CIRCL
Computer Incident
Response Center
Luxembourg

Modeling Sparse and Bursty Vulnerability Sightings

Forecasting Under Data Constraints - 2026 FIRST CTI Conference

https://www.vulnerability-lookup.org

Alexandre Dulaunoy - alexandre.dulaunoy@circl.lu Cédric Bonhomme - cedric.bonhomme@circl.lu

April 23, 2026

CIRCL https://www.circl.lu

Contents

Vulnerability-Lookup

Who is behind Vulnerability-Lookup?

Vulnerability-Lookup[1] is an Open Source project led by **CIRCL**.
It is co-funded by **CIRCL** and the **European Union**[2].
Used by many organisations including CSIRTs and ENISA (EUVD).
A reference implementation of the **GCVE** standards[3].

[1] https://www.vulnerability-lookup.org
[2] https://www.restena.lu/en/project/ngsoti
[3] https://gcve.eu

What is Vulnerability-Lookup?

- A unified place to **search**, **triage**, and **track** software and product vulnerabilities from different sources.
- Brings together vulnerability information, correlation of identifiers (e.g., CVE, GHSA, OSV), references, timelines, and risk signals in one view.
- Designed for **CSIRTs**, **SOCs**, **vulnerability managers**, and **developers**.
- **Web UI first** with strong **API** and automation options.
- Supports a complete **CVD process management**[4] along with the ability to fork vulnerability information. Distributed GNA directory.

[4] CNA Program, GCVE GNA Publication

Initial Challenges

- **Volume of data:** Handling a substantial dataset and heavy network traffic, currently over **2,233,664** security advisories and more than **250,000** sightings collected in a year and half[5].
- **Flexibility:** Balancing ongoing development with legacy issues while designing a future-proof architecture. It's complex and yes, sometimes chaotic[6].
- **Robustness:** Validating data even when external entities don't comply with their own JSON schemas.
- **Fast lookup:** Rapidly correlating identifiers across **diverse sources**, including unpublished advisories.

[5] The first sighting on Exploit-DB dates back 26 years.
[6] We enjoy challenges, especially when they lead to practical solutions.

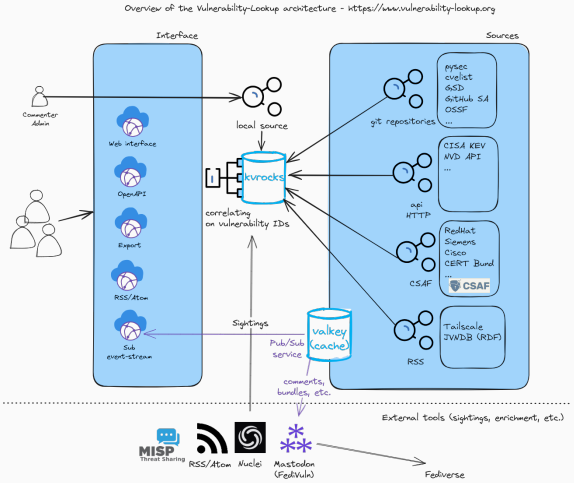
Ongoing Challenges and Development

- **Scoring vulnerabilities:** Aggregating a **large volume** of observations from diverse advisory types to improve vulnerability scoring. Automatic **weighting** of sightings by source or type (e.g., exploited). Automatic **detection of the type** independent of the source.
- **CVD process:** Building an open-source tool to support the Coordinated Vulnerability Disclosure process.[7]
- **Vulnerability numbering:** Enabling a distributed approach through the Global CVE Allocation System.[8]
- **Data quality:** Transforming messy data into clean datasets suitable for model training and analysis.

[7] Aligned with NIS 2 and the Cyber Resilience Act.
[8] `https://gcve.eu`

High-Level Architecture

- Collects and aggregates vulnerability data from 70+ sources.
- Preserves data integrity (never alters the original content).
- Harmonizes data using kvrocks.
- Correlates information across multiple sources (CVE, PySec, etc.).
- Provides both APIs and a Web interface.



