs) from shape feature inclusion. **Bottom right**: Corresponding reduction in MAE.

Figure 4.2 illustrates these effects. The top row shows how raw-only models fail to generalise on the absolute scale (dashed lines), while the shape-augmented models (solid lines) consistently stabilise performance. The lower row highlights the deltas explicitly: gains of +0.7 to +1.2 on R^2(Abs) on 5-day input windows, and MAE reductions of up to 20 points, with improvements becoming more pronounced as the output horizon extends.

The same improvement pattern can be observed across I/O windows and for activation thresholds. This comparison results indicate that encoding temporal shape patterns in early attention curves provides complementary information beyond raw engagement volume. The benefit is most visible on the absolute scale, where raw-only models exhibit unstable or even negative fit, whereas shape-augmented models achieve consistently improved R^2 and reduced error. Incorporating early engagement trajectory features into GHP popularity forecasting therefore provides a persistent and meaningful performance advantage. Based on these findings, we retain the **raw count + shape-based feature set** as the standard configuration for the following analyses.

Input-Output Window Trade-offs

Continuing with activation TH5 configurations, Table 4.7 reports detailed test-set results across all I/O window combinations. Reported metrics include log- and absolute-scale R^2, log- and absolute MAE, RMSE, and log-scale residual skewness. Figure 4.3 illustrates the performance trade-off across I/O windows more directly, comparing R^2(Abs) and MAE as input window lengthens.

Table 4.7: Performance across I/O window configurations (activation TH5).

Input	Output	R^2(Log)	R^2(Abs)	Log MAE	MAE	RMSE	Residual Skew
3d	**30d**	0.853	–0.055	0.393	36.78	264.00	–2.19
5d	**30d**	0.910	0.783	0.320	22.17	119.67	–1.63
7d	**30d**	0.929	0.763	0.269	18.18	124.99	**–0.37**
14d	**30d**	0.972	0.784	0.163	13.57	119.31	–2.49
3d	**60d**	0.833	–0.064	0.452	43.35	283.57	–1.77
5d	**60d**	0.897	0.650	0.368	30.11	162.57	–1.66
7d	**60d**	0.918	0.777	0.328	23.38	129.94	**–0.44**
14d	**60d**	0.957	0.832	0.239	15.82	112.67	–1.04
3d	**90d**	0.812	–0.130	0.486	49.30	314.17	–1.40
5d	**90d**	0.875	0.595	0.415	35.32	187.98	–1.32
7d	**90d**	0.899	0.820	0.370	24.36	125.34	**–0.22**
14d	**90d**	0.944	0.843	0.280	18.47	117.06	–0.89

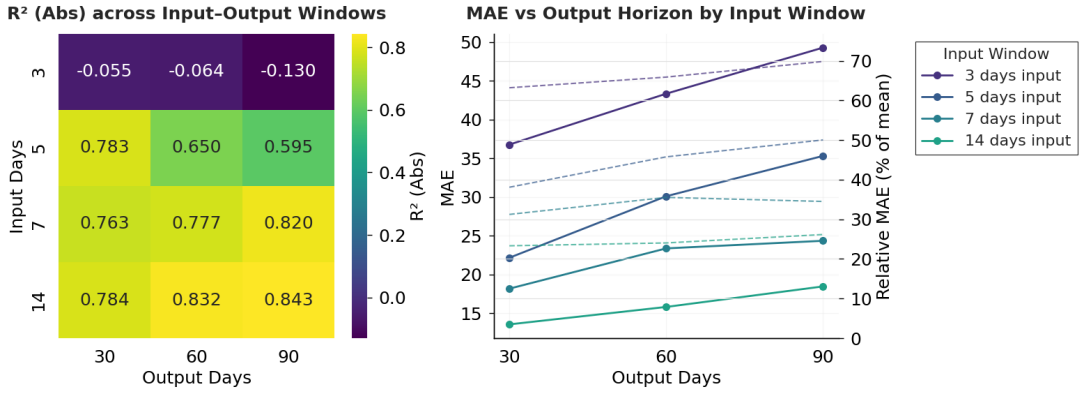


Figure 4.3: Evaluation across I/O windows using raw+shape features. **Left:** R^2 (Abs) heatmap; **Right:** MAE vs. forecast horizon. Solid lines show absolute error; dashed lines show MAE as a % of the target mean.

We observed from the I/O window performance that:

(1) 3-day input windows are insufficiently informative. Across all output horizons, models trained on 3-day input windows consistently underperform relative to longer inputs. For example, R^2(Abs) remains negative across the board (–0.055, –0.064, –0.130), with MAE as high as 36.8–49.3. These results indicate that, for the current model, GHP activity within the first 72 hours is generally too sparse or weakly differentiated to support robust forecasting on a global scale.

(2) 5-day and 7-day inputs offer the best trade-off. In contrast, 5-day and 7-day input windows achieve the best balance between predictive accuracy and usable lead time. At the short horizon, the 5→30d configuration attains R^2(Abs) = 0.783 with MAE = 22.2, a relatively strong performance while still preserving 83% of the forecast horizon. The 7→30d configuration further improves accuracy (R^2(Abs) = 0.763, MAE = 18.2) while maintaining stable error behaviour. At longer forecast horizons, 7→90d configuration reaches R^2(Abs) = 0.820 with MAE = 24.4, demonstrating strong predictive capacity. Notably, 7-day inputs also yield the most balanced residual skewness (–0.22 at 90 days), indicating a more symmetric error distribution and reduced systematic bias.

Taken together, 5- and 7-day input windows emerge as strong candidates for model integration, with 5-day windows offering a favourable accuracy-to-lead-time ratio, and 7-day windows providing relatively higher predictive accuracy and error stability.

(3) Fourteen-day inputs maximise accuracy but sacrifice lead time. Models with 14-day input windows achieve the highest absolute accuracy (e.g., R^2(Abs) = 0.843, MAE = 18.5 for 14→90). However, the gains over 7-day inputs are modest (e.g., only +0.023 R^2(Abs) at 90d). Figure 4.3 further shows that relative MAE (scaled to the target mean) is not substantially improved despite the longer observation period. Moreover, with a 14-day input, lead-time utility is severely reduced; in the 14→30d case, predictions are generated halfway through the forecast horizon. Overall, these factors indicate that 14-day input models are less practical for proactive vulnerability triage.

Based on these findings, we retain the **5- and 7-day input window** configurations for subsequent analyses. These settings (i) deliver consistent performance across both log- and absolute-scale metrics, (ii) strike a practical balance between forecast range and lead time, and (iii) exhibit relatively balanced residual skewness. Within this set, the 5→60, 5→90, and 7→90 configurations combines extended forecast horizons with short observation periods, and are therefore selected as the focused subset for evaluation and diagnostic analysis in later sections.

Activation Threshold Effects

As the activation threshold increases, more CVEs are filtered out, which could theoretically improve model performance by reducing noise from low-activity cases. To assess this effect, we compare model performance across thresholds of 1, 3, 5, and 10 interactions, with the aim of identifying the setting that best balances dataset coverage and predictive accuracy.

Table 4.8: Model performance across activation thresholds and output horizons (5-day input, raw+shape features).

Threshold	Output	Samples	R^2 (Log)	R^2 (Abs)	Log MAE	MAE	Residual Skew
1	30d	823	0.902	0.451	0.282	16.55	–2.17
3	30d	528	**0.914**	0.481	0.299	22.59	–1.92
5	30d	414	0.910	**0.783**	0.320	22.17	–1.63
10	30d	306	0.906	0.314	0.352	38.17	–1.81
1	60d	767	0.865	0.421	0.345	19.62	–2.80
3	60d	485	0.894	0.439	0.362	27.14	–1.71
5	60d	386	**0.897**	**0.650**	0.368	30.11	–1.66
10	60d	286	0.873	0.292	0.408	43.75	–1.34
1	90d	705	0.843	0.241	0.388	23.10	–2.51
3	90d	446	**0.876**	0.365	0.398	31.35	–1.49
5	90d	354	0.875	**0.595**	0.415	35.32	–1.32
10	90d	261	0.833	–0.035	0.468	55.37	–1.00

Within each activation threshold, the results exhibit the same feature-set and I/O window patterns observed in earlier sections. Comparing across them, Table 4.8 reports performance for 5-day inputs across the four thresholds.

The pattern is consistent across all horizons: TH5 provides the best overall performance, with the highest R^2(Abs) values (e.g., 0.783 at 30d, 0.595 at 90d) and relatively low error. TH3 ranks second, producing slightly higher errors but still competitive results. Threshold = 1 suffers from the inclusion of noisy, low-activity CVEs, yielding weaker absolute fit despite solid log-scale scores. In contrast, threshold = 10 shows clear degradation, most evident at the 90-day horizon where R^2(Abs) drops below zero (–0.035).

In terms of dataset coverage, for the 514 CVEs in the test set, TH5 retains 414 samples at 30 days (approx. 81%), 386 at 60 days (approx. 75%), and 354 at 90 days (approx. 69%), which is sufficient for stable evaluation. By contrast, threshold = 10 reduces coverage to 306 samples at 30 days (approx. 60%), with further drops at longer horizons.

From a practical perspective, thresholds 3 and 5 retain 67.3% and 55.4% of all CVEs with observed user interaction in the full dataset ($n = 3,803$), respectively, compared to that of 42.2% for threshold 10. Overly strict thresholds risk excluding a substantial portion of GHP interaction trajectories and reducing the diversity of patterns available for modelling.

Overall, TH5 offers the best balance, filtering out noise while retaining sufficient coverage for stable prediction. TH3 remains a viable alternative, albeit with slightly more noise. For subsequent model tuning, we therefore focus on datasets with **activation threshold 3 and 5**.

4.2.3 Hyperparameter Tuning

Based on the results of the previous section, our final model configuration adopts the **raw count + shape-based feature set**, with **activation thresholds of 3 and 5** and **input windows of 5 or 7 days**. To evaluate the robustness of this modelling setup and assess the potential for further performance gains, we conducted a series of

follow-up experiments incorporating lightweight hyperparameter tuning on these selected configurations.

The hyperparameter tuning focused on three strategic dimensions: (i) depth–learning rate trade-offs, (ii) regularisation, and (iii) subsampling sensitivity. To ensure time-consistent evaluation, the training data were partitioned chronologically: models were trained on CVEs disclosed between January 2018 and September 2023, and validated on those disclosed between October 2023 and March 2024. This nine-month holdout was selected to reflect recent vulnerability trends while maintaining a sufficient validation sample ($\approx 10\%$).

A curated sweep of **21 parameter configurations** was conducted, complemented with a few extreme baselines for robustness checks. The combinations were designed to systematically cover representative regions of the hyperparameter space:

- **Group A (Depth–Learning Rate Trade-off):** Depth $\in \{3, 6, 9\}$, learning rate $\in \{0.1, 0.05, 0.01\}$, estimators = 300–1000.
- **Group B (Regularisation Sensitivity):** Depth=6, learning rate=0.1, estimators=300, with $(\alpha, \lambda) \in \{(0, 1), (0.1, 1.5), (0.5, 2)\}$.
- **Group C (Subsampling Effects):** Depth=6, learning rate=0.1, estimators=300, with subsample, colsample_bytree $\in \{(1.0, 1.0), (0.8, 0.8), (0.7, 0.7)\}$.
- **Group D (Extreme Configurations):** Includes deliberately shallow or heavily regularised models to probe failure modes and serve as baselines for robustness.
- **Group E (Gap Fillers):** Adds edge cases such as deep but slow learners or conservative "patience" configurations (e.g., 1500 estimators with a low learning rate) to ensure coverage of intermediate possibilities not spanned by Groups A–C.

The complete set of parameter configurations is reported in Appendix Table 1. This setup makes the sweep appropriate as a lightweight robustness check, sufficient to validate that performance gains arise from the chosen feature and data configurations rather than arbitrary parameter choices. At the same time, it also probes the headroom for further improvement under more extensive tuning.

Tuning on TH5 dataset. We first conducted hyperparameter tuning on the TH5 dataset, since it provided the strongest baseline performance in earlier sections.

Across the 21 candidate configurations, results revealed a consistent trade-off between *predictive accuracy* and *bias symmetry*. Shallow models with conservative learning schedules (e.g., `md2_lr0.03_ne800`) achieved high log-scale performance (R^2(Log) = 0.908), but suffered from relatively strong negative skew (–1.81), systematically underpredicting high-attention CVEs. In contrast, deeper and faster learners (e.g., `md6_lr0.1_ne1000`) produced more balanced residuals (skew = –0.26) but only moderate absolute performance (R^2(Abs) = 0.883). Regularisation and subsampling delivered modest improvements in fit, particularly for medium-depth models with controlled estimator counts. A complete result is provided in Appendix, see Table 2.

From the sweep result, we selected four representative configurations for evaluation on post-2024 test set CVEs: a default baseline, a best log-scale performer, a subsampling variant for noice robustness, and a contrast model with balanced skew. Table 4.9 presents

the performance comparison among these four top candidate configurations for the 5→90 input–output window.

Table 4.9: Hyperparameter tuning results for selected configurations (5→90, TH5, test set).

Model	R^2(Log)	R^2(Abs)	Log MAE	MAE	RMSE	Residual Skew
default_md6_lr0.1_ne1000	0.875	0.595	0.415	35.32	187.98	–1.32
md2_lr0.03_ne800	0.904	0.467	0.360	35.11	215.82	–1.96
md4_lr0.03_ne500_ss0.6	0.904	0.604	0.358	30.33	185.92	–2.15
md6_lr0.01_ne1000	0.889	0.458	0.387	35.86	217.50	–1.76

All models retained robust log-scale accuracy (R^2(Log) ≥ 0.875), confirming the stability of early-attention trend modelling. Absolute accuracy, however, varied more sharply than in the training set, with R^2(Abs) ranging from 0.458 to 0.604 across configurations. This degradation in R^2(Abs) also reflects the ecosystem shift that has occurred since the start of 2024. The strongest absolute performer (`md4_lr0.03_ne500_ss0.6`) reduced MAE by approx. 14% relative to baseline, but incurred the most negative skew (–2.15), reinforcing the trade-off between error reduction and bias symmetry. Given this trade-off, we elected to retain the default model configuration for threshold 5, since this setting offered an over-all balance between prediction accuracy and bias symmetry.

Tuning on TH3 dataset. A similar tuning process was applied to the TH3 dataset using the same parameter sweep. Among all configurations, `md6_lr0.05_ne500` provided the clearest improvement over the default baseline (`md6_lr0.1_ne1000`). Absolute-scale fit increased from $R^2_{\text{Abs}} = 0.365$ to 0.566 (+55%), while MAE decreased from 31.35 to 27.54 (-12.2%). This configuration narrows the gap and even brings TH3 performance to a level comparable with TH5. Although similar to TH5, this configuration comes with the trade-off of a slightly more skewed residual (-1.73 compared to baseline -1.49), it remains within an acceptable range. We therefore adopt `md6_lr0.05_ne500` as the model parameter setting for TH3 in subsequent experiments.

Table 4.10: Performance of tuned TH3 models.

Input→Output	Samples	R^2(Log)	R^2(Abs)	Log MAE	MAE	Residual Skew
5→30	528	0.916	0.576	0.294	21.08	–2.10
7→30	528	0.940	0.843	0.236	12.48	–2.51
5→60	485	0.897	0.592	0.351	24.50	–1.94
7→60	485	0.919	0.826	0.302	15.94	–1.71
5→90	446	0.881	0.566	0.389	27.54	–1.73
7→90	446	0.903	0.822	0.342	17.42	–1.53

Table 4.10 shows the result for the tuned TH3 model. Comparing with the TH5 model, with a 7-day input window, TH3 outperforms TH5 across all horizons on absolute-scale accuracy, for example, $R^2_{\text{Abs}} = 0.843$ vs. 0.763 at 7→30. Log-space metrics also favour TH3, with higher R^2_{Log} and lower Log-MAE. In contrast, TH5 remains advantageous when (i) residual symmetry is prioritised (e.g., 7→90 achieves skew –0.22 with essentially identical R^2_{abs} to TH3: 0.820 vs 0.822), or (ii) the task emphasises short-horizon prediction

with minimal evidence (e.g., 5→30).

This tuning study highlights a fundamental challenge of global regression: the difficulty of simultaneously optimising for both accuracy and fairness. These tensions likely stem from the underlying heterogeneity in GHP growth dynamics, which a single global model struggles to capture. Rather than relying on increasingly fine-grained hyperparameter tuning to compensate, a more principled approach may lie in partitioning the CVE population into behavioural archetypes (Section 5.1), allowing the model to specialise in distinct growth patterns rather than forcing uniform generalisation.

With a competitive performance while retaining 2,561 CVEs compared to 2,105 for TH5 (approx. 22% more coverage), combined with its consistently stronger log-scale fit, TH3 better satisfies the requirements of coverage + trajectory diversity, and is therefore adopted for clustering and cluster-wise prediction experiments in Section 6.1 and Section 6.2.

4.2.4 Contextual-Feature Ablation

To test marginal gains from contextual metadata, we added three high-level variables: *CVE publish year*, *activation delay*, and *timing type* (see Section 4.3) to the feature set. Results differed across thresholds.

Threshold = 5. With a 7-day input window, contextual variables improved short- and medium-horizon accuracy. For example, R^2_{abs} increased from 0.763 to 0.798 at 7→30 (MAE 18.18 → 17.27) and from 0.777 to 0.784 at 7→60 (MAE 23.38 → 23.12). Log-scale fit also improved. However, residual skew worsened (e.g., $-0.37 \rightarrow -1.28$ at 7→30), indicating stronger underprediction of high-attention CVEs. At 7→90, contextual features degraded overall fit (R^2_{abs} 0.820 → 0.775; RMSE 111.8 → 140.2).

By contrast, at 5-day inputs, contextual variables consistently improved residual symmetry, for example, absolute skew dropped from 1.66 to 0.87 (–47.6%) at 5→60 window. But accuracy deteriorated (5→30: R^2_{abs} 0.783 → 0.558; MAE 22.17 → 27.18). Overall, contextual variables are beneficial when sufficient early evidence is available ($\geq$ 7 days) and horizons are short to medium ($\leq$ 60 days), but counterproductive for 5-day inputs or long horizons.

Table 4.11: Impact of adding contextual features (TH5).

Window	ΔR^2_{abs}	ΔMAE	ΔRMSE	ΔSkew
5→30	–0.225	+5.01	+51.28	+0.09
7→30	+0.035	–0.91	–9.48	–0.91
5→60	–0.025	+0.28	+5.88	+0.79
7→60	+0.007	–0.26	–2.17	–0.54
5→90	–0.192	+3.50	+40.34	+0.49
7→90	–0.045	–0.37	+14.89	–0.23

Threshold = 3. For TH3, contextual features consistently reduced absolute accuracy. For instance, R^2_{abs} fell from 0.592 → 0.403 (MAE 24.50 → 26.78) at 5→60, and from 0.843 → 0.790 (MAE 12.48 → 13.71) at 7→30. Improvements in residual skew were minor, and

Table 4.12: Impact of adding contextual features (TH3).

