Project Report: Building Honeypots to Track AI Web Scrapers

Submitted to KAUST Academy by Hamza Ayfan

Group Members and Contributions

1. Hamza Ayfan – King Abdulaziz Senior Student

- Performed EDA to find trends in the dataset.
- Created visualizations of various dataset characteristics, statistics and trends.
- Enhanced the processing pipeline.
- Worked on the clustering model using K-Means and DBSCAN.
- Helped with debugging the honeypots website.
- Set up an Ubuntu virtual machine on Oracle Cloud to host the website. Other tools used include Nginx, Gunicorn, and Supabase for storing logs.
- Developed the matching algorithm to determine spoofing.
- Worked on presentations.

2. Eva Aggarwal – Duke Sophomore Student

- Worked on clustering using HDBSCAN to find anomalies.
- Created visualizations for the clustering results.
- Researched and reviewed literature regarding alternative anti-scraping methods.
- Worked on developing a CLI application to package the models for deployment.
- Managed organizing meetings, role assignments, and progress reports.
- Communicated with project lead and project manager on behalf of team.

3. James Xaio – Duke Sophomore Student

- Developed and maintained the honeypots website using Flask.
- Performed EDA on the dataset.
- Created visualizations for the presentations.
- Collected ground truth data for spoofing detection models.

4. Adonias Ketema – Duke Sophomore Student

- Developed the final clustering model based on HDBSCAN.
- Analyzed anomalies in the results of the clustering.
- Created labels for clustering results to be used for scraper categorization.

5. Zini Yang – Duke Master's Student

- Created the pipeline used to preprocess the dataset for the models.
- Tested various classification models to be used in spoofing detection. Including: XGBoost, RandomForest, LinearRegression, etc.
- Developed the classification model for spoofing detection.

Mentors

Dr. Emily Wenger – Project lead and faculty member

Cindy Gao – Project manager and master's student

Project objectives and goals

1. **Track AI web scrapers** using honeypot websites.
2. **Detect spoofing** based on mismatches between user agents and IP/ASN data.
3. **Cluster bot behavior** to discover usage patterns and identify anomalous activity.
4. **Develop deployable tools** including CLI apps for real-time spoof detection.

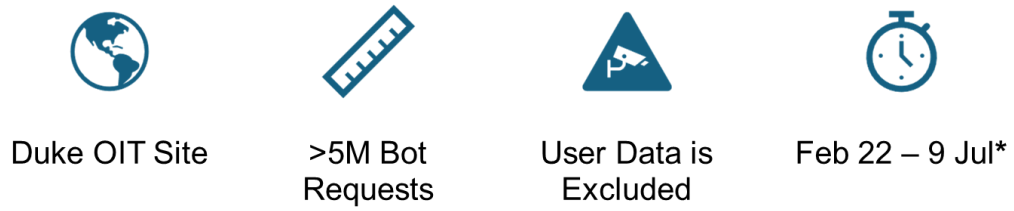
Current Progress and Milestones Achieved

Phase	**Tasks**	**Status**
Week 1–2	Literature Review, Tool Setup	✅ Completed

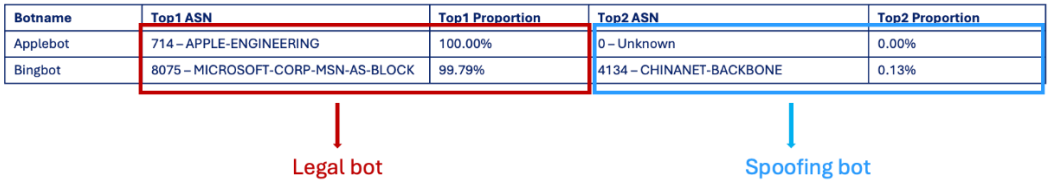
Week 3–5	EDA and Visualization	✅ Completed
Week 6–7	Clustering and Anomaly Detection	✅ Completed
Week 8–9	Spoofing Classifier	✅ Completed
Week 10–11	Honeypot Deployment	✅ Completed
Week 12	CLI App Development	In-Progress
Week 13	Poster and Artifact	✅ Completed

Methodology and Approach

- **Data Collection**: Server logs from Duke OIT's public-facing services.