Context and Motivation

Why Forecast Vulnerability Sightings?

- Vulnerability sightings are observable events: PoC code, scanner detections, Fediverse discussions, blog references, exploit databases.
- Tracking sightings provides **concrete evidence** that a vulnerability is active.
- Forecasting sightings can help defenders **prioritize patches** and **assess risk** based on real-world activity.

Can we anticipate the evolution of vulnerability-related activities?

https://arxiv.org/abs/2604.16038

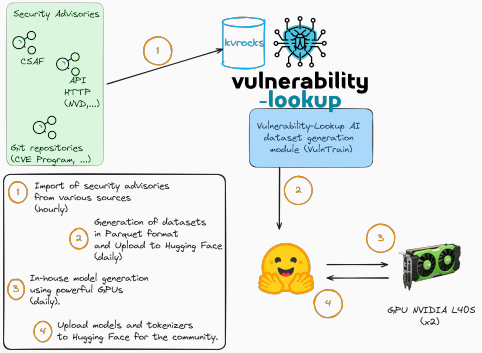
The Data Challenge

- Sighting data is **sparse**: mostly zeros with occasional small counts.
- Activity is **bursty**: sudden spikes followed by rapid decay.
- Time series are **short**: often only 10–30 days of observations per CVE.
- Having around 10 days of sightings is **already valuable for analysis**, but insufficient for many classical forecasting methods.

These characteristics make standard time-series methods unreliable.

Building on Prior Work

- **VLAI**: a RoBERTa-based model predicting vulnerability severity from textual descriptions[a].
- VLAI severity score used as an **exogenous indicator** of potential impact.
- **TARDISsight**[b]: forecasting toolkit for vulnerability sightings.
- **ML-Gateway**[c]: real-time inference from trained models.



[a] https://arxiv.org/abs/2507.03607
[b] https://github.com/vulnerability-lookup/TARDISsight
[c] https://github.com/vulnerability-lookup/ML-Gateway

SARIMAX Forecasting

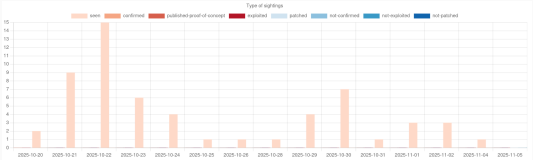
Why we began with SARIMAX

We initially turned to **SARIMAX**[9] (Seasonal Auto-Regressive Integrated Moving Average with eXogenous variables) because it is the "classic" tool for short-term time-series forecasting. Our hope was that it would:

- Capture any underlying autoregressive structure in daily sightings (e.g., "if there was chatter yesterday, there is likely chatter today").
- Allow us to inject external knowledge through the VLAI severity score as an exogenous regressor, testing the hypothesis that higher-severity vulnerabilities generate more sustained attention.
- Provide confidence intervals that would quantify uncertainty for defenders.

[9] https://arxiv.org/abs/2012.03814

What the Data Looks Like



CVE-2025-61932

CVE-2025-59287

Sparse, bursty, short time series. Mostly zeros with occasional spikes. This is the data we are trying to forecast.

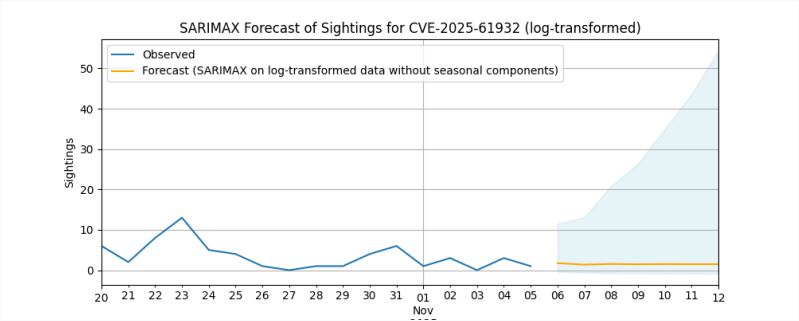
SARIMAX: The Classical Approach

- First attempt: SARIMAX with seasonal terms on daily sighting counts.
- $\log(x + 1)$ transformation to stabilize variance and handle zeros.
- VLAI severity score included as exogenous regressor.