Window	ΔR^2_{abs}	ΔMAE	ΔRMSE	ΔSkew
5→30	–0.098	+1.27	+16.28	–0.11
7→30	–0.053	+1.23	+14.19	+0.18
5→60	–0.189	+2.28	+32.93	–0.04
7→60	–0.042	+0.45	+11.72	+0.03
5→90	–0.189	+2.56	+34.55	+0.20
7→90	–0.009	–0.03	+2.80	+0.20

log-metric gains negligible. Given this inconsistency, contextual variables are excluded from the global TH3 models. Instead, residual asymmetry is addressed via cluster-specific modelling with enhanced regression (see Section 6.2.2).

4.2.5 Global SHAP Analysis

To better understand why our models make specific predictions, we adopt TreeSHAP, a feature attribution method tailored for tree-based models such as XGBoost. This analysis helps identify which input variables drive prediction outcomes across different feature sets, and reveals structural limitations in the global forecasting model.

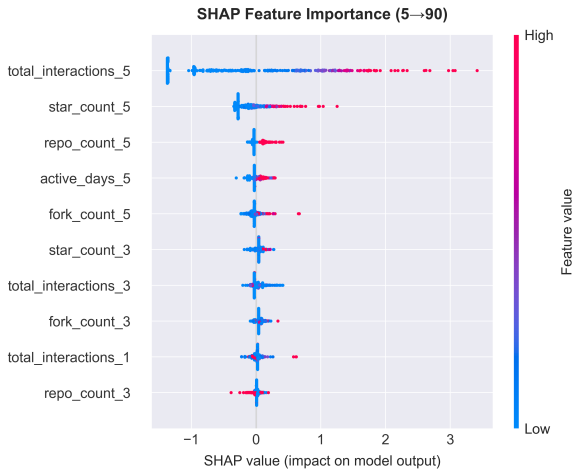
SHAP analysis for TH3 dataset is reported in this section. We focus on the two long-range configurations, 5→90 and 7→90, as these represent the most practically valuable forecasting scenarios. These configurations also exhibit consistent SHAP behaviour across shorter output horizons. As such, they offer a representative and interpretable view into the model's decision-making process.

For each input window, we analyse three progressively enriched feature sets: raw counts only, raw + shape features, and raw + shape + contextual feature set. All SHAP values were computed only on the held-out test set. Figure 4.4 presents the SHAP swarm plot, visualising both the importance and the effect of the features for each configuration. Each point represents a single CVE, with the x-axis showing the SHAP value (effect on prediction). The colour indicates the raw feature value, with red representing high values and blue representing low values.

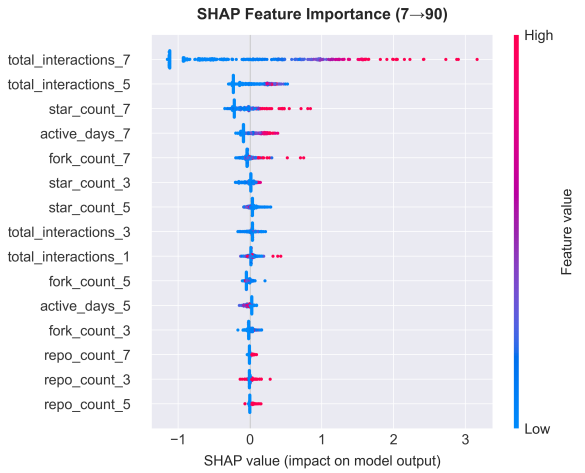
Raw Count Feature Set

In the raw-count only configuration, the model remains dominated by volume-based interaction features. For both the 5-day and 7-day input settings, the total interaction count across the input window is by far the strongest predictor (top-ranked in both plots). This confirms that aggregate user engagement is the single most influential early indicator of GHP trajectory.

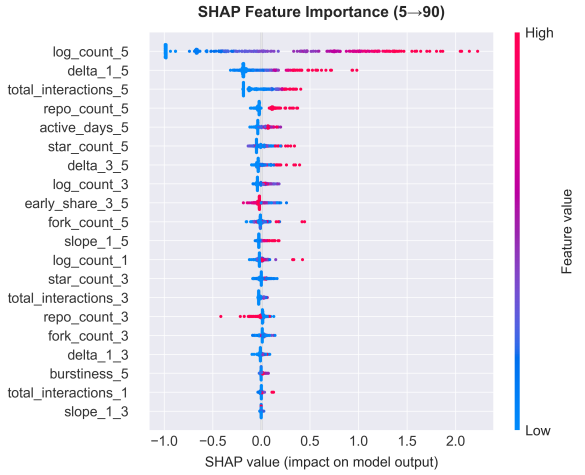
The star count features consistently rank in the top three, underscoring the role of GitHub stars as high-signal endorsements. Meanwhile, active days also appear among the top contributors, reflecting the importance of sustained rather than one-off bursts of attention. Fork counts follow closely, generally reinforcing the signal of substantive user engagement. The SHAP colour patterns confirm this interpretation: high feature values (red) correspond to positive SHAP effects, particularly for total interactions and stars, meaning that more concentrated and sustained early attention pushes predictions of long-range popularity upward.



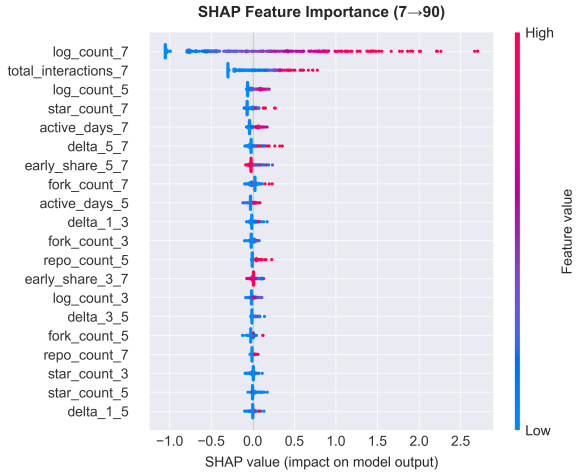
(a) 5-day input, raw counts



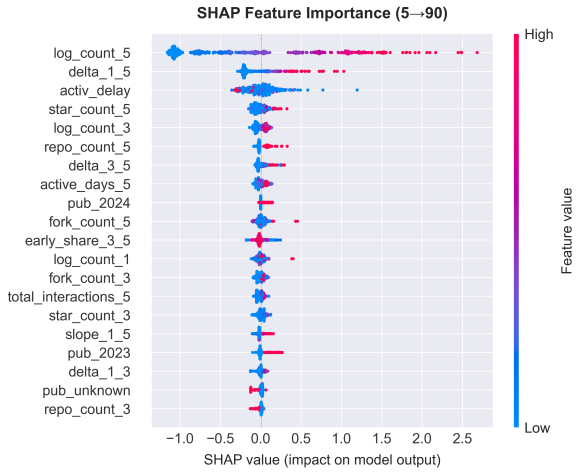
(b) 7-day input, raw counts



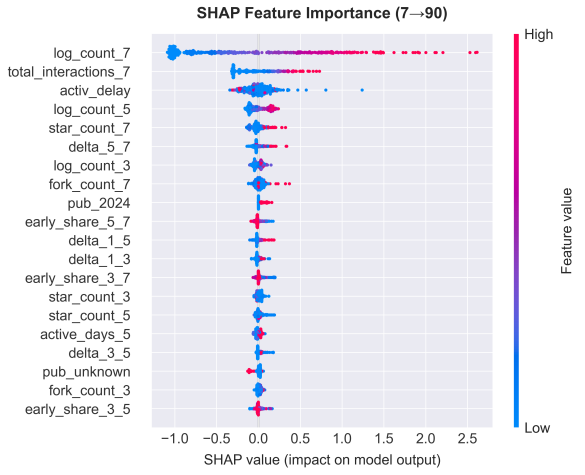
(c) 5-day input, raw + shape



(d) 7-day input, raw + shape



(e) 5-day input, full set



(f) 7-day input, full set

Figure 4.4: Global SHAP feature impact across feature configurations and input windows. (a, b) raw counts only; (c, d) with shape-based features; (e, f) full feature set with contextual variables.

A notable contrast arises in repo count features. In both models, `repo_count_3` and `repo_count_5` tend to exert weaker or even negative influence when values are high, whereas `repo_count_7` begins to reflect a more positive signal. This suggests that very

early repository proliferation (within 3–5 days) might be linked to redundant forks, template reuse, or noisy activity that the model learns to discount. By contrast, a sustained repository number increase over a full 7-day window is interpreted as evidence of genuine community adoption.

Taken together, the raw-count only SHAP analysis highlights the model's preference for strong, authentic engagement trends: high interaction volume, accumulating stars, and distributed activity across days. Interaction intensity and persistence are the key drivers of predictive power in the absence of engineered shape features.

Raw + Shape Feature Set

With the addition of shape-based features, the model's reliance shifts from raw counts to temporal dynamics. Across both the 5-day and 7-day input windows, log-transformed interaction counts (`log_count_5`, `log_count_7`) consistently surpass total interactions in importance, highlighting the stronger explanatory power of scale-normalised growth patterns over raw volumes.

Temporal gradient features such as `delta_1_5` (5-day model) and `delta_5_7` (7-day model) emerge among the top contributors, indicating that early acceleration or deceleration trends carry predictive weight for long-term attention. Similarly, proportional indicators like `early_share_3_5` and `early_share_5_7` consistently show negative SHAP values at high magnitudes, suggesting that CVEs with front-loaded bursts but lacking sustained activity are penalised by the model as weak long-term signals.

Overall, the SHAP distributions confirm that not only the magnitude but also the trajectory of early attention meaningfully informs predictive performance. This evidence support our subsequent decision to explicitly cluster CVEs by trend shape (Chapter 5), allowing models to specialise on distinct behavioural archetypes of GHP attention growth.

Raw + Shape + Context Feature Set

For TH3 dataset, adding contextual features (`activ_delay`, `cve_publish_year`, `delay_type`) systematically degraded model performance.

The SHAP profiles explain this outcome: `activ_delay`, although ranked among the top predictors, shows highly dispersed contributions with no consistent monotonic effect. This volatility indicates that delays in GHP activation do not provide stable predictive signal once early interaction dynamics are already accounted for. Year-based variables (`pub_2023`, `pub_2024`) contribute weakly, acting as noisy priors rather than meaningful differentiators. In short, with activation threshold = 3, the contextual features introduce instability without improving bias symmetry or over-all fit.

Conclusion

In conclusion, the SHAP analysis confirms that early GHP activity, both in terms of interaction volume and temporal shape, is the dominant driver of long-range CVE popularity forecasts. Temporal shape features, in particular, allow the model to differentiate between CVEs with comparable early interaction counts but divergent momentum patterns. This observation motivates our subsequent use of unsupervised clustering (Section 5.1) to group CVEs by trajectory shape, enabling more targeted modelling of distinct behavioural archetypes.

Chapter 5

Trend Clustering

The preceding evaluation of global models highlighted a persistent limitation: population heterogeneity. Despite strong aggregate performance, even finely tuned models struggled to balance accuracy and residual symmetry across all GHP types. Meanwhile, our SHAP-based feature attribution confirmed that early interaction shape, beyond raw-count volume, enabled the model to better distinguish between GHPs with similar early counts but differing momentum.

We therefore adopt a more principled strategy: explicitly partitioning the GHP population by early growth dynamics. This section introduces our clustering framework, extracting these temporal shape signals directly from interaction time-series, which aims to isolate distinctive attention trend trajectories, such as flat cold-starters, front-loaded bursts, and jittery or irregular growth patterns.

We choose TH3 dataset for its better coverage of early GHP interaction trajectories (see Section 4.2.3). We conduct clustering on two early input windows (5 and 7 days), consistent with prior model configurations. A z-score normalisation on the interaction time-series ensures that clustering is shape-based, not size-based.

5.1 Clustering Method

To capture meaningful early-shape subgroups, we adopt HDBSCAN ([Malzer and Baum, 2020]) as our primary clustering method. HDBSCAN is a density-based algorithm that identifies clusters as dense regions of similar points in the embedding space. This property is particularly well suited to our dataset, where CVEs exhibit heterogeneous and often noisy early interaction patterns. HDBSCAN can flexibly capture local structure in this space, for example, dense pockets of bursty growth, without relying on uniformity assumptions.

Traditional clustering methods such as KMeans require the number of clusters to be specified in advance and typically perform best when clusters are roughly spherical and equally sized. By contrast, HDBSCAN infers the number of clusters directly from the data. Its behaviour is controlled by a small set of interpretable parameters:

- `min_cluster_size`: the minimum number of CVEs required to form a dense region;
- `min_samples`: controls the level of strictness when separating core from border points, influencing noise tolerance.

This design eliminates the need for prior assumptions about the number of trend types, allowing behavioural archetypes to emerge naturally from the data. In addition,

HDBSCAN's built-in noise detection automatically excludes ambiguous or weakly clustered CVEs, resulting in cleaner, more interpretable clusters that better reflect coherent early-growth dynamics.

For comparison, we also include KShape clustering using Shape-Based Distance (SBD) as a baseline method (see Section 5.2.3). KShape is specifically designed for time-series data: it aligns input sequences via phase-invariant cross-correlation and computes centroids that preserve temporal shape. This allows us to evaluate how well density-based methods like HDBSCAN perform relative to fixed-partition, shape-aware alternatives.

5.1.1 Grid Search and Clustering Evaluation

To ensure robust discovery of trend-based CVE archetypes, we conduct a multi-stage grid search over key HDBSCAN parameters, in conjunction with UMAP projection to reduce the dimensionality of z-normalised interaction timelines.

Our search explores the following parameter space:

- UMAP dimensions: {3, 5, 7};
- Minimum cluster size (`min_cluster_size`): $[25, 50]$;
- Minimum samples (`min_samples`): $[20, 35]$, tuned to correspond proportionally to the selected `min_cluster_size`;
- Distance metric: {Euclidean, Manhattan, Correlation};
- Cluster selection method: {EOM (excess of mass), Leaf}.

We first conduct the grid search using z-normalised interaction time-series for both 5-day and 7-day input windows. This initial phase focuses on exploring combinations of UMAP dimensions, `min_cluster_size`, and `min_samples`. Based on the strongest candidates from this stage we perform a second, finer-grained search across distance metrics and cluster selection methods (`eom`, `leaf`) to optimise the final cluster structure.

Each configuration is evaluated using a combination of quantitative metrics and qualitative diagnostics, designed to assess internal coherence, separability, and downstream utility for time-series modelling.

5.1.2 Evaluation Metrics

To evaluate the quality and interpretability of each clustering configuration, we apply a combination of quantitative and visual metrics. These are designed to assess not only statistical structure, but also the operational relevance of emerging behavioural groups.

Silhouette Score

We compute the mean silhouette score across all non-noise CVEs, capturing how well each CVE aligns with its assigned cluster compared to others. Higher values indicate stronger intra-cluster cohesion and inter-cluster separation, suggesting that early interaction shapes are consistently grouped.

Noise Percentage

We monitor the proportion of CVEs labelled as noise. While moderate noise levels are expected, excessively high noise rates (e.g., above 20%) may signal overly aggressive filtering or an embedding space that fails to support cluster-able structure.

Cluster Count and Size Distribution

For downstream analysis and forecasting, we favour configurations that yield a moderate number of clusters (typically 3–8), each with a sufficient number of samples (ideally

>30 CVEs). Clusterings that fragment the data into many small groups are penalised, as they reduce generalisability and complicate interpretability.

Cluster UMAP Visualisation

We project cluster assignments onto the 2D UMAP space to visually assess regional coherence and spatial separation. Effective clusterings display well-defined, non-overlapping regions with minimal noise bleed or overlap between trend types.

Trend Shapes

For each candidate configuration, we compute cluster-wise median interaction curves, based on z-score normalised 5-day or 7-day timelines anchored to the activation day. Preferred clusters should exhibit distinct and interpretable temporal shapes, as these trend archetypes form the behavioural foundation for subsequent forecasting analysis and cluster-specific modelling.

5.2 Clustering Trend Result and Evaluation

In this section, we evaluate candidate clustering configurations based on both quantitative metrics and the trend shape quality. Our goal is to identify a configuration that strikes a balance between statistical cohesion, shape interpretability, and prediction utility.

5.2.1 Initial Grid Search Setup

As previously mentioned, we begin with a broad grid search using:

- UMAP dimensions: 3D and 5D,
- HDBSCAN parameters: `min_cluster_size` $\in [25, 50]$, `min_samples` $\in [20, 35]$,
- Distance metric: Euclidean,
- Selection method: EOM (Excess of Mass).

Each configuration is applied to z-score normalised 5-day interaction timelines, anchored to the day of GitHub PoC activation.

5.2.2 Evaluation

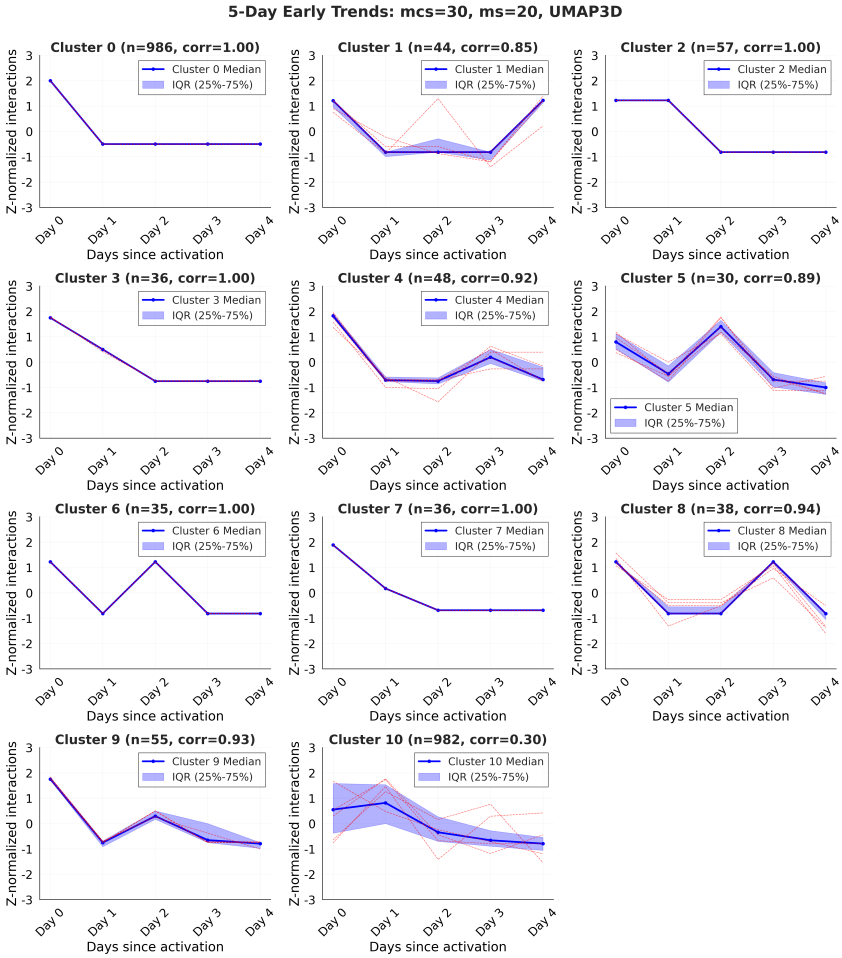
Table 5.1 summarises key metrics from representative configurations:

Table 5.1: Summary of Clustering Configurations and Evaluation Metrics (5-Day Input).