- **Spoofing Detection**: Labeled spoofers via ASN-botname inconsistency; trained classifiers with ASN and behavioral features.

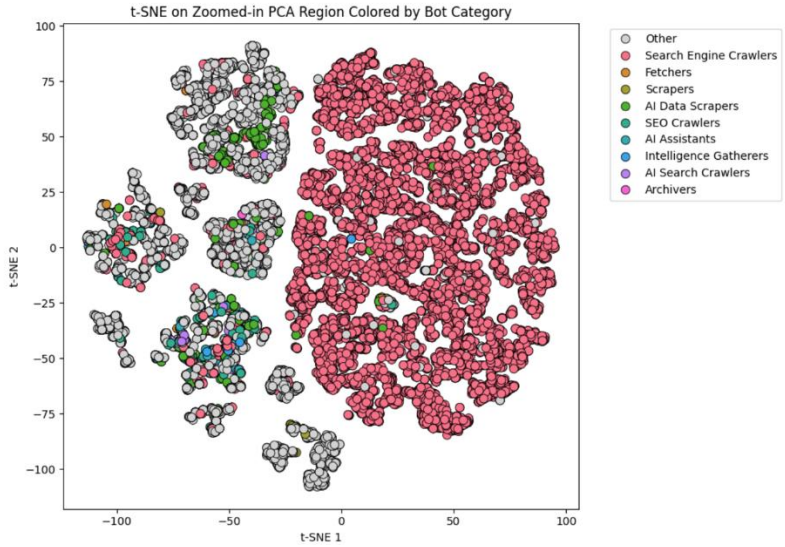


Botname	Top1 ASN	Top1 Proportion	Top2 ASN	Top2 Proportion
Applebot	714 – APPLE-ENGINEERING	100.00%	0 – Unknown	0.00%
Bingbot	8075 – MICROSOFT-CORP-MSN-AS-BLOCK	99.79%	4134 – CHINANET-BACKBONE	0.13%

Legal bot

Spoofing bot

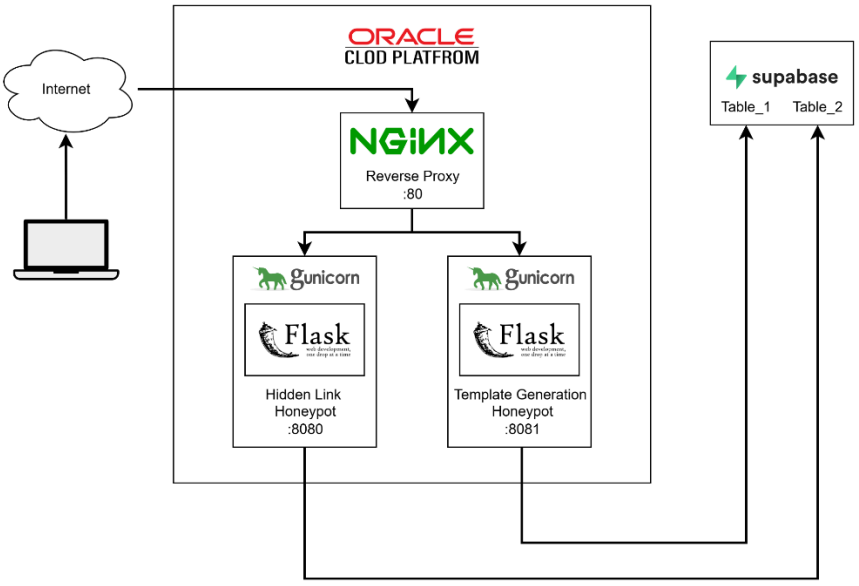
- **Clustering:** Used HDBSCAN to identify behavioral groupings and spot outliers.