Practical workflow:

1. Aggregate daily sightings per CVE across sources.
2. Compute daily VLAI severity (nearly constant after publication).
3. Log-transform counts as target, severity as covariate.
4. Forecast using average recent severity as exogenous value.

SARIMAX: Results on CVE-2025-61932



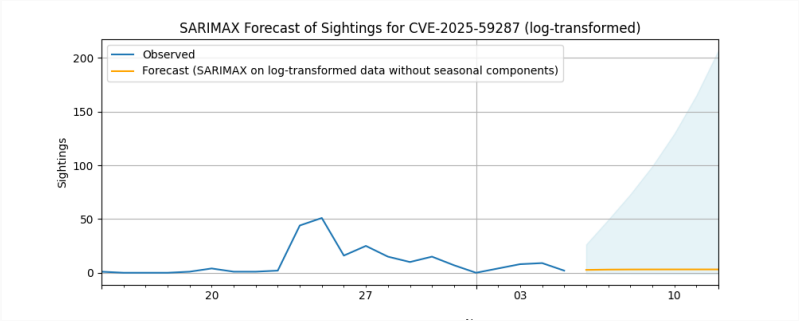
SARIMAX with log-transform, without seasonal components.

SARIMAX: Why It Fails Here

- SARIMAX assumes a **relatively smooth, stationary series** with clear autoregressive or seasonal structure.
- Our data: low-count, burst-like, very short series.
- Confidence intervals **expand unrealistically**.
- Negative forecasts can appear (even with log-transform).
- Sudden spikes cause the model to **extrapolate downward** into negative territory.

SARIMAX typically requires 50–100 observations for reliable estimation. We often have 10–15.

SARIMAX: Wide Confidence Intervals (CVE-2025-59287)



Confidence intervals remain very wide, regardless of sighting distribution.

Poisson Regression

Poisson Regression: A Count-Based Alternative

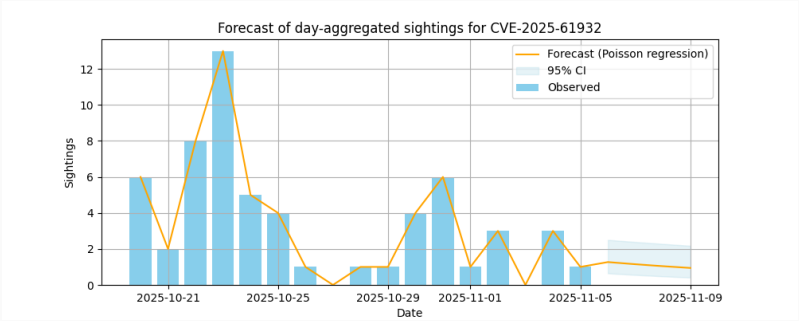
- Designed for **non-negative integer data**.
- Naturally ensures forecasts ≥ 0.
- Can include covariates: days since disclosure, severity, etc.
- Negative binomial variant available for overdispersion.

Applied to top vulnerabilities from monthly reports[10]: **improved results** compared to ARIMA-based methods.

Outcomes indicate general trends but should not be interpreted as precise predictions.