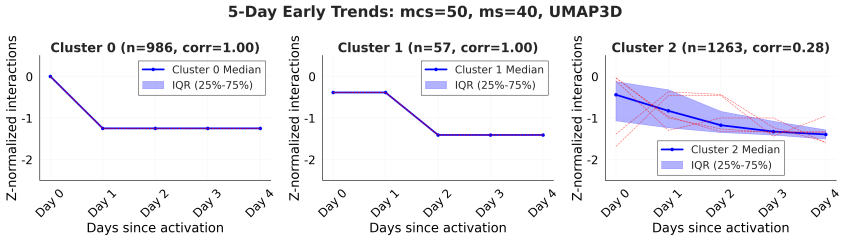
mcs	ms	UMAP Dim	#Clusters	Noise %	Silhouette	Largest Cluster
30	20	3D	11	1.5%	0.495	986
30	20	5D	9	0.0%	0.556	1076
35	25	3D	10	1.8%	0.512	986
35	25	5D	9	0.0%	0.556	1076
40	30	3D	5	1.5%	0.499	1212
40	**30**	**5D**	**6**	**1.5%**	**0.589**	**1111**
45	35	3D	6	6.4%	0.660	986
45	35	5D	5	4.7%	0.594	1079
50	40	3D	3	3.2%	0.658	1263
50	40	5D	4	3.9%	0.632	1183

Lower `min_cluster_size` values tend to produce a larger number of clusters and minimal noise rates, but often at the cost of trend redundancy. Upon closer inspection of the resulting median trend shapes, these configurations frequently yield clusters that are scattered, overly similar, or behaviourally indistinct.

Conversely, the more coarse-grained configurations show the opposite failure mode: they collapse diverse temporal trajectories into oversized general-purpose clusters, undermining interpretability and learnability. These contrasting effects are illustrated in Figure 5.1, which visualises both trend redundancy under over-fragmentation and the shape dilution associated with excessive aggregation.



(a) Over-fragmented clustering result



(b) Coarse-grained clustering result

Figure 5.1: Comparison of two clustering configurations, visualised as median interaction curves for the 5-day input setting.

Overall, the clustering behaviour is highly sensitive to the trade-off between cluster

granularity and shape distinctiveness. As highlighted in Table 5.1, we select 5day_mcs40_ms30_umap5d as our final clustering configuration. This setup offers a moderate number of clusters (6), low noise rate (1.5%), balanced silhouette score (0.589), and clearly differentiated and interpretable trend shapes.

Distance Metric and Selection Method Tuning

To refine our final clustering configuration, we conduct a focused grid search over distance metrics (`euclidean`, `manhattan`, `correlation`) and cluster selection methods (`eom`, `leaf`) while holding other parameters fixed (`mcs=40`, `ms=30`, `UMAP=5D`, 5-day input window).

Table 5.2: Evaluation of Distance Metrics and Cluster Selection Methods (5-Day Input).

Metric	**Selection Method**	**# Clusters**	**Noise %**	**Silhouette Score**	**Largest Cluster**
euclidean	eom	6	1.511	0.589	1111
euclidean	leaf	13	51.238	0.556	235
manhattan	eom	4	1.511	0.626	1260
manhattan	leaf	14	45.531	0.575	195
correlation	eom	12	24.759	0.702	986
correlation	leaf	13	49.895	0.225	257

As shown in Table 5.2, `leaf` selection consistently yields high noise rates, often exceeding 30%, and generates scattered clusters with low cohesion. This instability is observed across all distance metrics. We therefore prioritise `eom` (Excess of Mass) as the selection method for all further experiments.

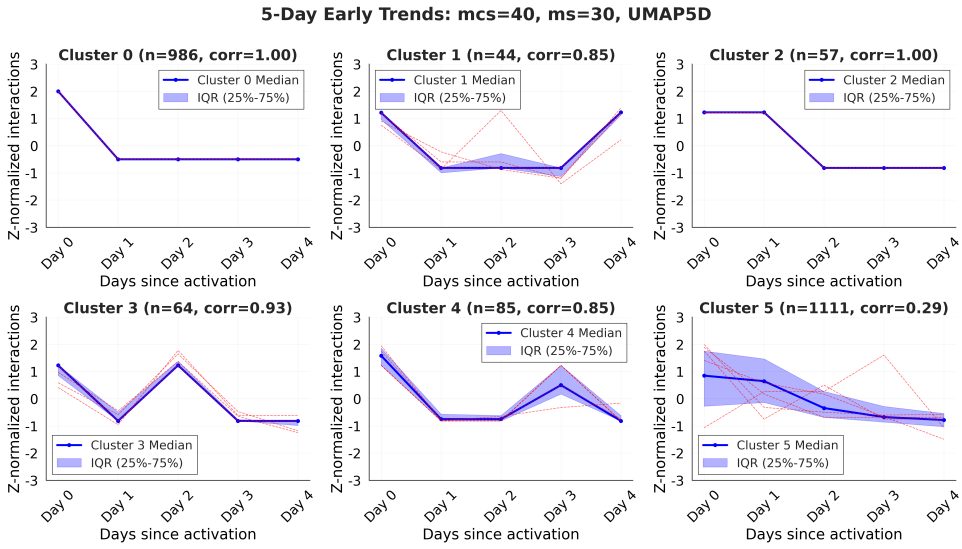
Among distance metrics, correlation-based clustering consistently produces more fragmented results, yielding finer clusters at the cost of increased noise (e.g., 24.76% for correlation–eom). By contrast, Euclidean and Manhattan metrics yield similar outcomes under `eom`, with Manhattan achieving a slightly higher silhouette score (0.589 vs. 0.626), but producing only four clusters compared to Euclidean's six.

We therefore kept Euclidean–EOM as the final metric–selection pair for 5-day clustering. The final clustering results are visualised in Figure 5.2.

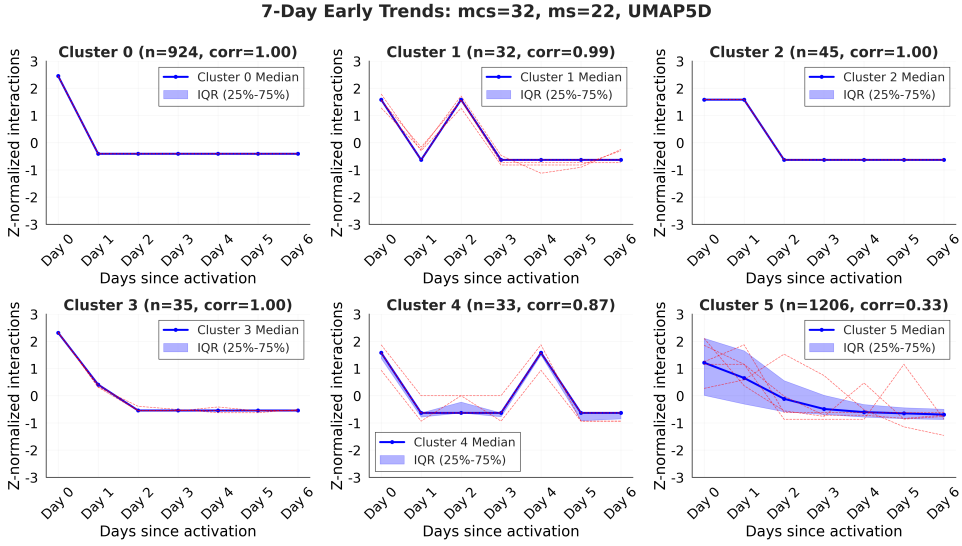
For the 7-day input window, we apply the same two-stage grid search and selection logic. The configuration with mcs = 32, ms = 22 and umap 5d demonstrates the best clustering result, while maintaining smooth continuity with the clusters identified in the 5-day window. The result trend shapes are shown in Figure 5.2. Further analysis of the 7-day cluster behaviours is presented in Section 5.3.

5.2.3 K-shape + SBD Method Comparison

To assess the value of our HDBSCAN-based clustering strategy, we compare it against a traditional time-series clustering method: KShape with Shape-Based Distance (SBD). Both methods operate over z-normalised 5-day GHP interaction timelines. For consistency, we set the number of clusters in KShape to 6, mirroring the best-performing HDBSCAN configuration (`5day_mcs40_ms30_umap5d`).



(a) Interaction curves for the 5-day input setting.



(b) Interaction curves for the 7-day input setting.

Figure 5.2: Final clustering results visualised as interaction curves for the 5-day and 7-day input windows. The dotted red lines highlight examples of individual CVE trends.

While the interaction trajectories of HDBSCAN clusters (Figure 5.2) reveal distinct and explainable archetypes, KShape's centroid curves (Figure 5.3) show significantly less temporal differentiation. Multiple clusters converge toward similar trend shapes, with wider IQRs, indicating lower intra-cluster consistency. This lack of distinctiveness reduces the interpretability and modelling value of the clustering output. We observe this most clearly in cold-start CVEs: while HDBSCAN' cluster 0 cleanly isolates this population, KShape's Cluster 1 is internally more noisy and heterogeneous, resulting in a diffuse grouping.

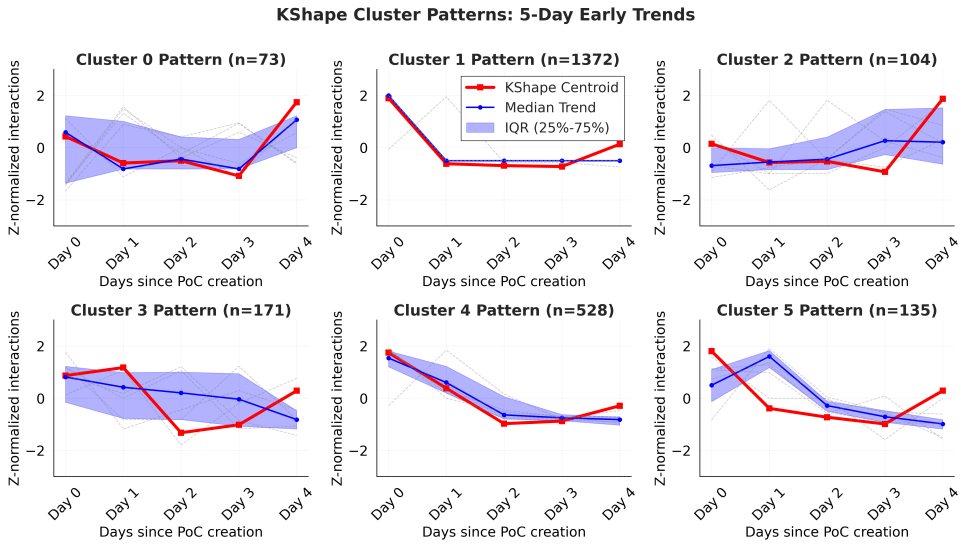
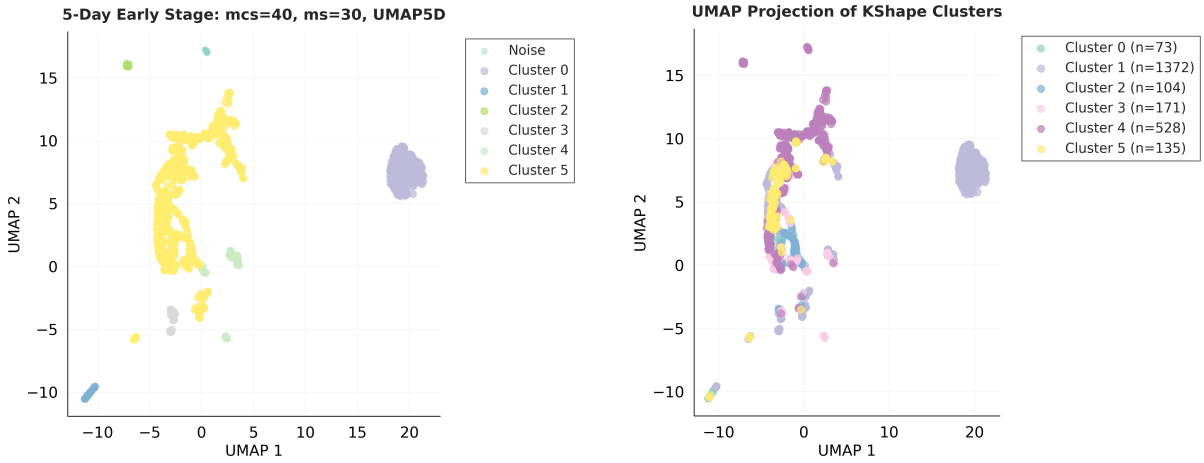


Figure 5.3: KShape Clustering result as median trend. The dotted grey lines highlight examples of individual CVE trends.

Furthermore, as shown in Figure 5.4, the HDBSCAN cluster projection in UMAP space forms well-separated, compact regions. Each cluster occupies a distinct subspace, reflecting meaningful divisions in early interaction behaviour. By contrast, the UMAP projection of KShape clustering reveals substantial overlap among flat or weak-signal trajectories. Cluster 1, in particular, disperses across multiple regions, diluting interpretability and reducing its diagnostic value. The method's silhouette score of only 0.318, consistent with its weak spatial distinction, underscores KShape's limitations in capturing subtle, low-momentum trend patterns in this research setting.



(a) UMAP projection with HDBSCAN clustering.

(b) UMAP projection with KShape clustering.

Figure 5.4: UMAP 2D clustering projections for CVE interaction shapes.

Taken together, while KShape succeeds at aligning raw time series, it fails to generate actionable, shape-coherent clusters in the short 5-day setting. In this case, HDBSCAN offers better cluster separation and clearer archetypes, thus is more effective for early GHP trend detection.

5.3 Cluster Profiling

To better understand the behavioural archetypes uncovered by our trend clustering framework, we analyse the final clustering results for both the 5-day and 7-day input windows. As shown in Figure 5.2, cluster boundaries evolve between the two time windows. From the 5-day to the 7-day input, some trends persist, while others are merged, restructured, or refined. Despite these boundary shifts, the underlying behavioural archetypes remain stable, preserving their explanatory power and interpretability across time.

5.3.1 Trend Archetypes

Cluster 0 (5-day vs 7-day) — Cold Starters
Both clusterings robustly isolate the same archetype: CVEs with near-zero activity beyond Day 1. These cold-starters exhibit flat, dormant trajectories across the early window and demonstrate perfect shape consistency (correlation = 1.00). While the cluster size drops from 986 CVEs in the 5-day configuration to 924 in the 7-day, this reduction reflects a cleaner and more conservative grouping, filtering out CVEs that begin to show late interaction signals on Day 6 or 7. The persistence of this cluster across window lengths confirms its status as a foundational behavioural group, especially relevant for modelling low-signal, hard-to-predict GHPs.

Cluster 5 (5-day vs. 7-day) — Long-tail Tapers
These clusters are the largest in both input windows, reaching 1111 CVEs in the 5-day configuration and 1206 in the 7-day. They exhibit strong early activity, typically peaking at Day 0 or Day 1, followed by a smooth, gradual decline, though attention persists throughout the window. The modest increase in cluster size under the 7-day input suggests better absorption of borderline cases, especially short-term bursts that were ambiguous under the 5-day model.

While the intra-cluster correlation scores indicate looser alignment in precise temporal shape, subsequent prediction analysis (see Section 6.1) confirms that this cluster yields strong predictive performance, reinforcing its relevance as a coherent and operationally meaningful group.

Among the smaller clusters, trend shapes also evolve distinctly when extending the observation window from 5 to 7 days:

- **Early Plateauers (C2 → C2):** modest activation followed by a quick flattening; smaller in size at 7 days (45 vs. 57), indicating improved separation from dormant and late-rising CVEs.
- **Oscillators (C1 → C4):** rebound-like bursts that consolidate more clearly under 7 days, confirming oscillatory behaviour as genuine rather than noise.
- **Refined Rebounders (C3 → C1):** initial rebounds become sharper at 7 days, with flatter variants redistributed into neighbouring clusters.
- **Refined Decliners (C4 → C3):** early peak and steady decline re-emerge as a smaller, purer trajectory, with ambiguous cases absorbed elsewhere.

All four pair of clusters show a shared pattern: extending the observation horizon consistently reduces cluster sizes while sharpening their internal coherence, leaving behind cleaner and more interpretable behavioural shapes. Ambiguous members that blurred boundaries at Day 5 are redistributed into neighbouring archetypes by Day 7, most likely into Cluster 5, the long-tail taper group.

Table 5.3 summarises the recurring patterns observed across both the 5-day and 7-day input windows.

Table 5.3: Behavioural Archetypes Across 5-Day and 7-Day Clusters

Behaviour Type	**5d Cluster(s)**	**7d Cluster(s)**	**Description**
Cold-starters	Cluster 0	Cluster 0	Highly consistent flat trend after Day 1
Noisy oscillators	Cluster 1, 3	Cluster 1, 4	Rebound bursts or fluctuating oscillations after initial drop
Early plateauers	Cluster 2	Cluster 2	Modest initial signal, then quickly dormant
Refined decliners	Cluster 4	Cluster 3	Early spike followed by steady decline
Long-tail tapers	Cluster 5	Cluster 5	Gradual and extended decline with high intra-cluster variance

In conclusion, while the 7-day window provides clearer disambiguation of shape ambiguity, particularly in non-monotonic or low-signal CVEs, both the 5-day and 7-day configurations consistently capture the same four core behavioural archetypes: cold-starters, front-loaded tapers, early-plateauers/decliners, and rebound-oscillators.