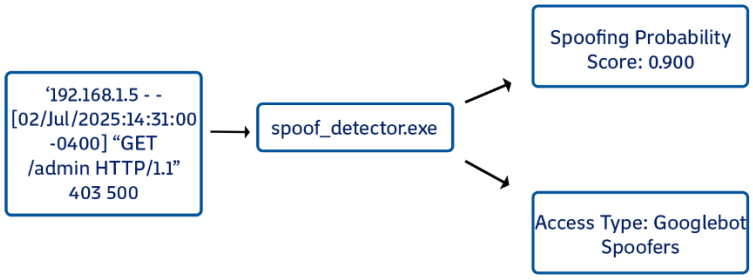
- **Honeypots**: Designed trap websites with realistic templates and hidden links to attract scrapers.



- **Deployment**: Hosted via Oracle Cloud using Flask, Gunicorn, and Nginx; logs sent to Supabase.



- **CLI App**: Enables real-time spoofing classification using pretrained models and structured logs.



Preliminary Results or Findings

Classification

- XGBoost achieved up to **95% recall** with ASN features.

Model	Recall	Recall (without ASN)	Precision (with ASN)	Precision (without ASN)
Random Forest	88.41%	42.68%	**100%**	**100%**
XGBoost	95.12%	**62.80%**	**100%**	89.57%
Gradient Boosting	**95.73%**	38.41%	96.91%	79.23%

- Performance drops without ASN, confirming its importance in spoofing detection.

Clustering

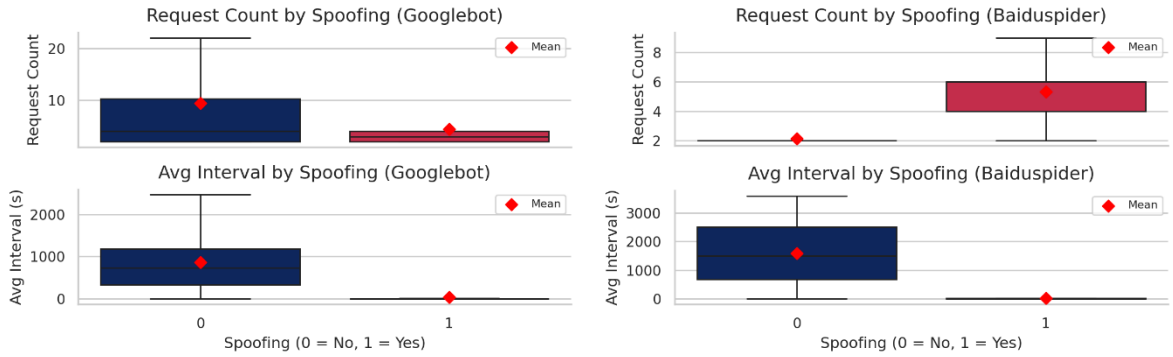
- Identified over **250 small and 10 large clusters**.
- Certain spoofers (e.g., “Mozlila”) revealed through anomalous behavior and user-agent analysis.

Spoofing Rate

Less than 1%, but often from **suspicious ASNs** or obfuscated headers.

Time Series Analysis

Spoofers tend to have shorter request intervals and denser activity bursts.



Challenges and Obstacles Encountered

Data Imbalance: Spoofing bots were <1% of total samples, requiring careful handling and metrics beyond accuracy.

Labeling Ambiguity: Some spoofing cases are subtle or intentionally ambiguous.

Backend Complexity: Coordinating deployment on cloud servers with Nginx and Supabase required multiple debugging sessions.

Realism in Honeypots: Creating convincingly real yet trap-worthy content proved challenging.

Next Steps and Future Work

Enhance Realism in Honeypots: Automate generation of human-like content using LLMs.