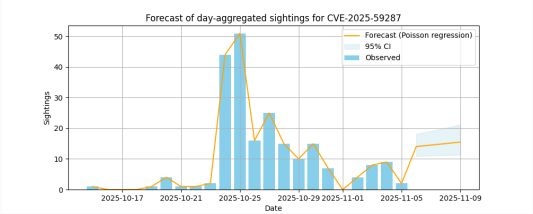
[10] https://www.vulnerability-lookup.org/tags/vulnerabilityreport

Poisson Regression: CVE-2025-61932

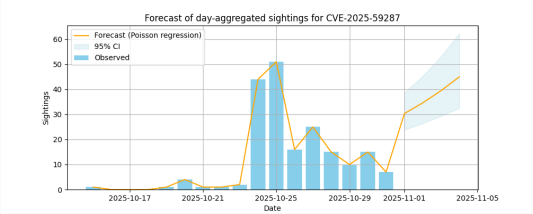


With sufficient sightings, Poisson produces results comparable to exponential decay.

Poisson Regression: CVE-2025-59287



Poisson forecast



Simulation (data up to 2025-11-01)

Growth appears stronger in the simulation; final prediction **overestimates** reality. A sudden drop in sightings is often not captured.

Poisson: Issues and Limitations

- **Very short series**: 10–30 days pushes most models to their limits.
- **Spikes and outliers**: sudden jumps dominate the fit.
- **Dispersion**: under- or over-dispersion violates the Poisson assumption.
- **Exogenous variables**: VLAI severity is nearly constant after disclosure, providing limited variation. Hard to estimate reliably until many observations accumulate.
- **Exploding confidence intervals**: the model "knows" it has too little information.

Poisson assumes **independent events at a roughly constant rate**. Sightings are neither: they arrive in bursts driven by external events (PoC releases, media attention), breaking the core assumption.

Exponential Decay and Logistic Growth

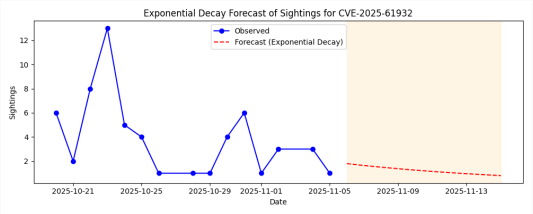
Exponential Decay: Post-Peak Forecasting

The decaying exponential:

$$y(t) = a \cdot e^{-bt} + c$$

- a: initial amplitude above the floor (height of the spike)
- b: decay rate (how fast attention fades)
- c: asymptotic baseline (long-term residual activity)

Appropriate for vulnerabilities **past their peak**.
If still rising, the fit may be flat or unreliable.



CVE-2025-61932

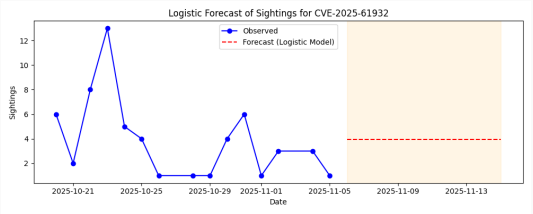
Logistic Growth: Burst-and-Fade Dynamics

The logistic function:

$$y(t) = \frac{L}{1 + e^{-k(t-t_0)}}$$

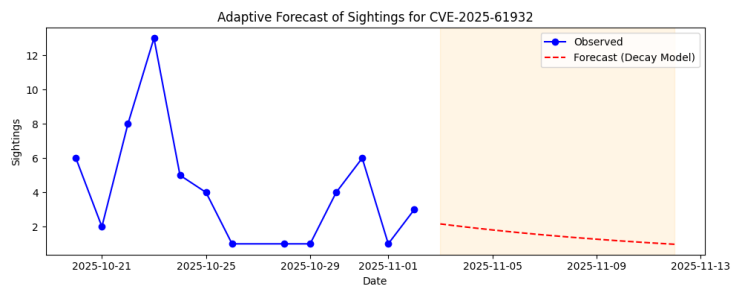
- L: plateau (maximum expected sightings as $t \to \infty$)
- k: growth rate (steepness of the rise)
- t_0: day of inflection (midpoint of the S-curve — growth is fastest)

Captures rapid increase **after disclosure**, then plateau and slow decay.