5.3.2 Plateau Timing Analysis Across 5-Day Clusters

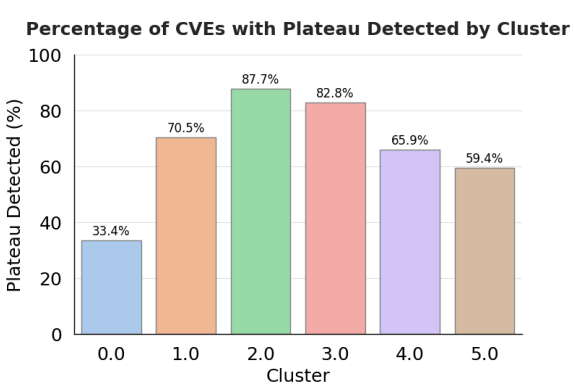
To test our interpretation of trend archetypes, we conduct a small-scale plateau timing analysis across the 5-day clustering result. We define a plateau point as the moment when public engagement meaningfully tapers off after a CVE's initial burst of attention.

Operationally, a CVE is considered to have plateaued if its daily interaction count remains below 3% of its peak value for at least 15 consecutive days. This thresholding heuristic offers a consistent and interpretable marker of GHP attention deceleration, and reflects realistic disengagement patterns in public PoC interest.

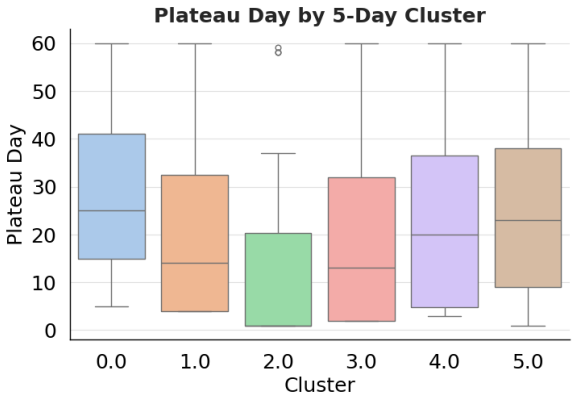
As shown in Figure 5.5, Cluster 2 (early plateauers)exhibits both the highest plateau detection rate (87.7%) and the earliest median plateau onset across clusters. This confirms its profile as one defined by short-lived but genuine early engagement, followed by rapid dormancy.

At the opposite end, Cluster 0 (cold starters) shows the lowest plateauing rate (33.4%), aligning with its defining flat trajectory and negligible early activation. These CVEs rarely accumulate enough signal to register a plateau event, gathering ecosystem attention well past the observation window.

Clusters 1 and 3 (oscillatory groups) both plateau at high rates (70.5% and 82.8% respectively), but with somewhat earlier onsets than tapering groups. This suggests that despite their rebound behaviour, these CVEs still decelerate into stability relatively quickly once secondary bursts subside.



(a) Percentage of CVEs per cluster that plateaued.



(b) Distribution of plateau timing across clusters (in days post-activation).

Figure 5.5: Plateau analysis across PoC trend clusters.

Clusters 4 and 5 (tapering groups) plateau at intermediate frequencies (65.9% and 59.4%) and show broader distributions of plateau onset days. This reflects their longer-lived decline trajectories, where interaction levels diminish more gradually over extended periods.

Together, these findings reinforce the validity of the clustering scheme: the alignment between early shape-based groups and long-term engagement dynamics underscores both the interpretive and predictive value of the identified trend archetypes.

Chapter 6

Cluster-wise Prediction

6.1 Cluster-wise Model Performance Evaluation

In this chapter, we evaluate the global prediction model's performance at the cluster level, examining how different early-shape trend archetypes impact forecasting accuracy. Here, we recalculate the global prediction results within each cluster, using the CVEs' assigned cluster IDs to derive performance metrics specific to that group. This analysis allows us to isolate where the model generalises well and where it consistently fails, offering practical guidance for cluster-wise prediction model refinement.

We focus our evaluation on three long-range I/O configurations: 5→60, 5→90, and 7→90, which offer the most promising balance between forecastability and operational lead time (see Section 4.2.2). For comparison, we also include the 7→30 configuration as a performance reference, since it achieved the strongest results in the TH3 model, with the lowest log-scale MAE (0.236) and highest Log R^2 (0.940), serving as a best-performance benchmark.

Figure 6.1 presents four diagnostic scatter plots for these configurations. In each plot, the y-axis is fixed to Log MAE, allowing for consistent comparison of prediction error across clusters. The x-axes vary across key evaluation dimensions: Log R^2, Absolute R^2, Log Residual Skewness, and SHAP Volatility (defined as the average interquartile range (IQR) across all SHAP features within a cluster, capturing the diversity of learned feature attributions).Each point in the plot represents a cluster under a specific I/O configuration. Marker size reflects the number of CVEs in the cluster, while colour intensity corresponds to Outlier Density, the percentage of CVEs within the cluster whose absolute prediction error exceeds a threshold of 100.

A complete heatmap of cluster-wise evaluation metric scores across the 5→60, 5→90, 7→30, and 7→90 configurations is provided in Appendix Figure 1.

Cluster 0 (5-day & 7-day) — Cold-starters

As described in Section 5.3, Cluster 0 represents a highly consistent trend archetype across both the 5-day and 7-day configurations, namely, the cold-starters. CVEs in this group exhibit flat, dormant interaction trajectories, with little to no signal during the early input window. Consequently, the global model performs poorly: residual skewness is strongly negative (approx. -5.0), Log R^2 hovers near zero or negative values, and SHAP volatility is extremely low (approx. 0.02). This indicates the model is relying on a narrow, low-variance feature set which offers limited explanatory value due to the uniform flatness of the input signals.

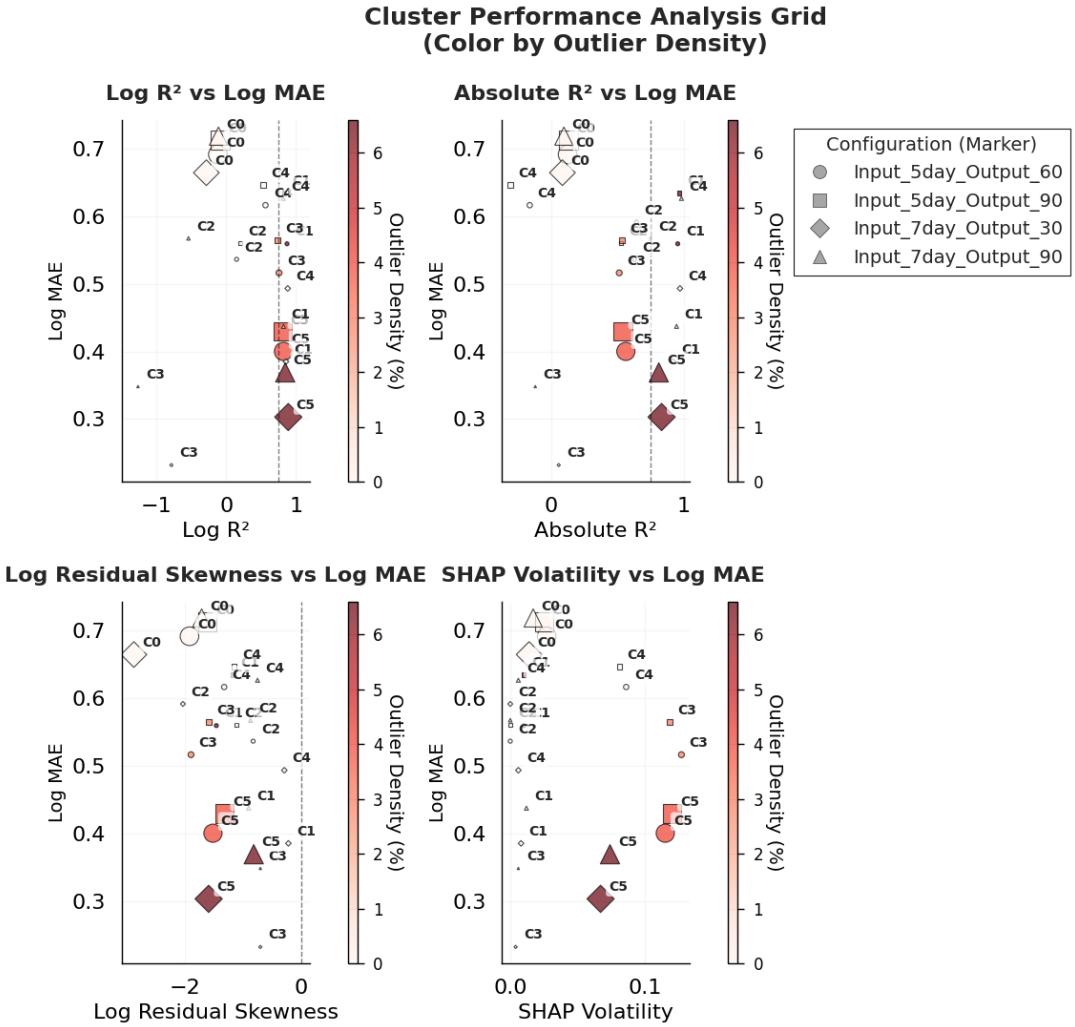


Figure 6.1: Cluster Performance Analysis Grid. Four diagnostic scatter plots visualising cluster-level prediction results for selected I/O configurations (5→60, 5→90, 7→30, 7→90). Each subplot uses Log MAE as the y-axis and varies the x-axis as follows: (1) Log R^2, (2) Absolute R^2, (3) Log Residual Skewness, (4) SHAP Volatility. Point colour encodes outlier density (percentage of CVEs with large prediction errors), and point size reflects cluster size. Labels denote cluster identity (e.g., C0 = Cluster 0).

Cluster 0 represents a structural blind spot. These CVEs simply lack informative early-stage GHP activity, rendering them effectively invisible to models trained on short input windows. To improve learnability for this group, two complementary strategies may be required: (1) extending the input window to capture delayed activation patterns, and (2) incorporating non-GHP contextual features, such as vulnerability type, exploitability heuristics, or disclosure metadata.

Since 7-day Cluster 0 is a cleaner, more homogeneous cold-start cohort, we adopt Cluster 0 from the 7-day input window as the basis for cluster-wise prediction, detailed in Section 6.2.1.

Cluster 5 (5-day & 7-day) — Long-tail Tapers

This cluster pair captures the trend shape with an immediate spike in interactions around Day 0 and Day 1, followed by a gradual and consistent taper in engagement. This trend is where the model performs most consistently well, achieving Log R^2 scores in the range of approx. 0.80–0.89 and Log MAE between approx. 0.30 and 0.43 across

all long-range I/O configurations. Performance gains are underpinned by high SHAP volatility ($\approx$ 0.07–0.12), indicating that the model leverages a diverse set of interacting features to explain the variation within this group.

However, as shown in the third plot (residual skewness & Log MAE), the consistently negative residual skew and elevated outlier density means this archetype holds most high-attention GHPs and exhibits systematic under-prediction. To address this limitation, we select 5 day Cluster 5 for further cluster-wise enhancement, for this setting offers a shorter input window with practical lead time. Its combination of high learnability and persistent residual skewness makes it an ideal testbed for targeted refinement. In particular, we explore whether techniques such as asymmetric loss functions, quantile regression, or feature augmentation can improve model bias in skewness, without sacrificing overall fit. These enhancement strategies are detailed in Section 6.2.2.

For the remaining smaller size clusters, the sample size for test set gets highly limited (around 10-20), which constrains the reliability of absolute error metrics. Nonetheless, their relative R^2 scores and SHAP volatility values provide useful insight into which temporal archetypes are learnable and which reflect intrinsic unpredictability.

Cluster 2 (5-day & 7-day) — Early Plateauers

These clusters consist almost entirely of dormant CVEs, with modest activation followed by stagnation. Their log-R^2 scores are modest (0.15–0.20 in 5-day), while absolute R^2 remains relatively high (0.53–0.75). This discrepancy arises because absolute scale tends to overstate predictive performance for consistently flat shapes. The accompanying zero outlier density and minimal SHAP volatility indicate that the model is not uncovering genuine structure, but merely reproducing the lack of variation in the data.

Cluster 1 (5-day) → Cluster 4 (7-day) — Oscillators

Oscillatory CVEs achieve consistently high predictive fit, with log-$R^2 \approx 0.83$–0.87 and absolute $R^2 \approx 0.95$–0.98. The model clearly aligns with their non-monotonic rebound patterns. However, SHAP volatility remains near zero, which suggests overfitting to a stereotyped surge pattern: the model recognises the "rebound template" but gains little diagnostic insight into feature variation.

Cluster 3 (5-day) → Cluster 1 (7-day) — Refined Rebounders

In the 5-day view, Cluster 3 achieved moderate fit (log-$R^2 \approx 0.76$, absolute $R^2 \approx 0.51$), coupled with higher SHAP volatility, reflecting heterogeneous rebound shapes. At 7-day, these re-emerge as Cluster1, with log-R^2 of 0.82 and absolute R^2 near 0.95, but with reduced volatility, reflecting a more uniform rebound pattern.

Cluster 4 (5-day) → Cluster 3 (7-day) — Refined Decliners

5-day Cluster 4 showed moderate log-R^2 (0.56) but weak absolute fit ($R^2 \approx -0.16$). By 7-day, the restructured Cluster 3 is smaller and more homogeneous, yet its absolute R^2 remains negative, signalling that these declining trends are learnable in relative terms but harder to capture on the raw scale.

From Figure 6.1, it is evident that trend archetypes with similar shapes generally appear close to each other across evaluation metric spaces. This alignment suggests that trend clusters not only reflect coherent temporal behaviours, but also exhibit similar prediction performance. In the subsequent cluster-wise prediction experiments, we further

examine how these behavioural distinctions translate into predictive utility.

6.2 Cluster-wise Prediction

6.2.1 Cluster 0 - 7day

We begin our cluster-specific refinement with Cluster 0 from the 7-day input window, comprising cold-start CVEs with flat early interaction patterns and minimal signal within the first few days.

We first conduct a cluster-specific prediction within Cluster 0, using the same 7→90 configuration as the global model, to assess whether localisation alone, without extending the input window, can improve model performance by isolating a more behaviourally coherent subgroup.

Table 6.1: Cluster 0 (7-day): Global vs Local Model Performance (7→90 configuration)

Model	IO window	Log R^2	R^2 (Abs)	Log MAE	Residual Skew
Global	7-90	-0.109	0.095	0.718	-1.718
Local	7-90	0.306	0.083	0.345	-2.47

Under the global model with a 7-day input window, Cluster 0 achieves a log R^2 of –0.109 and Log MAE of 0.718, reflect the model's inability to capture meaningful variance for CVEs whose growth occurs well beyond the input horizon. Naturally, the improvement in Log R^2 (from –0.109 to 0.306) and a corresponding drop in Log MAE (from 0.718 to 0.345) indicate improved shape learning. However, the residual skewness becomes more negative (–2.47), suggesting that the model still fails to predict the late risers.

These results confirm that while localisation improves learnability, the short input horizon remains a bottleneck. The next step involves evaluating whether extending input window length to 14-day or 30-day inputs could allow the model to better detect emerging growth trends.

Extended Input Windows

Table 6.2: Cluster 0: Local Model Performance Across IO Configurations

IO Window	Log R^2	R^2 (Abs)	Log MAE	MAE	Residual Skew
7 → 30	0.363	0.059	0.214	0.87	-3.72
14 → 30	0.601	0.173	0.135	0.60	-5.34
7 → 60	0.326	0.097	0.300	1.25	-2.82
14 → 60	0.535	0.366	0.231	0.94	-3.61
30 → 60	0.843	0.906	0.132	0.49	-2.56
7 → 90	0.306	0.083	0.345	1.55	-2.47
14 → 90	0.497	0.440	0.287	1.20	-2.81
30 → 90	0.712	0.782	0.218	0.92	-1.90

To evaluate whether longer early observation periods can improve learnability in Cluster 0, we extend the input window from 7 to 14 and 30 days while keeping the output

window fixed (30, 60, or 90 days). The feature set is held constant in form, consisting of raw interaction counts and derived shape-based metrics; it is extended only in temporal coverage, incorporating additional days of observations as the input window lengthens.

Across all output horizons, we observe consistent performance gains with input extension:

- Log R^2 improves systematically, from only 0.31 in the 7→90 setting to 0.84 in the 30→60 configuration. Absolute R^2 also shows marked improvements, rising from near-zero levels in the short 7-day settings (0.06–0.10) to as high as 0.91 in the 30→60 configuration.
- Log MAE and MAE both decline steadily, reaching as low as 0.13 (log scale) and 0.49 (absolute) in the 30→60 configuration. Similarly, RMSE contracts from values above 3.5 at 7→90 to below 1.0 at 30→60, confirming that outlier cases are better accommodated once more early signals are captured.
- Residual skewness remains negative across all settings, reflecting systematic underprediction of late-rising CVEs. However, the relatively low MAE and RMSE values across all configurations suggest that the majority of CVEs in this cluster attract modest attention. As a result, the residual skew is unlikely to distort real-world triage, since most underpredicted cases remain within a low-risk range.

These results confirm the hypothesis that Cluster 0's poor early predictability stems from input truncation. An extended observation window unlocks the model's true learnability, where the 30-day input window emerges as sufficient, resulting in reliable prediction performance across 60 and 90 output horizons.

Raw + Shape + Contextual Feature Set

Next, we introduce contextual features to the cluster-specific model to see if external temporal and ecosystem context could improve performance.

Table 6.3: Cluster 0: Performance Delta (Raw+Shape+Contextual minus Raw+Shape)

IO Window	ΔLog R^2	ΔR^2 (Abs)	ΔLog MAE	ΔMAE	ΔSkew
7 → 30	-0.055	+0.056	+0.027	+0.05	+0.77
14 → 30	-0.078	+0.053	+0.010	+0.05	+1.34
7 → 60	-0.084	-0.022	+0.007	+0.04	+0.77
14 → 60	-0.118	-0.032	+0.008	+0.04	+0.89
30 → 60	-0.022	-0.022	+0.004	+0.06	+1.25
7 → 90	-0.068	-0.003	+0.002	+0.04	+0.73
14 → 90	-0.117	-0.046	+0.001	+0.03	+0.51
30 → 90	+0.015	+0.014	-0.025	-0.08	-0.36

Table 6.3 shows the performance difference between the Raw+Shape baseline and the Raw+Shape+Contextual feature sets. The inclusion of contextual features has mixed effects.

On the positive side, residual skewness decreases in almost all configurations, moving closer to zero and indicating reduced systematic underprediction of late-rising CVEs. This effect is most pronounced at 30→60, where skew improves by +1.25 (from −2.56 to −1.31). The only exception is at 30→90, where skew worsens slightly (−0.36).

On the other hand, explanatory performance generally declines. Across 7- and 14-day inputs, both Log R^2 and Absolute R^2 drop substantially (e.g. ΔLog $R^2 = -0.118$ and $\Delta R^2_{\text{Abs}} = -0.032$ at 14→60). Similarly, error metrics (Log MAE and MAE) mostly increase, suggesting that contextual features introduce noise rather than improving predictive power. The only configuration where both fit and error improve is 30→90, though gains remain marginal.