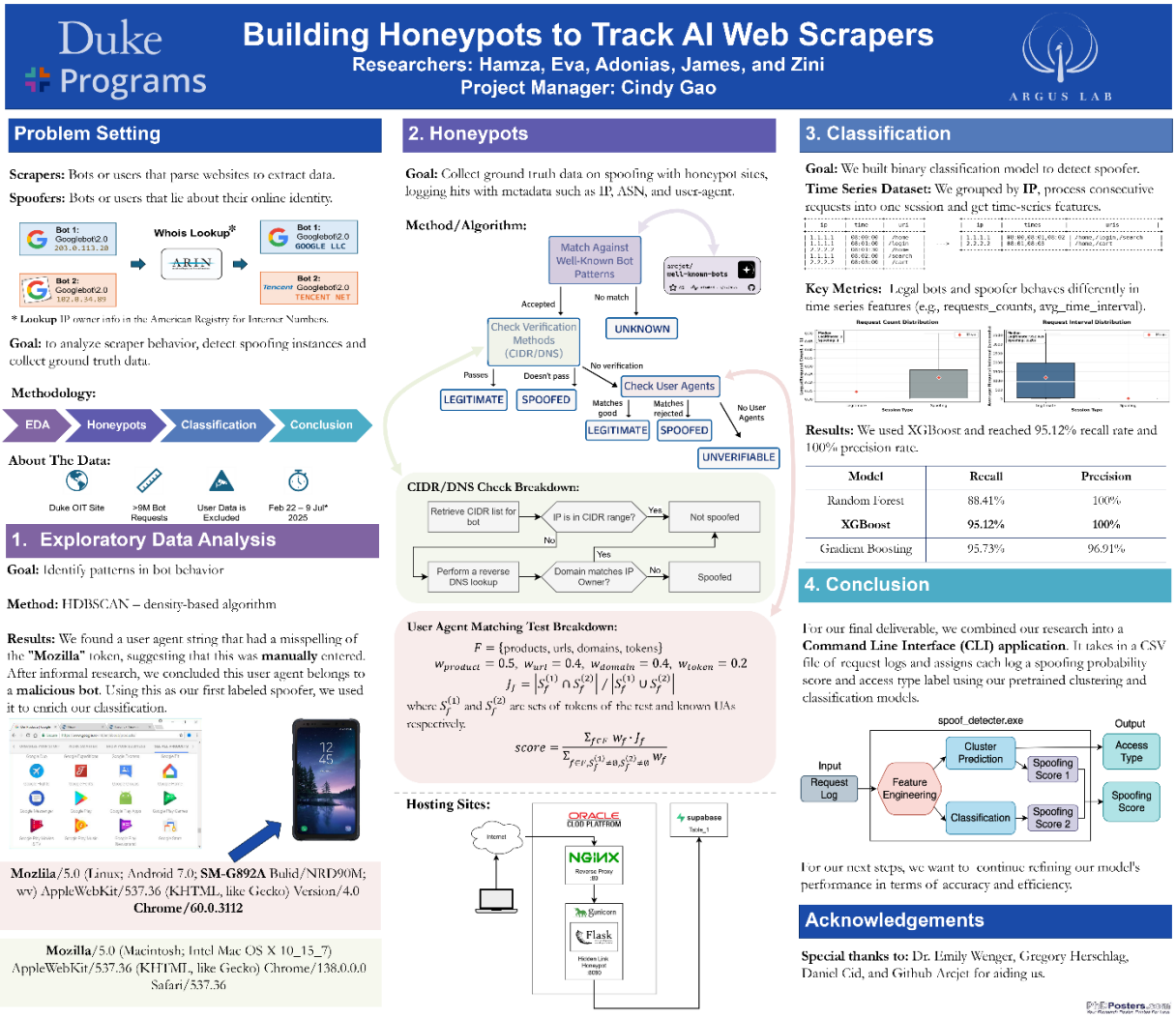
Streaming Detection: Integrate real-time log ingestion and detection pipelines.

Partnerships with Security Teams: Collaborate with campus security or IT for live deployment and alerts.

Expand Detection Features: Include DNS patterns, JavaScript behavior, and rate-limiting analysis.

Poster and Artifact

The artifact is a video explaining the poster and can be found on the project page on the Duke website as it becomes ready.



Embedded poster file:

Data+ Honeypots Poster (1).pdf

Resources and Tools Used

Cloud Infrastructure: Oracle Cloud VM, Nginx, Gunicorn

Database/Logging: Supabase

Frameworks: Flask (website), Scikit-learn, XGBoost, HDBSCAN

Visualization: Matplotlib, Seaborn

Deployment Tools: CLI wrapper, ArgumentParser, shell scripting