CVE-2025-61932

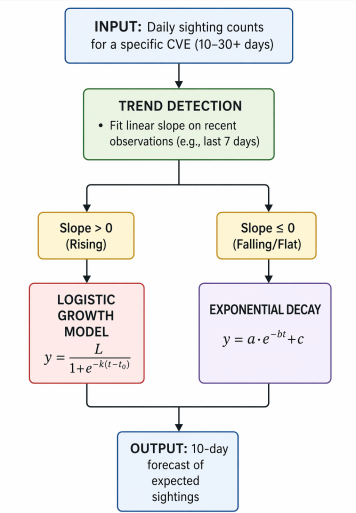
Logistic Model: Forecasting Simulation



Logistic model trained only on sightings up to 2025-11-01. The forecast captures the general trend.

Adaptive Model Selection

Overview of the Adpative Model Selection

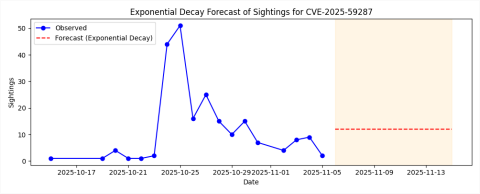


TLP:CLEAR

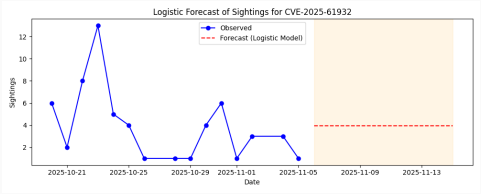
Adaptive Logistic or Exponential Decay

Automatically select the best model based on recent trend:

1. **Trend detection**: check if recent sightings are increasing or decreasing (linear slope).
2. **Model selection**: use logistic if slope > 0, otherwise exponential decay.
3. **Curve fitting**: estimate parameters via `curve_fit`.
4. **Forecasting**: extend the model into the future.



CVE-2025-59287: adaptive selects decay.



CVE-2025-61932: adaptive selects logistic.

From Theory to Practice

Forecasting by Sighting Source

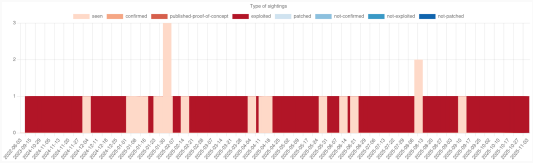
The adaptive approach was applied to forecast sighting activity across all major sources[11]:

- **Social platforms** (Fediverse, Bluesky): volatile, event-driven patterns reflecting community discussions. Early detection signals.
- **Code sharing** (Gist): sharp spikes coincide with PoC releases and data leaks. Rising trend, stabilizing around 32 sightings/day.
- **Structured intelligence** (MISP, Shadowserver): smoother, more predictable trends. Higher-confidence signals. MISP activity peaking near 96/day.
- **Exploitation tools** (Metasploit): peaks align with module releases. Exponential decay pattern.

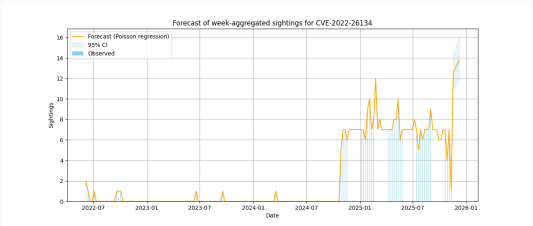
Social sources detect early. Structured sources confirm and validate.

[11] https://www.vulnerability-lookup.org/2025/12/02/end-of-year-threat-intelligence-sightings-forecast/

CVE-2022-26134: Extended Observation Period



Sightings over time



Poisson regression

The logistic model appears largely unaffected by recent bursts for vulnerabilities monitored over an extended period. Confirmed with Shadowserver data.

TARDISsight vs EPSS

Both help answer *"which CVE should I patch first?"* — but they answer different questions.