Overall, for Cluster 0, the current contextual feature set acts primarily as a calibration aid: it reduces underprediction bias but does not enhance, and often degrades, explanatory performance. In practical terms, contextual metadata helps to balance systematic skew, but raw and shape-based features remain the dominant drivers of accuracy.

Raw + Shape (Extended) Feature Set

As a further enhancement step, we investigate whether extended trend-based feature engineering can improve prediction performance for Cluster 0. Specifically, we examine the GHP interaction patterns from Day 7 to Day 30, using sub-cluster trend analysis and raster histograms to isolate common late-rising attention behaviours. For each input window (14-day and 30-day), we design an extended feature set, to capture patterns such as near-Day 14 growth, rebound dynamics, entropy of tail behaviour, and post-peak decay patterns.

To further explore trade-offs in feature capacity, we conducted a two-stage ablation study:

1. First, we removed low-SHAP-contribution features and de-correlated redundant variables.
2. Next, we reintroduced only features that consistently appeared in the top 20 SHAP contributors across validation folds.

Both ablated variants underperformed relative to the full extended feature set, confirming that no smaller or cleaner subset yielded superior predictive value. (Full ablation results are provided in Appendix Table 3 and 4.)

These results suggest that we have reached a diminishing return point for feature engineering based solely on GHP signals. While extended shape features offer marginal gains in residual calibration, they are not sufficient on their own to significantly improve broader model performance. This finding reinforces the need for alternative or extensive contextual signals to improve prediction for cold-start CVEs.

6.2.2 Cluster 5 - 5day

Cluster 5 represents a long-tail taper archetype, characterised by a sharp early spike in attention followed by a smooth decline. This group remains one of the most predictable trend shapes: global models achieve log R^2 between 0.81–0.82 and absolute R^2 between 0.54–0.56 across the 5→60 and 5→90 horizons, with errors (MAE 55–58) well aligned to the cluster's growth scale.

As with Cluster 0, we retrained local models using only raw + shape features. Results (Table 6.4 and Table 6.5) confirm that localisation yields no performance benefit. While log-scale metrics remain strong, absolute R^2 falls slightly at 5→60 (0.521 vs. 0.563) and improves only marginally at 5→90 (0.550 vs. 0.541). MAE values remain effectively

Table 6.4: Cluster 5 (5-day): Model Performance Across Feature Sets and Output Horizons

Model	IO Window	R^2 (Log)	R^2 (Abs)	Log MAE	Residual Skew	MAE
Global	$5 \rightarrow 60$	0.823	0.563	0.400	-1.524	54.64
	$5 \rightarrow 90$	0.808	0.541	0.429	-1.318	58.32
Local	$5 \rightarrow 60$	0.815	0.521	0.390	-1.510	54.82
	$5 \rightarrow 90$	0.795	0.550	0.413	-1.330	56.96
Local-Subcluster	$5 \rightarrow 60$	0.826	0.414	0.383	-1.480	61.48
	$5 \rightarrow 90$	0.813	0.302	0.406	-1.160	67.40

Table 6.5: Cluster 5 (5-day): Delta Performance Relative to Global Model ($5 \rightarrow 60$ and $5 \rightarrow 90$)

IO Window	Model	Δ Log R^2	Δ R^2 (Abs)	Δ Log MAE	Δ Residual Skew	Δ MAE
$5 \rightarrow 60$	Local	–0.008	–0.042	–0.010	+0.014	+0.18
	Local-Subcluster	+0.003	–0.149	–0.017	+0.044	+6.84
$5 \rightarrow 90$	Local	–0.013	+0.009	–0.016	-0.012	-1.36
	Local-Subcluster	+0.005	-0.239	-0.023	+0.158	+9.08

unchanged. These outcomes indicate that the global model already generalises well over the taper archetype without overfitting to other shapes.

In addition, we experimented with adding a sub-cluster ID feature, motivated by the relatively large size and internal heterogeneity of Cluster 5. Using HDBSCAN, we performed sub-clustering within Cluster 5 and included the resulting cluster IDs as one-hot encoded inputs alongside raw and shape-based metrics. However, this configuration also failed to surpass the global model's performance, suggesting that the primary trend archetype is already sufficiently representative and predictive, and therefore does not benefit from further structural segmentation.

Enhanced Regression Models

While cluster-local prediction confirms the global model's already strong performance on Cluster 5 (long-tail tapers), residual skewness persists, indicating systematic underprediction of high-impact CVEs. To address this, we explore enhanced regression formulations aimed at reducing skewness and improving sensitivity to high-impact CVEs.

Asymmetric Loss with Sample Weighting. First, we developed a custom regression strategy that combines asymmetric loss weighting with sample-level importance weighting to better address underprediction in high-impact CVEs. The combined training objective is passed to XGBoost via a custom DMatrix.

The asymmetric loss function increases the penalty for underpredicted cases by scaling the gradient when the residual (prediction minus target) is negative. After testing multiple configurations, we selected the setting that best balanced error minimisation and residual skewness reduction: $\alpha = 2.0$ and $\beta = 1.0$. This produces a piecewise linear loss function with an asymmetric slope, where underpredictions are penalised more strongly than overpredictions. To ensure compatibility with XGBoost's second-order optimiser, the loss includes a scaled Hessian term, preserving numerical stability during training.

In parallel, we apply a sample weighting scheme to reflect the varying importance of GHPs based on their interaction volume. Two weighting strategies were explored:

1. Scaled weighting: Each CVE is assigned a weight based on a power-scaled normal-

isation of its target value:

$$w_i = 1 + \left(\frac{y_i}{y_{\max}} \right)^{\alpha},$$

where $\alpha = 0.7$ controls the curvature. This design smoothly emphasises larger-target CVEs while maintaining continuity across the dataset, ensuring that high-impact cases receive proportionally greater influence without overly diminishing smaller ones.

2. Thresholded weighting: To more aggressively prioritise tail CVEs, we implement a stepwise weighting scheme informed by the empirical distribution of total interactions at Day 90:

   - CVEs with target > 150 receive a weight of 3.0
   - CVEs with target > 500 receive a weight of 20.0
   - CVEs with target > 850 receive a weight of 50.0

   All other samples retain a default weight of 1.0.

The threshold-based sample weighting strategy, combined with the asymmetric loss configuration described above, constitutes the final setup reported as the best-performing one for this model.

Quantile Regression Models. Secondly, we train quantile regression models to complement the asymmetric loss formulation. Unlike standard regressors that estimate the conditional mean, quantile models are designed to predict specified upper percentiles of the target distribution. In our case, we evaluate models trained to estimate the 75th, 90th, and 95th percentiles. This approach intentionally biases predictions upward, providing a conservative estimate that better captures tail risk.

For both asymmetric-loss and quantile regression models, we adopt refined hyperparameter settings to improve convergence and model stability. This includes adjustments to tree depth (e.g. `max_depth = 8`), learning rates (`eta = 0.03`–`0.05`), subsampling ratios(`subsample = 0.9`, `colsample_bytree = 0.8`), and objective-specific optimisers.

Table 6.6: Cluster 5 (5-day input): Comparison of Enhanced Regression Strategies Across Output Horizons

Model	Output Days	R^2 (Log)	R^2 (Abs)	Log MAE	Residual Skew	MAE
Baseline	30	0.841	0.561	0.353	–1.88	49.73
	60	0.815	0.521	0.390	–1.51	54.82
	90	0.795	0.550	0.413	–1.33	56.96
Asym. Loss	30	0.841	0.304	0.365	–1.51	58.86
	60	0.801	0.322	0.409	–1.24	64.39
	90	0.783	0.363	0.431	–1.13	65.52
Quantile 95	30	0.803	0.584	0.464	–1.53	60.11
	60	0.704	0.646	0.573	–0.75	65.71
	90	0.674	0.651	0.619	–0.94	71.81

Table 6.6 shows model performance across enhanced regression strategies, comparing baseline, asymmetric loss with sample weighting (`asym_loss_sw`), and quantile regression models at the 95th percentiles.

Among the enhanced regression strategies, the asymmetric loss with sample weighting (`asym_loss_sw`) does not provide the intended gains. While Log R^2 values remain close to the baseline, the absolute R^2 scores deteriorate sharply (e.g., 0.304 at 5→30 versus 0.561 for the baseline), and both MAE and RMSE increase. This suggests that reweighting the loss fails to improve calibration, and may even amplify noise in the already limited feature space.

By contrast, quantile regression models introduce a clearer trade-off. The 95th-percentile model shows a strongest bias toward the upper tail: while log R^2 notably declines (to 0.674 at 5→90), absolute R^2 surpasses the baseline (0.651 versus 0.550), and residual skewness improves materially, approaching symmetry (–0.75 at 60 days). This indicates that the model is better calibrated for high-volume CVEs, though at the cost of systematic overprediction on the majority class.

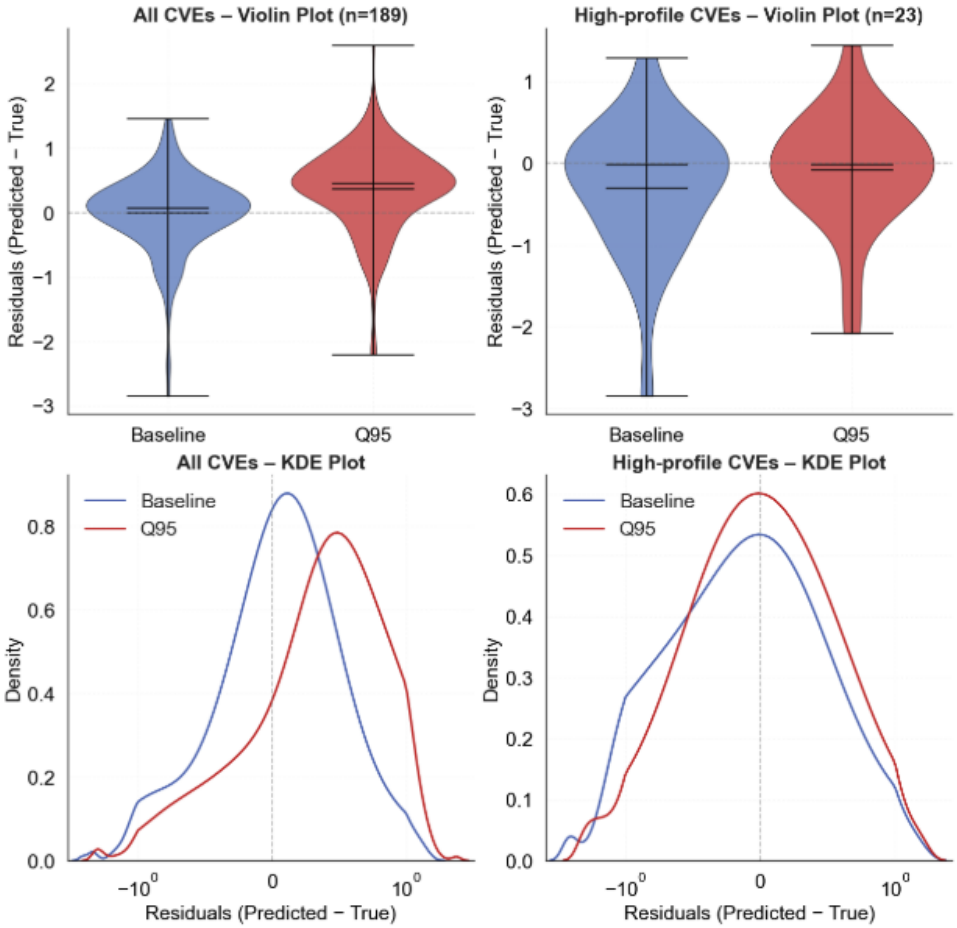


Figure 6.2: Residual distributions for baseline & Quantile 95 (Q95) models. **Left:** All CVEs in Cluster 5. **Right:** High-profile CVEs in Cluster 5 with over 150 total interactions at day-90.

Figure 6.2 presents violin and KDE plots comparing residual (defined as predicted - true) distributions between the baseline and Q95 quantile models. For all CVEs in cluster 5, the Q95 model exhibits slightly wider dispersion and a clear rightward shift, reflecting its intentional upward bias. On the other hand, within the high-profile subset (CVEs with total 90-day interactions >150), the Q95 model clearly shifts the residual distribution closer to zero and reduces the negative skewness observed in the baseline. The KDE plots reinforce this: while the baseline density peaks just left of zero (indicative

of systematic underprediction), the Q95 curve is more symmetric and better centered, suggesting improved calibration for impactful CVEs.

In conclusion, quantile-based models, specifically Q95, best aligns with our risk-sensitive objective. Its tendency to favour balance, or even mild overprediction, over systematic underestimation makes it a practical high-reference estimate for downstream prioritisation.

6.2.3 Small Cluster Prediction: Learnability and Limitations

This chapter investigates whether local prediction in small-sample clusters (with CVE counts around 50) offers any improvement over global models, or whether the prediction outcomes themselves help reinforce prior insights about cluster structure, learnability, and trend signal quality.

We tested two model configurations tailored to low-data scenarios:

- A conservative setup with `n_estimators = 300` and `learning_rate = 0.05`
- A slightly more aggressive variant with `n_estimators = 200` and `learning_rate = 0.1`

The conservative configuration consistently outperformed the alternative in log-space metrics and residual stability, and is therefore adopted for all small-cluster evaluations reported below. We also experimented with early stopping, but found that it degraded performance on these small clusters, likely due to premature convergence, so it is excluded from final results.

Table 6.7: 5-day Cluster 1: Global vs Local Model Performance (60 and 90-day outputs)

IO Window	Model	R^2 (Log)	R^2 (Abs)	Log MAE	Residual Skew	MAE
$5 \rightarrow 60$	Global	0.874	0.955	0.559	-1.47	15.46
	Local	0.739	0.175	0.571	-1.08	60.99
$5 \rightarrow 90$	Global	0.831	0.967	0.634	-1.19	15.24
	Local	0.755	0.195	0.592	-1.12	65.53

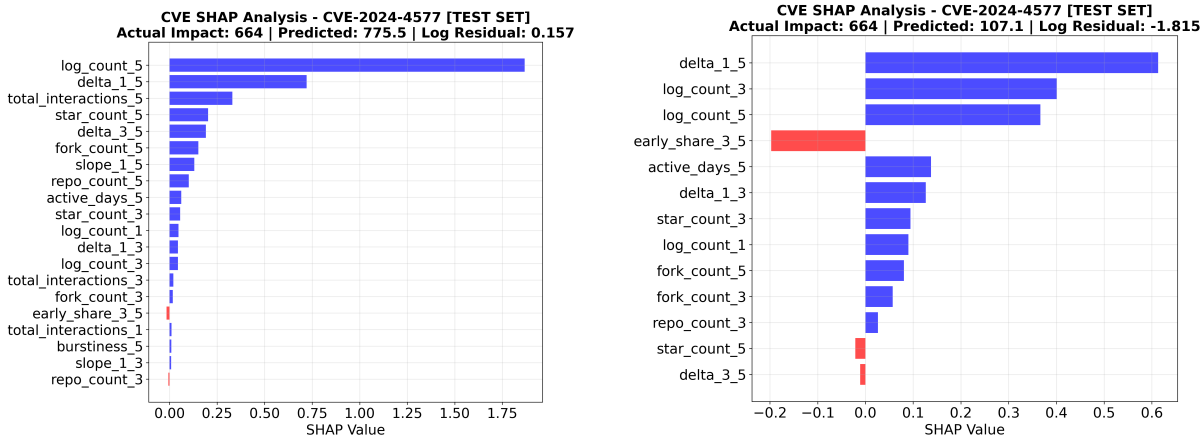
A representative case is 5-day Cluster 1 (oscillators), a small group associated with high R^2 but low SHAP volatility, suggesting superficially predictable behaviour with limited generalisability.

Local retraining on this group leads to a striking divergence: while log R^2 remains moderate (0.739 at 5→60, 0.755 at 5→90), absolute R^2 collapses (0.175 / 0.195), and MAE increases fourfold (from ≈15 to ≈61–66). In other words, the local model locks onto early oscillations but fails to accommodate the heavier tail, thereby generalising poorly.

This limitation becomes especially evident in the case of `CVE-2024-4577`, a high-impact outlier within Cluster 1. The local model underpredicts its 90-day interaction count by over 84%, whereas the global model slightly overestimates, successfully reflecting its overall risk magnitude. (True: 664; Global prediction: 775.5; Local prediction: 107.1).

The SHAP contribution plots in Figure 6.3 offer clear insight into this discrepancy:

- Global model (left): The top features, `log_count_5`, `delta_1_5`, and `total_interactions_5`, reflect overall GitHub activity volume and early growth.



(a) Global model.

(b) Local Cluster 1 model.

Figure 6.3: Feature impact for `CVE-2024-4577` within global and local models.

Because the global model is trained across diverse trend archetypes, these features retain strong discriminative power even for atypical rebound cases.

- Cluster 1 local model (right): The local model, constrained by a narrow and homogeneous training set, relies most heavily on `delta_1_5`, which captures the rebound between Day 1 and Day 5, a defining characteristic of Cluster 1's trend shape. In contrast, features that should matter most for late-rising CVEs, such as total aggregates (`log_count_5`, `total_interactions_5`) and tail-growth signals (`delta_3_5`), receive dampened contributions overall. This imbalance leads the model to undervalue the sustained growth phase, resulting in significantly lower predictions for high-impact, late-activating cases.

Together, these results illustrate the risk of small-cluster overfitting: when raw-count features collapse due to local homogeneity, the model over-relies on shape-based secondary metrics that, while representative of the trend archetype, can mischaracterise high-impact outliers, ultimately reducing both accuracy and robustness, compared to the global model.

Chapter 7

Model Performance on High-Impact CVEs and Real-World Risk Alignment

This section evaluates the model's effectiveness in prioritising CVEs that matter most in practice: those that draw significant ecosystem attention, exhibit signs of in-the-wild exploitation, or represent high systemic risk.

7.1 Ranking Quality Evaluation

Since vulnerability triage is often a matter of prioritisation rather than precision, we evaluate our model not only on absolute prediction accuracy, but also on ranking fidelity, assessing whether the most critical CVEs are correctly surfaced at the top of the queue, and how different temporal trend archetypes perform in this ranking context.

We first use Spearman's rank correlation coefficient (ρ) to measure the ordinal agreement between the model's predicted ranking of CVEs and their actual ranking based on 90-day GHP interaction totals. Using the 5→90 prediction window with the raw+shape feature set on the full test set of 446 CVEs, we obtain a Spearman ρ of 0.9219, indicating a high degree of consistency in the ranking produced by the model.

To further validate this result, we conduct a comparative inspection of the top 15 CVEs by both predicted and observed GHP popularity. The top three CVEs by actual 90-day interaction volume, CVE-2024-3094, CVE-2024-1086, and CVE-2024-6387, are all ranked within the model's top three predictions, albeit in a different order. This alignment suggests that the model is effectively capturing high-priority cases that reflect large-scale ecosystem interest and are indicative of subsequent widespread impact. However, several notable under-predictions remain, most notably CVE-2024-38063, which was significantly underestimated despite ranking among the top-5 most interacted-with CVEs in the dataset. This case is examined in further detail in Section 7.3.

In addition to global ranking, we further assess how well the model surfaces the high-attention CVEs, specifically those behaved as giants (top 10% by total 90-day interactions) or early-bursts (top 10% by day-3 activity). The union of these two groups forms a reference priority set of 61 CVEs. For evaluation, we select the top 15% of the test set (66 CVEs in total), based on their predicted rankings, and compute Precision@K and Recall@K with respect to the high-attention priority set. The model achieves a precision of 0.8182 and a recall of 0.8852. Given that this performance is achieved using only early

GHP signals without access to external contextual metadata, and under the most challenging I/O window configuration (5→90), this performance demonstrates meaningful operational value for early-stage triage.

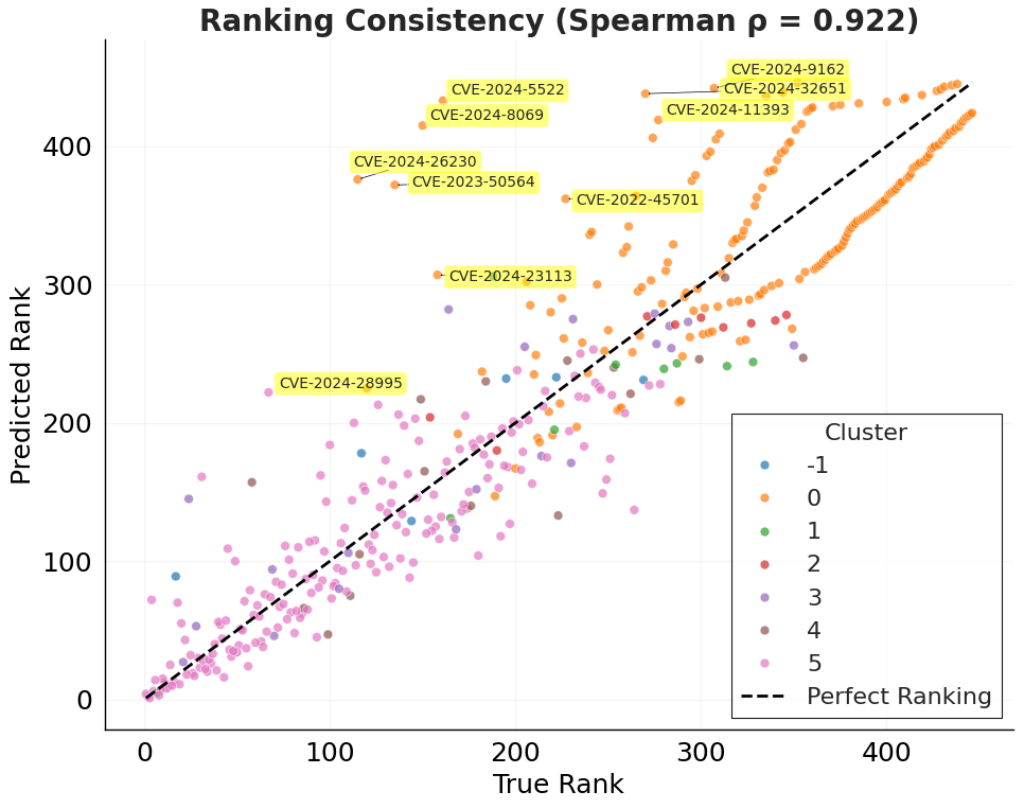
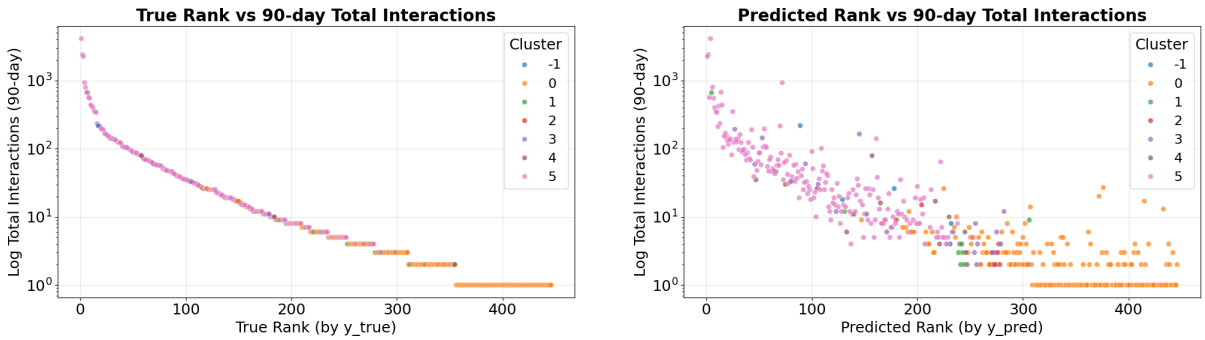


Figure 7.1: Ranking Consistency Between Model and Ground Truth (Spearman $\rho = 0.917$)

Figure 7.1 provides a direct visualisation of ranking consistency across the full test set, based on the 5→90 prediction window and the raw+shape feature set configuration. In the true vs. predicted scatter plot, most points cluster tightly around the diagonal, confirming that the model's predicted ranking closely aligns with the ground truth CVE popularity. Ranking misalignments appear as points that lie far from the diagonal line of perfect agreement. Among the top 10 CVEs with the largest rank discrepancies (labelled by CVE ID in the plot), all are under-predicted cases, with the majority belonging to Cluster 0. This pattern reinforces earlier findings that Cluster 0's cold-start archetype, characterised by weak early signals, introduces greater ambiguity and poses significant challenges for accurate prediction.

Figure 7.2 further contextualises model ranking performance by comparing actual GHP popularity (y-axis) against true and predicted rank (x-axis), with cluster labelling shown via colour coding. In the left panel, the rank-to-popularity curve follows a clear power-law distribution. In the right panel, the model's predicted rankings largely preserve this shape, especially among the most popular CVEs. However, distortions emerge in the lower and mid-rank regions, most notably among Cluster 0 CVEs.



(a) True rank vs actual 90-day GHP count (log scale).

(b) Predicted rank vs actual 90-day GHP count.

Figure 7.2: Comparison of CVE popularity rankings based on GitHub PoC (GHP) activity.

Together, these plots highlight both the model's robust prioritisation at the top end and the systematic errors that emerge across different temporal trend types, reinforcing the need for the cluster-specific enhancements introduced in Section 6.2.

7.2 Comparison with External Exploitation Signals

In this section, we link our dataset to other exploit signal sources to evaluate two key questions: (1) whether GHP activity provides a lead time relative to downstream exploitation events, and (2) how the model's predictions align with confirmed in-the-wild exploitation, as captured by external sources such as KEV, Metasploit, and ransomware datasets.

7.2.1 Lead Time of GHP Signals

To assess whether our model predictions provides genuine foresight rather than a reflection of already-established risk, we compare the appearance time of downstream exploitation signals, specifically ExploitDB, Metasploit, and KEV, to the model's input window cutoff (Day 5 in the 5–90 configuration). For each CVE with a valid signal timestamp, we identify the first observed occurrence of each exploitation source and compare it to the sixth calendar day following the CVE's GHP activation date.

We include ExploitDB (EDB) in our analysis as a commonly used repository of exploit code, often referenced in vulnerability triage studies. However, it is worth noting that like GHP, EDB is more accurately characterised as a proof-of-concept aggregator, reflecting researcher publication activity rather than confirmed exploitation in the wild. Accordingly, we treat EDB as a comparative timing signal, useful for mapping the chronology of exploit code emergence, but different from real-world risk indicators such as Metasploit (tooling integration) and KEV (confirmed exploitation). As such, in the following evaluation (Section 7.2.2), we exclude EDB from the in-the-wild exploitation scoring framework to ensure conceptual clarity and avoid conflating different classes of signals.

For CVEs that eventually received each exploitation signal, we calculate the proportion for which the signal appeared after the model's input window cutoff. Figure 7.3 (left) presents these percentages for both the full dataset (covering CVEs from January 2018 to March 2025) and a focused subset of the 61 high-attention CVEs in the test set.

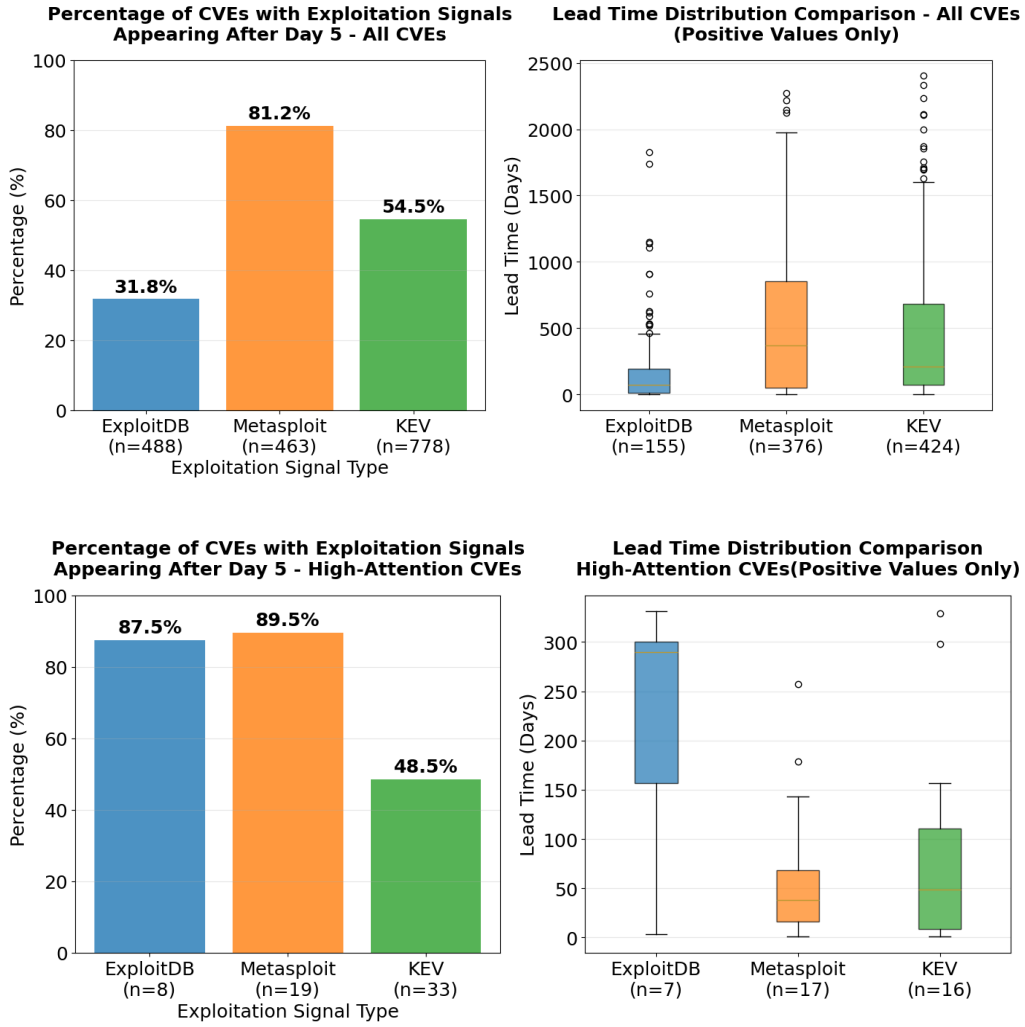


Figure 7.3: Comparison of exploitation signal timing relative to GHP activation. **Top:** All CVEs. **Bottom:** 61 High-attention CVEs.

Across the full dataset, we observe that 81.2% of Metasploit entries and 54.5% of KEV inclusions occur after Day 5, indicating that these signals are not yet visible to the model at prediction time. ExploitDB, by contrast, appears earlier, with only 31.8% of entries emerging after the cutoff. This consistent with its similar role as a PoC aggregator. Among the 61 high-attention CVEs in test set, this trend becomes even more pronounced. Nearly 89.5% of Metasploit and 87.5% of ExploitDB signals occur after Day 5. Even for KEV, which often benefits from pre-coordinated disclosure, vendor reporting, or retroactive tagging – nearly half of the entries (48.5%) emerge post-input window.

The box plots further illustrate the temporal lag between GHP activation and the appearance of downstream exploitation signals. In the full dataset, the median lead time from GHP to Metasploit is 370.5 days, while KEV trails by a median of 210.0 days. ExploitDB exhibits a much shorter median lead time of 74.0 days, again highlighting its alignment with early publication cycles rather than downstream risk validation.

For the high-attention CVE subset, we observe a compressed but still distinct delay across all exploitation signal types. Median lead times remain substantial: 290.0 days for ExploitDB, 38.0 days for Metasploit, and 48.5 days for KEV. Notably, given that our test set consists exclusively of CVEs published in 2024, the long delay associated with ExploitDB may reflect a broader ecosystem shift, wherein GitHub has increasingly

become the primary platform for PoC code disclosure, displacing EDB's historical role as a first-stop publication hub.

The consistency of these lead times reinforces the conclusion that GHP trends provide a meaningful predictive advantage. They enable the identification of high-attention CVEs well before attacker tooling, curated advisories, or institutional databases register the risk, offering an early look at which vulnerabilities are likely to enter the exploitation pipeline. This pattern holds especially true in the post-2024 ecosystem, where traditional signalling mechanisms lagged.

7.2.2 Alignment with Real-World Exploitation

To further evaluate our model's prediction for real-world exploitation, we introduce a Exploitation Signal Score based on confirmed in-the-wild exploitation signals, aggregating presence in Metasploit exploitation modules, VulnCheck KEVs and ransomware leak reports (from [1] and [2]). This scoring system helps surface CVEs with varying levels of real-world risk, based on observed offensive activity.

The Vulners KEV database is an open-source resource that provides transparent, timestamped evidence of in-the-wild exploitation. In this study, we chose VulCheck KEV over CISA's canonical list for two primary reasons: broader coverage and faster signal availability. A cross-validation conducted on our dataset confirmed that VulnCheck includes all CVEs present in CISA's KEV, with matching or earlier exploitation timestamps for every entry. This further supports the use of VulnCheck as not only a superset of the CISA KEV list, but also as a more representative real-world signal source for linking our test set to in-the-wild exploitations.

The composite Exploitation Signal Score reflect observed in-the-wild exploitation signals. For each CVE, the score is computed as follows:

```
exploitation_signal_score = (3 × ransomware_association)
                          + (2 × KEV_presence)
                          + (1 × metasploit_exploit_module_presence)

Where:

KEV_presence = 1
      if the CVE appears in the Vulners KEV database, else 0
ransomware_association = 1
      if the CVE is linked to known ransomware campaigns, else 0
metasploit_module_presence = 1
      if a Metasploit exploit module exists for the CVE, else 0
```

The resulting Exploitation Signal Score is then mapped to a qualitative risk label using the following rule set:

```
if exploitation_signal_score   >= 5:   risk_label = "ultra-high"
elif exploitation_signal_score >= 3:   risk_label = "high"
elif exploitation_signal_score >= 2:   risk_label = "medium"
```

[1] https://github.com/BushidoUK/Ransomware-Vulnerability-Matrix.git

[2] https://blog.qualys.com/vulnerabilities-threat-research/2025/05/08/inside-lockbit-defense-lessons-from-the-leaked-lockbit-negotiations

```
elif exploitation_signal_score >= 1:    risk_label = "low"
else:                                   risk_label = "none"
```

Figure 7.4 presents the cluster-wise breakdown of real-world risk levels within the test set. As expected, Cluster 5, characterised by front-loaded burst dynamics, dominates in high and ultra-high risk CVEs. Notably, Cluster 0, which largely comprises cold-start and low-to-moderate interaction CVEs, also contains several high-risk cases (3 high, 34 medium). This reinforces the importance of cluster-specific enhancements aimed at recovering underpredicted but potentially critical CVEs. In contrast, Clusters 1 ,3 and 4 show minimal presence, mostly due to small sample sizes. Cluster 2 displays no overlap with in-the-wild exploitation signals, further confirming its association with low-risk, early-tapering trajectories.

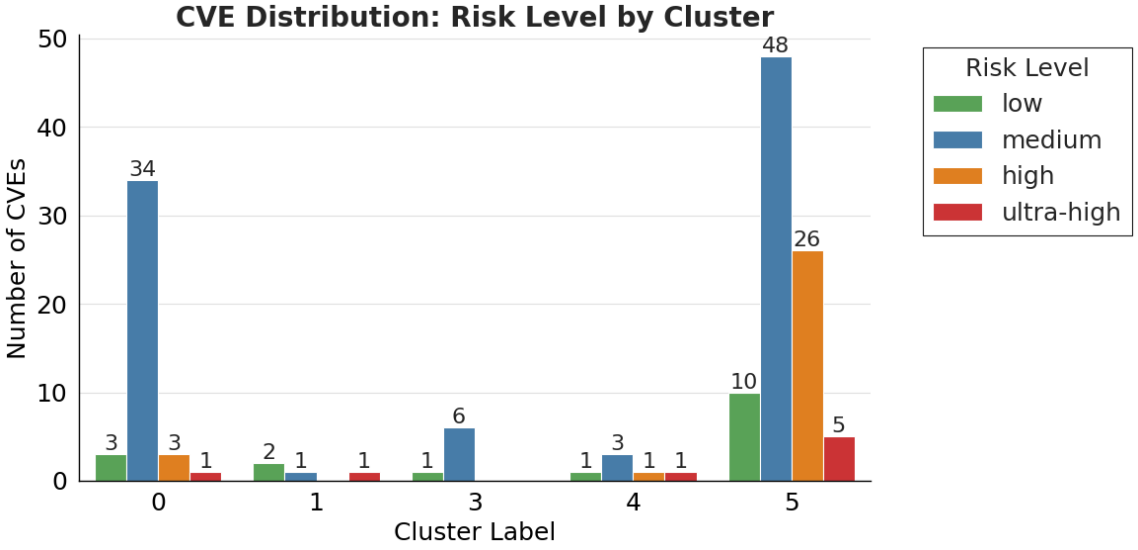


Figure 7.4: Risk labels derived from the Exploitation Signal Score, visualised by trend cluster (excluding noise cluster –1).

Furthermore, we evaluate how well the model predicts the actual eventual interaction volume of the 61 high-attention CVEs in the test set, in relation to their real-world threat relevance.