	EPSS[12]	TARDISsight[13]
Predicts	P(exploitation) within 30 days	Expected # sightings per day/week
Output	Probability $\in [0, 1]$ + CI	Count forecast + CI
Approach	Gradient-boosted trees on ∼1000 static features	Time-series (Poisson, decay, logistic) + VLAI
Input	Different sources including closed sources	Real-time sightings stream (∼10–30 obs)
Horizon	30 days fixed, updated daily	1–14 days, recomputable hourly
Burstiness	Absorbed in the 30-day label	Explicitly modeled (logistic rise, decay fade)
Blind spot	When/intensity invisible	Useless on day zero; wide CI on short series

Complementary, not competing. EPSS tells you *whether* to worry; TARDISsight tells you *how much* and *for how long*.

[12] https://arxiv.org/pdf/1908.04856

[13] https://arxiv.org/pdf/2604.16038

Practical Recommendations

- **Gather more data**: combine multiple sources, extend observation windows.
- **Start simple**: for very short windows, a rolling average or exponential decay will outperform SARIMAX.
- **Data quality**: check for missing days, duplicates, extreme spikes. Apply clamping and logarithmic transformation before modeling.
- **Use count-based models**: switch to Poisson if negative forecasts appear.
- **Detect bursts early**: helps select the appropriate model and set initial parameters for logistic growth.
- **Reserve SARIMAX**: only for much longer series (months of observations).

Future Work

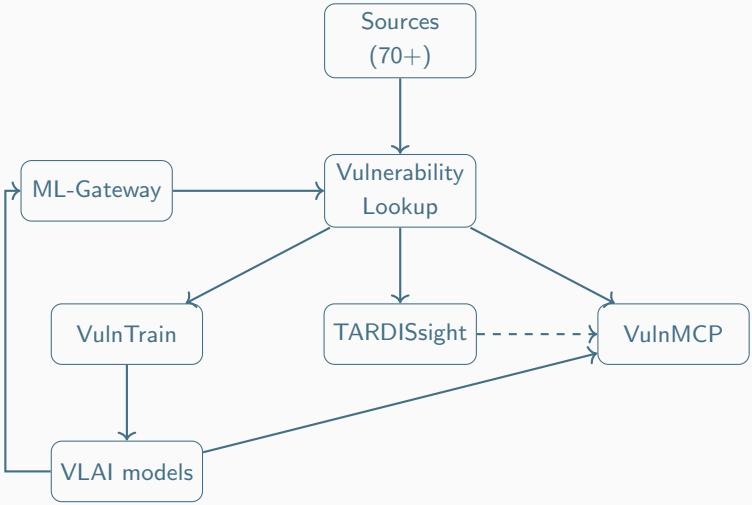
What Comes Next

- **Real-time updating**: a production forecasting module in Vulnerability-Lookup updating predictions daily as new sightings arrive.
- **Anomaly detection**: detecting unusual spikes and missing days to automatically select the most appropriate model.
- **Differentiated sightings**: incorporate sighting types (Seen, Confirmed, Exploited, Patched) and VLAI severity scores.
- **Link to exploitation**: establish a connection between sighting patterns and actual vulnerability exploitation.
- **VulnMCP**[14]: expose forecasting and sighting data to AI agents via the Model Context Protocol, enabling automated querying, enrichment, and report generation.

[14] https://github.com/vulnerability-lookup/VulnMCP

Closing

From Vulnerability Data to Intelligence



References

- https://www.vulnerability-lookup.org
- https://vulnerability.circl.lu
- https://github.com/vulnerability-lookup/TARDISsight
- https://github.com/vulnerability-lookup/vulnerability-lookup
- https://social.circl.lu/@circl
- https://huggingface.co/CIRCL

Thank you for your attention

@adulau@infosec.exchange

@cedric@fosstodon.org

For support and questions: info@circl.lu

TLP:CLEAR