Figure 7.5 presents a scatter plot comparing predicted versus actual log-transformed interaction counts, with each point coloured according to its corresponding exploitation signal tier. Most high-attention CVEs, including those with confirmed real-world exploitation, fall close to the diagonal or within the ± 0.5 log error band. Notably, all ultra-high risk CVEs, and 8 out of 11 high-risk CVEs, lie within this margin. These results suggest that the model captures more than just surface-level popularity: GHP activity serves as a meaningful proxy for latent signals of emerging threat relevance, and the model's predictions reflect substantive real-world risk.

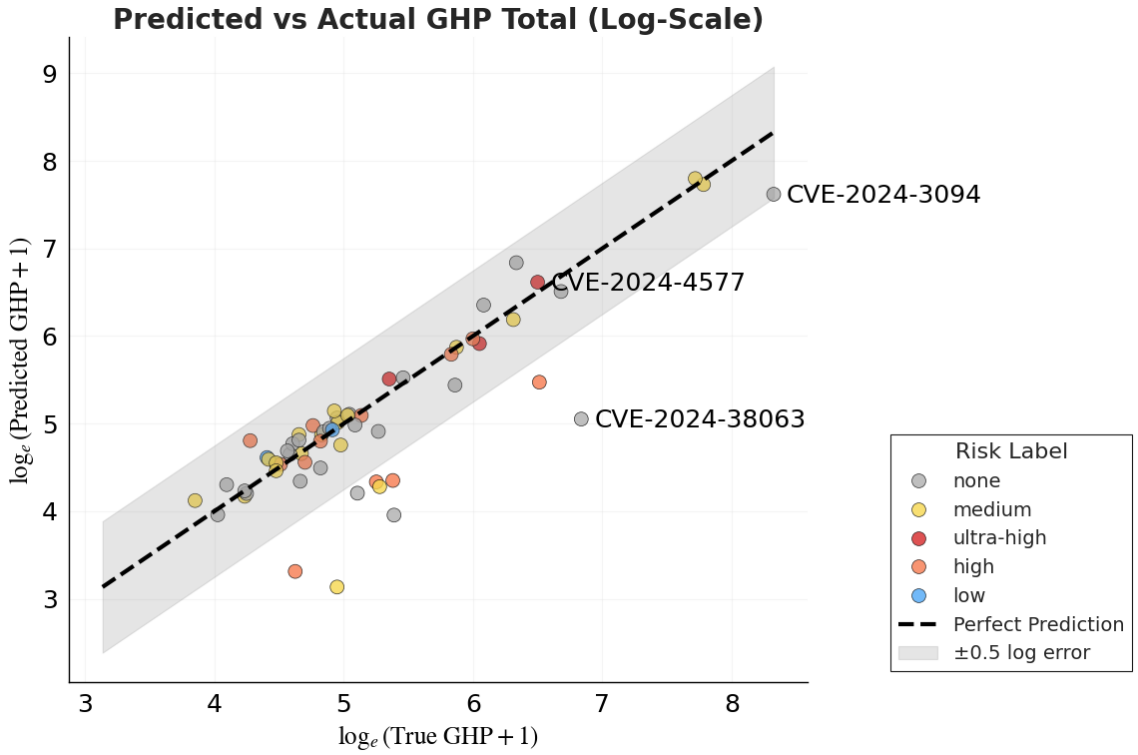


Figure 7.5: Prediction Accuracy for High-Exploitation-Signal CVEs

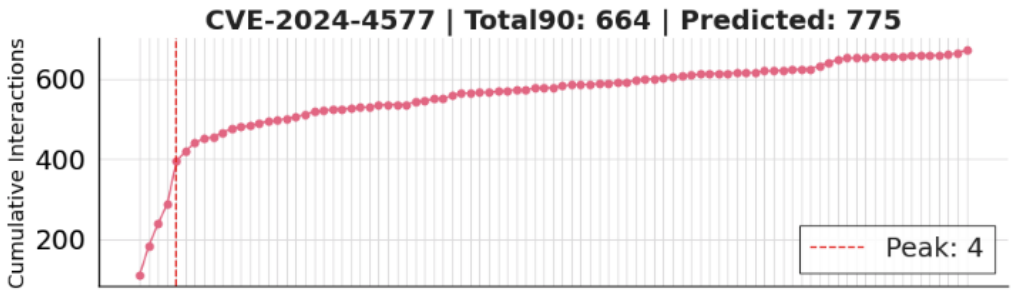
7.3 High-Attention CVEs - Case Study

In this subsection, we examine three representative CVEs drawn from the 61 high-attention cases in our test set: CVE-2024-4577, CVE-2024-38063 and CVE-2024-3094. Each illustrates a distinct prediction scenario: a well-predicted CVE with ultra-high exploitation signals, a severely underpredicted case, and a case of anticipatory attention: lacking confirmed exploitation, yet triggering an exceptional community response. Together, these examples highlight the model's strengths, expose its current limitations, and demonstrate its potential to surface latent ecosystem risk.

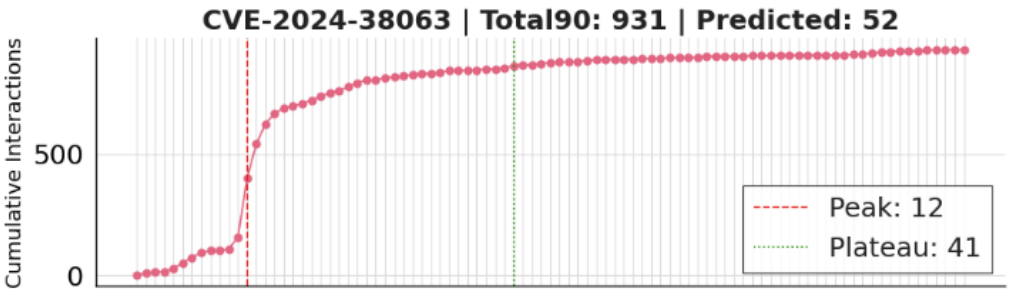
CVE-2024-4577

Among all CVEs with ultra-high exploitation signals: CVE-2024-4577 is the one with the highest 90 day total GHP attention. CVE-2024-4577 is a remote argument injection vulnerability in PHP for Windows, affecting versions up to 8.1.28 / 8.2.19 / 8.3.7. Exploitation relies on Windows "Best-Fit" character substitution, which can allow attackers to pass unintended command-line options to the PHP-CGI binary, leading to arbitrary code execution or source code disclosure. The vulnerability was officially listed in KEV in June 2024, has a Metasploit module, and is linked to ransomware activity. This vulnerability belongs to the oscillator trend archetype, landing in 5-day Cluster 1 and 7-day Cluster 3.

Figure 7.6 (top) shows the cumulative interaction timelines of `CVE-2024-4577`. Despite the lacking of a clear plateau in the 90 day forecast horizon, it is accurately predicted by the model, received a strong and accurate prediction (775 vs. 664 actual). This accurate output can be attributed to `CVE-2024-4577`'s immediate and sustained early



(a) Cumulative interaction timeline for `CVE-2024-4577`.



(b) Cumulative interaction timeline for `CVE-2024-38063`.

Figure 7.6: Cumulative GitHub PoC interaction timelines for two high-profile CVEs.

activity: the CVE attracted significant attention on Day 1, followed by steady growth and a resurgence that peaked around Day 5, well within the model's early observation window. It then continued to accumulate interactions throughout the full 90-day period. The model's feature space effectively captures this early burst profile, enabling it to assign high predictive confidence despite continued strong and sustained growth over the full 90-day window.

CVE-2024-38063

On the other hand, CVE-2024-38063, a Windows TCP/IP Remote Code Execution vulnerability, reached 927 interactions within 90 days, yet was the most severely underpredicted by the global model (predicted: 52; log residual: –1.815). The performance is clearly shown and annotated in Figure 7.5. In Figure 7.6 (bottom), the interaction trajectory of CVE-2024-38063 helps explain the model's underestimation: although early indicators remained flat, the CVE experienced a sharp rise in attention starting around Day 10, peaking at Day 12, and plateauing by Day 41. This delayed burst pattern fell just outside the model's input window, causing early features to miss the signal entirely.

As methods for addressing this limitation, the Cluster 5 Q95 regression as detailed in Section 6.2.2 demonstrate marked improvement: the predicted value increased to 115.1, representing a 106% improvement over the raw+shape baseline.

CVEs like CVE-2024-38063, which exhibit delayed but intense engagement, challenge the base model that rely solely on early raw interaction features. By using enhanced regression strategies, shifting the predictive target to better accommodate high-attention CVEs, the model is proved to provide more accurate estimates for the cases that demand urgent prioritisation.

CVE-2024-3094

Finally, CVE-2024-3094 presents a distinct case: it received the highest 90-day GHP attention in our test set, yet has no confirmed in-the-wild exploitation. The activation date of this GHP is the same as the CVE's publication date in the NVD, indicating immediate community mobilisation upon disclosure. This vulnerability is a supply-chain backdoor inserted into `xz Utils`, a widely used Linux compression library, and was designed to intercept SSH authentication during build time via obfuscated tarballs. Although it was never deployed in production versions of major distributions, it caused widespread alarm due to its sophistication, stealth, and potential for catastrophic compromise.

In the prediction accuracy scatter plot under the 7–90 day configuration (Figure 7.5), CVE-2024-3094 appears annotated in the top-right corner, falling within the ± 0.5 log error band. It also ranks as the third-highest predicted CVE, reflecting both its prominence and the model's strong alignment with observed GHP activity. This case illustrates GHP-based signal's ability to surface vulnerabilities attracting urgent and widespread attention at early stages, anticipating systematic urgency and ecosystem-level risk before confirmed exploitation or public tooling becomes available.

Chapter 8

Discussion and Limitations

8.1 Discussion

8.1.1 GHP Dynamics as Early Risk Signals

Over the years, GitHub have increasingly become the primary platform for PoC code disclosure, and GHP repositories have grown not only in number but also in responsiveness and social visibility over time. In aggregate, GHP attention exhibits a front-loaded and long-tailed nature: for a majority of cases, the bulk of interaction occur within the first few days of ecosystem exposure, followed by a slower and more diffuse residual phase. Yet a subset of GHP do sustain or regain user engagement over time, often a distinguishing feature of high-risk CVEs that warrant early prioritisation in vulnerability triage workflows.

For our model using XGBoost, a short early observation window, such as 5 or 7 days of GHP activity, is sufficient to support robust long-range forecasts. Using only features derived from user interaction patterns (raw count and temporal shape), the global model achieves strong predictive performance across 90-day horizons. Spearman rank correlations exceed 0.91, and the model demonstrates high precision and recall in identifying high-attention vulnerabilities, those that exhibit intense early burst of engagement or accumulate extensive traction over time.

Early GHP trends encode distinct attention archetypes, with trend showing cold starts, early plateaus, oscillators or steady tapers. These archetypes are not only interpretable but also operationally relevant: they align with plateau timing, cluster-specific prediction performance, and downstream exploitation signals. Notably, high-impact vulnerabilities that are later weaponised or linked to ransomware campaigns, often attract sustained attention within the first few days of GHP publication. In such cases, early GHP activity provides a latent signal of emerging risk, often surfacing well before formal evidence of exploitation becomes available.

Accordingly, we argue that GHP engagement reflects more than surface-level popularity. It serves as an emergent triage signal that complements static scores like CVSS, and fills the gaps left by delayed scoring or incomplete metadata. In particular, GitHub trends are capable of surfacing both technically severe CVEs and those that capture community attention due to tooling potential, stealth characteristics, or symbolic urgency (e.g., supply chain backdoors).

8.1.2 Prediction Challenges and Modelling Trade-offs

Prediction fidelity is not uniform across all CVE types. When early GHP trend exhibit strong user engagement and match clear behavioural archetypes, the model tend to deliver accurate prediction result. However, cold-start CVEs with flat or weak early activity, and late-breaking attention well beyond the observation window, remain the greatest challenge for early-stage forcasting. For these CVEs, improved accuracy may require cluster-specific extension of the observation window (e.g., to 30 days) or the inclusion of external contextual features such as vulnerability metadata.

On the other hand, cluster-wise quantile regression, particularly at the 95th percentile, improves the model's ability to account for upper-bound risk, offering more conservative predictions when early signals are ambiguous. This is especially valuable in triage contexts, where systematic under-prediction is more harmful than cautious overestimation.

For smaller trend groups, however, cluster-wise modelling shows diminishing returns. In such cases, the global model retains better generalisability, benefiting from shared feature variance across the dataset. While cluster membership may still serve as a valuable interpretive lens, training independent regressors per cluster introduces risk of overfitting, particularly in settings with limited data availability.

8.2 Limitations

Several limitations constrain the scope and generalisability of this study:

Coverage Bias. The dataset is restricted to CVEs with publicly available PoCs on GitHub—a small and non-random subset of the total CVE population. This introduces selection bias toward vulnerabilities that attract research or community interest. Many real-world exploited CVEs lack a public PoC altogether, or circulate privately in closed groups and underground forums.

API Constraints. Due to GitHub API limitations, watcher counts could not be reliably retrieved, and were therefore excluded from the user interaction dataset. While stars and forks are generally stronger indicators of active interest, the absence of watcher data slightly reduces feature dimensionality and may obscure weaker signals of passive engagement.

Activation Threshold Sensitivity. The operational definition of "activation" is based on a fixed user interaction threshold (e.g., $k = 3$ or 5). Varying this threshold alters the assigned activation date and the resulting CVE cohort, potentially affecting both model training and evaluation outcomes. Although $k = 3$ and 5 were selected empirically to balance early signal against noise, this design choice introduces latent sensitivity into the modelling pipeline.

One-to-One PoC–CVE Mapping. To preserve interpretability, the dataset excludes multi-CVE repositories from prediction tasks. This design choice avoids conflated activation timelines and ambiguous mappings, but also omits attacker-relevant artefacts such as exploit frameworks and vulnerability chains—both of which are common in real-world exploitation workflows and could impact the model's relevance in practice.

External Validity and Temporal Scope. This study focuses on GitHub-hosted PoCs published between 2018 and early 2025. Shifts in disclosure practices, platform usage, or attacker behaviour could reduce the generalisability of the findings. For operational deployment, models would require ongoing retraining to remain effective under evolving ecosystem conditions.

Temporal Split and Variance. While the model adopts a time-aware train–test split to preserve causal ordering and reflect deployment conditions, future work could incorporate multiple temporal splits or cross-validation within constrained windows. This would reduce sensitivity to test-set artefacts and improve generalisability across time.

Metadata Noise and Platform Manipulation. Repository creation timestamps (`repo_created_at`) do not reliably indicate the true GHP publication date. Some repositories may accumulate engagement for unrelated purposes before being repurposed as PoC hosts, introducing temporal ambiguity. To mitigate this, we filter out repositories whose creation date precedes the associated CVE's publication by more than 270 days and anchor activation to the first observed user interaction. However, this approach does not fully resolve timing uncertainties. Additionally, platform-level manipulation, such as the emergence of fake-star campaigns in 2024 ([He et al., 2024]), further complicates engagement signals. While these campaigns primarily targeted phishing or malware repositories, not legitimate PoCs, their presence during the 2024 evaluation period introduces potential noise in the test set and may confound time-sensitive popularity modelling.

Chapter 9

Conclusion and Future Work

In this study, we explored an alternative approach to vulnerability triage that bypasses centralised scoring systems like NVD, in favour of real-time, decentralised signals. GitHub PoC activity reflects behavioural engagement from a diverse set of actors: researchers, defenders, opportunistic observers, and potential attackers. These signals capture early interest, collaborative response, and the construction of exploit pathways, all emerging independently of formal metadata pipelines. As such, they offer an actor-diverse and time-coded proxy for assessing vulnerability risk in its earliest stages. This thesis demonstrates that early GHP attention is not a downstream echo of CVSS, but a stand-alone indicator, capable of forecasting risk trajectories even before structured metadata becomes available.

Specifically, using a time-aligned GHP user interaction dataset, we defined a forecasting framework that models early GHP attention as a temporal prediction task, and uncover behavioural trend archetypes through unsupervised clustering of early engagement trajectories. We then evaluate both global and cluster-specific models across these trend types, identifying cases where early signals are learnable, and where extended input windows or enhanced regressions are required to overcome model limitations. Finally, we assess the model's ability to surface high-risk CVEs in real-world scenarios, showing the alignment between early GHP-driven predictions and confirmed exploitation signals such as KEV inclusion, Metasploit modules, and ransomware records.

Together, these results demonstrate the viability of GHP activity as a forecastable early signal of vulnerability risk, and highlight the importance of temporal modelling and decentralised signal sources in building more responsive triage systems.

One approach for future extensions is contextual enrichment. A promising direction lies in extracting further coarse-grained signals from GitHub repository metadata, such as declared programming languages, topic tags, or README content, to approximate vulnerability type or technological domain. These features can be observed in real-time and align with the decentralised, behaviour-first ethos of this work.

Structured fields such as CPE (Common Platform Enumeration) and CWE (Common Weakness Enumeration) may also offer auxiliary value, particularly for cold-start CVEs or delayed-activation cases. However, to preserve the model's temporal integrity, such features should be included only if they are available at or before the end of the GHP activation observation window. This constraint would ensure compatibility with early-stage triage, where enrichment delays are common. In all cases, these contextual features should be treated as optional complements, not prerequisites.

Another possible extension of this work is to explore sequence-to-sequence architectures, such as LSTMs or transformer-based models, to better capture temporal depen-

dencies and fine-grained progression in GHP engagement trends. Given the structured, timestamped nature of GHP interaction data, these models may uncover latent dynamics that are less accessible to tree-based regressors. However, the relatively modest dataset size and strong skew in interaction patterns pose challenges for effective training. Simple oversampling is unlikely to address this imbalance. As a more promising avenue, future work could explore generative approaches, for instance, using GANs to simulate realistic GHP interaction trajectories. These synthetic series could serve both as data augmentation and as a tool for stress-testing model generalisability under rare or extreme attention scenarios.

Bibliography

Luca Allodi and Fabio Massacci. Comparing Vulnerability Severity and Exploits Using Case-Control Studies. *ACM Transactions on Information and System Security*, 17(1): 1–20, August 2014. ISSN 1094-9224, 1557-7406. doi: 10.1145/2630069.

Tegawendé F. Bissyandé, Ferdian Thung, David Lo, Lingxiao Jiang, and Laurent Réveillère. Popularity, Interoperability, and Impact of Programming Languages in 100,000 Open Source Projects. In *2013 IEEE 37th Annual Computer Software and Applications Conference*, pages 303–312, July 2013. doi: 10.1109/COMPSAC.2013.55.

Hudson Borges and Marco Tulio Valente. What's in a GitHub Star? Understanding Repository Starring Practices in a Social Coding Platform. *Journal of Systems and Software*, 146:112–129, December 2018. ISSN 01641212. doi: 10.1016/j.jss.2018.09.016.

Hudson Borges, Andre Hora, and Marco Tulio Valente. Understanding the Factors That Impact the Popularity of GitHub Repositories. *2016 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pages 334–344, October 2016. doi: 10.1109/ICSME.2016.31.

Haipeng Chen, Rui Liu, Noseong Park, and V.S. Subrahmanian. Using Twitter to Predict When Vulnerabilities will be Exploited. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3143–3152, Anchorage AK USA, July 2019. ACM. doi: 10.1145/3292500.3330742.

Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*, pages 785–794. ACM, August 2016. doi: 10.1145/2939672.2939785.

Thiago Figueiredo Costa and Mateus Tymburibá. Challenges on prioritizing software patching. In *2022 15th International Conference on Security of Information and Networks (SIN)*, pages 1–8, November 2022. doi: 10.1109/SIN56466.2022.9970537.

Daniel Alves de Sousa, Elaine Ribeiro de Faria, and Rodrigo Sanches Miani. Evaluating the performance of twitter-based exploit detectors. In *arXiv preprint arXiv:2011.03113*, November 2020. URL `https://arxiv.org/abs/2011.03113`.

Yutong Du, Cheng Huang, Genpei Liang, Zhihao Fu, Dunhan Li, and Yong Ding. ExpSeeker: Extract public exploit code information from social media. *Applied Intelligence*, 53(12):15772–15786, June 2023. ISSN 1573-7497. doi: 10.1007/s10489-022-04178-9.

Hao He, Haoqin Yang, Philipp Burckhardt, Alexandros Kapravelos, Bogdan Vasilescu, and Christian Kästner. 4.5 million (suspected) fake stars in github: A growing spiral of popularity contests, scams, and malware. In *arXiv preprint arXiv:2412.13459*, December 2024. URL https://arxiv.org/abs/2412.13459.

Jay Jacobs, Sasha Romanosky, Benjamin Edwards, Idris Adjerid, and Michael Roytman. Exploit Prediction Scoring System (EPSS). *Digital Threats*, 2(3):20:1–20:17, July 2021. doi: 10.1145/3436242.

Yuning Jiang, Nay Oo, Qiaoran Meng, Hoon Wei Lim, and Biplab Sikdar. A survey on vulnerability prioritization: Taxonomy, metrics, and research challenges. In *arXiv preprint arXiv:2502.11070*, February 2025. URL https://arxiv.org/abs/2502.11070.

Kobra Khanmohammadi, Zakeya Namrud, François Labrèche, and Raphaël Khoury. ExploitabilityBirthMark: An Early Predictor of the Likelihood of Exploitation. In Kamel Adi, Simon Bourdeau, Christel Durand, Valérie Viet Triem Tong, Alina Dulipovici, Yvon Kermarrec, and Joaquin Garcia-Alfaro, editors, *Foundations and Practice of Security*, pages 187–199, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-87496-3. doi: 10.1007/978-3-031-87496-3_13.

Kentaro Kita, Yuta Gempei, Tomoaki Mimoto, Takamasa Isohara, Shinsaku Kiyomoto, and Toshiaki Tanaka. Prioritization of Exploit Codes on GitHub for Better Vulnerability Triage:. In *Proceedings of the 11th International Conference on Information Systems Security and Privacy*, pages 27–38, Porto, Portugal, 2025. SCITEPRESS - Science and Technology Publications. ISBN 978-989-758-735-1. doi: 10.5220/0013100800003899.

Triet H. M. Le, Huaming Chen, and M. Ali Babar. A Survey on Data-driven Software Vulnerability Assessment and Prioritization. *ACM Computing Surveys*, 55(5):1–39, May 2023. ISSN 0360-0300, 1557-7341. doi: 10.1145/3529757.

Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS)*, pages 4765–4774. Curran Associates, Inc., November 2017. doi: 10.48550/arXiv.1705.07874. URL https://arxiv.org/abs/1705.07874.

Claudia Malzer and Marcus Baum. A Hybrid Approach To Hierarchical Density-based Cluster Selection. In *2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 223–228, September 2020. doi: 10.1109/MFI49285.2020.9235263.

Deep Parekh. Exploring popularity and usage: A comparative analysis of github stars and pypi downloads in python libraries. ResearchGate, 2024. URL https://www.researchgate.net/publication/380728909.

Carl Sabottke, Octavian Suciu, and Tudor Dumitras. Vulnerability disclosure in the age of social media: Exploiting twitter for predicting Real-World exploits. In *24th USENIX Security Symposium (USENIX Security 15)*, pages 1041–1056, Washington, D.C., August 2015. USENIX Association. ISBN 978-1-939133-11-3. URL https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/sabottke.

Madeline Schiappa, Graham Chantry, and Ivan Garibay. Cyber Security in a Complex Community: A Social Media Analysis on Common Vulnerabilities and Exposures. In *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 13–20, October 2019. doi: 10.1109/SNAMS.2019.8931883.

Ensar Seker and Weizhi Meng. XVRS: Extended Vulnerability Risk Scoring based on Threat Intelligence. In *2023 IEEE International Conference on Metaverse Computing, Networking and Applications (MetaCom)*, pages 516–523, June 2023. doi: 10.1109/MetaCom57706.2023.00094.

Prasha Shrestha, Arun Sathanur, Suraj Maharjan, Emily Saldanha, Dustin Arendt, and Svitlana Volkova. Multiple social platforms reveal actionable signals for software vulnerability awareness: A study of GitHub, Twitter and Reddit. *PLOS ONE*, 15(3): e0230250, March 2020. ISSN 1932-6203. doi: 10.1371/journal.pone.0230250.

Octavian Suciu, Connor Nelson, Zhuoer Lyu, Tiffany Bao, and Tudor Dumitras. Expected exploitability: Predicting the development of functional vulnerability exploits. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 377–394, Boston, MA, August 2022. USENIX Association. ISBN 978-1-939133-31-1. URL `https://www.usenix.org/conference/usenixsecurity22/presentation/suciu`.

Soufian El Yadmani, Robin The, and Olga Gadyatskaya. Beyond the surface: Investigating malicious CVE proof of concept exploits on GitHub. In *arXiv preprint arXiv:2210.08374*, October 2022. URL `https://arxiv.org/abs/2210.08374`.

Awad A. Younis and Yashwant K. Malaiya. Comparing and Evaluating CVSS Base Metrics and Microsoft Rating System. In *2015 IEEE International Conference on Software Quality, Reliability and Security*, pages 252–261, August 2015. doi: 10.1109/QRS.2015.44.

Fengli Zhang and Qinghua Li. Dynamic Risk-Aware Patch Scheduling. In *2020 IEEE Conference on Communications and Network Security (CNS)*, pages 1–9, June 2020. doi: 10.1109/CNS48642.2020.9162225.

Appendix

List of Abbreviations

Abbreviation	Definition
CVE	Common Vulnerabilities and Exposures
PoC	Proof of Concept
GHP	GitHub-hosted PoC (repository linked to a CVE)
IQR	Interquartile Range
TH3	Activation threshold = 3 interactions
TH5	Activation threshold = 5 interactions
MAE	Mean Absolute Error
R^2	Coefficient of Determination
UMAP	Uniform Manifold Approximation and Projection
SHAP	SHapley Additive exPlanations
CVSS	Common Vulnerability Scoring System
EPSS	Exploit Prediction Scoring System
NVD	National Vulnerability Database
KEV	Known Exploited Vulnerability
EDB	ExploitDB
GAN	Generative Adversarial Network

Hyperparameter Tuning Configuration

Table 1: Curated hyperparameter configurations used in the lightweight tuning sweep (21 total).

Group	Max Depth	Learning Rate	Estimators	Subsample / Colsample	Reg. (α, λ)
Depth–Rate	3	0.10	300	1.0 / 1.0	0.0, 1.0
	6	0.10	300	1.0 / 1.0	0.0, 1.0
	9	0.10	300	1.0 / 1.0	0.0, 1.0
	6	0.05	500	1.0 / 1.0	0.0, 1.0
	6	0.10	1000	1.0 / 1.0	0.0, 1.0
	6	0.01	1000	1.0 / 1.0	0.0, 1.0
	3	0.05	500	1.0 / 1.0	0.0, 1.0
Regularisation	6	0.10	300	1.0 / 1.0	0.0, 1.0
	6	0.10	300	1.0 / 1.0	0.1, 1.5
	6	0.10	300	1.0 / 1.0	0.5, 2.0
Subsampling	6	0.10	300	1.0 / 1.0	0.0, 1.0
	6	0.10	300	0.8 / 0.8	0.0, 1.0
	6	0.10	300	0.7 / 0.7	0.0, 1.0
Extremes	2	0.03	800	1.0 / 1.0	0.0, 1.0
	4	0.05	500	1.0 / 1.0	10.0, 10.0
	4	0.05	500	0.5 / 0.5	0.0, 1.0
Gap Fillers	9	0.01	1000	1.0 / 1.0	0.0, 1.0
	2	0.10	300	1.0 / 1.0	0.0, 1.0
	3	0.01	1500	1.0 / 1.0	0.0, 1.0
	6	0.03	1000	1.0 / 1.0	10.0, 10.0
	4	0.03	500	0.6 / 0.6	5.0, 5.0

Hyperparameter Tuning Results (Train Set)

Table 2: Full train-set evaluation results for 21 hyperparameter configurations (5→90 input–output window, activation threshold = 5). All models trained on CVEs disclosed Jan 2018–Sep 2023; validated on Oct 2023–Mar 2024 (n=182).

Model	R^2(Log)	R^2(Abs)	Log MAE	MAE	RMSE	Skew	Notes
cfg_md3_lr0.1_ne300	0.897	0.871	0.398	18.67	51.47	–1.45	depth=3, lr=0.1, ne=300
cfg_md6_lr0.1_ne300	0.867	0.884	0.432	19.14	48.79	–0.48	depth=6, lr=0.1, ne=300
cfg_md9_lr0.1_ne300	0.863	0.875	0.442	20.20	50.54	–0.29	depth=9, lr=0.1, ne=300
cfg_md6_lr0.05_ne500	0.870	0.865	0.433	20.27	52.58	–0.58	depth=6, lr=0.05, ne=500
cfg_md6_lr0.1_ne1000	0.857	0.883	0.448	19.66	49.03	–0.26	default baseline
cfg_md6_lr0.01_ne1000	0.890	0.876	0.411	18.95	50.40	–1.15	conservative learner
cfg_md3_lr0.05_ne500	0.900	0.867	0.395	19.13	52.24	–1.56	medium depth
cfg_md6_lr0.1_ne300_ra0.0_rl1.0	0.867	0.884	0.432	19.14	48.79	–0.48	no reg
cfg_md6_lr0.1_ne300_ra0.1_rl1.5	0.870	0.873	0.434	19.51	50.90	–0.55	light reg
cfg_md6_lr0.1_ne300_ra0.5_rl2.0	0.875	0.878	0.424	19.46	49.89	–0.69	stronger reg
cfg_md6_lr0.1_ne300_ss1.0_cs1.0	0.871	0.878	0.431	20.05	49.94	–0.46	no subsampling
cfg_md6_lr0.1_ne300_ss0.8_cs0.8	0.867	0.884	0.432	19.14	48.79	–0.48	subsample 0.8
cfg_md6_lr0.1_ne300_ss0.7_cs0.7	0.867	0.850	0.435	20.43	55.37	–0.49	subsample 0.7
cfg_md2_lr0.03_ne800	0.908	0.857	0.378	19.28	54.18	–1.81	shallow, slow learner
cfg_md4_lr0.05_ne500_ra10_rl10	0.909	0.866	0.375	18.74	52.37	–2.01	heavy reg
cfg_md4_lr0.05_ne500_ss0.5_cs0.5	0.891	0.889	0.409	18.50	47.76	–1.20	subsample 0.5
cfg_md9_lr0.01_ne1000	0.872	0.873	0.430	19.85	51.06	–0.51	deep, slow learner
cfg_md2_lr0.1_ne300	0.905	0.866	0.384	18.96	52.44	–1.72	shallow, aggressive
cfg_md3_lr0.01_ne1500	0.907	0.876	0.381	18.39	50.38	–1.72	patience config
cfg_md6_lr0.03_ne1000_ra10_rl10	0.908	0.873	0.377	18.66	50.99	–1.98	reg + deeper learner
cfg_md4_lr0.03_ne500_ss0.6_ra5_rl5	0.906	0.887	0.382	18.08	48.15	–1.94	reg + subsample combo

Cluster-Wise Evaluation Metrics Heatmap

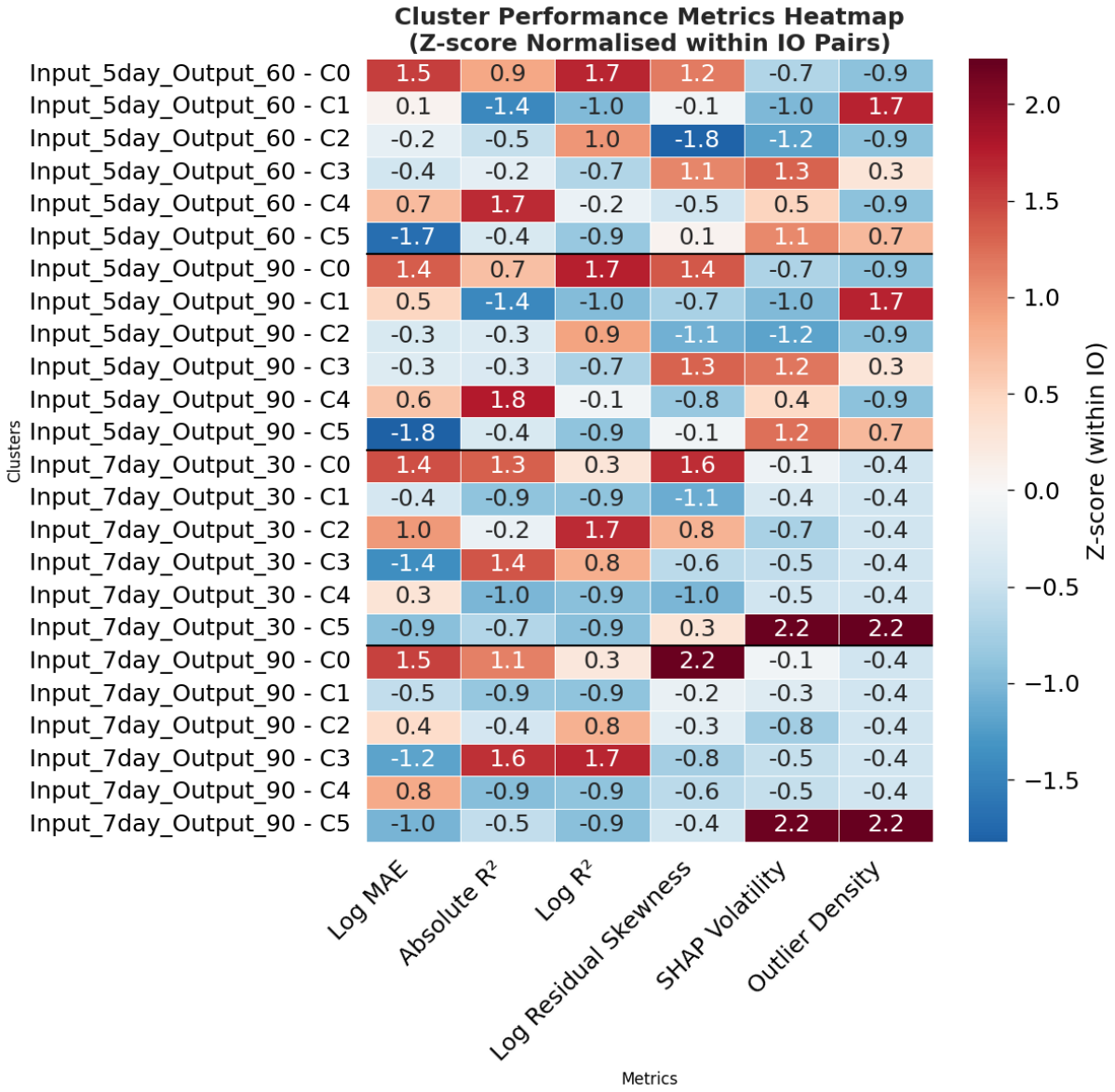


Figure 1: Heatmap of cluster-wise evaluation metrics across four I/O configurations (5→60, 5→90, 7→30, and 7→90).

Cluster 0 Extended Shape Features Ablation Analysis

14-Day Features

- `activation_delay_14`
- `rolling_slope_7_14`
- `peak_day_post_day5_14`
- `last_5day_growth_14` *(included in ablation try 2)*
- `tail_std_14` *(included in ablation try 2)*

30-Day Features

- post_peak_flat_days
- rolling_std_15_30
- burst_span_days
- tail_growth_15_30
- tail_peak_day *(included in ablation try 2)*
- peak_drop_ratio *(included in ablation try 2)*
- activation_shape_code *(included in ablation try 2)*
- tail_delta_ratio *(included in ablation try 2)*

Ablation Results

Table 3: Ablation 1 results for Cluster 0 local model

Input Days	Output Days	Samples	Log R^2	R^2 (Abs)	Log MAE	Residual Skew	MAE	RMSE
7	30	188	0.363	0.059	0.214	-3.72	0.87	2.46
14	30	188	0.567	0.171	0.146	-4.59	0.66	2.31
7	60	188	0.326	0.097	0.300	-2.82	1.25	2.95
14	60	188	0.492	0.307	0.244	-3.10	1.03	2.58
30	60	188	0.839	0.902	0.134	-2.26	0.52	0.97
7	90	188	0.306	0.083	0.345	-2.47	1.55	3.55
14	90	188	0.454	0.385	0.299	-2.47	1.29	2.91
30	90	188	0.680	0.632	0.228	-1.62	1.08	2.25

Table 4: Ablation 2 results for Cluster 0 local model

Input Days	Output Days	Samples	Log R^2	R^2 (Abs)	Log MAE	Residual Skew	MAE	RMSE
7	30	188	0.363	0.059	0.214	-3.72	0.87	2.46
14	30	188	0.576	0.331	0.144	-4.79	0.60	2.08
7	60	188	0.326	0.097	0.300	-2.82	1.25	2.95
14	60	188	0.486	0.255	0.246	-3.03	1.07	2.68
30	60	188	0.838	0.898	0.134	-2.19	0.53	0.99
7	90	188	0.306	0.083	0.345	-2.47	1.55	3.55
14	90	188	0.456	0.369	0.299	-2.48	1.31	2.94
30	90	188	0.696	0.627	0.223	-1.90	1.06	2.27