LinkedIn Anti-Abuse AI

Preventing Abuse Using Unsupervised Outlier Detection

KDD 2021: Outlier Detection and Description Workshop

2021-08-15

James Verbus

jverbus@linkedin.com

Unique Challenges in Anti-Abuse

Unsupervised outlier detection is a solution

Labels

Few “ground truth” labels for model training or evaluation

Unique Challenges in Anti-Abuse

Unsupervised outlier detection is a solution

Labels

Few “ground truth” labels for model training or evaluation

Signal

Limited signal for individual fake accounts

Unique Challenges in Anti-Abuse

Unsupervised outlier detection is a solution

Labels

Few “ground truth” labels for model training or evaluation

Signal

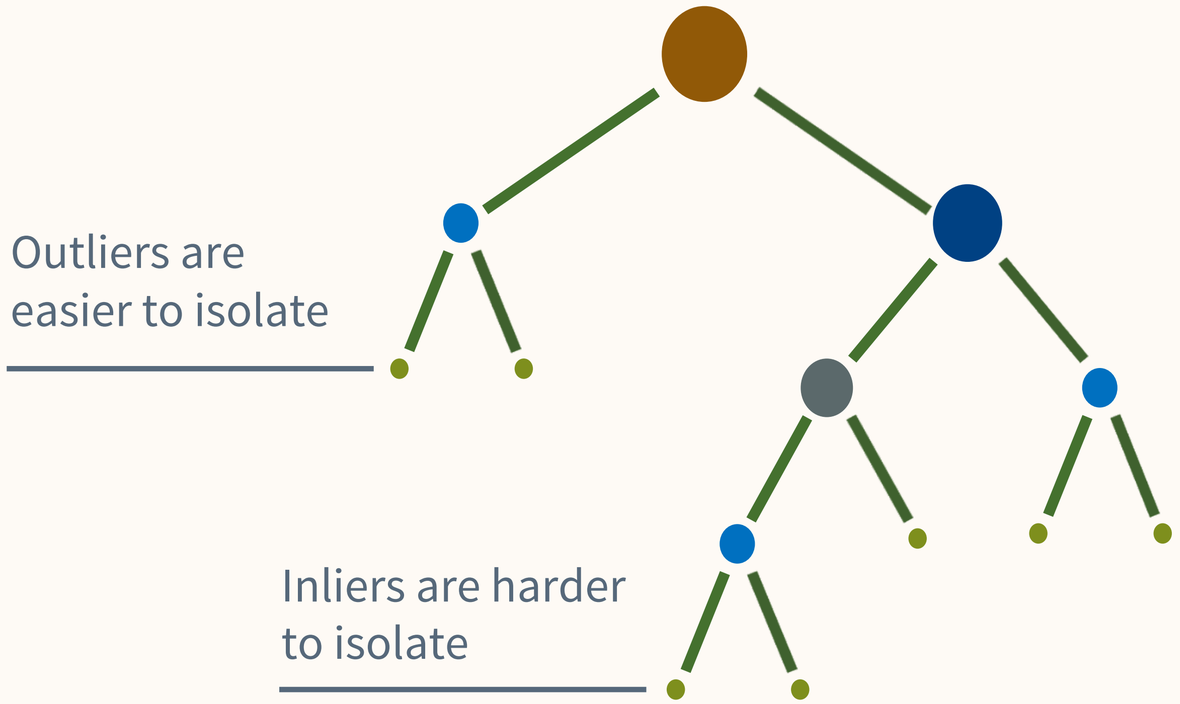
Limited signal for individual fake accounts

Adversarial

Attackers are very quick to adapt and evolve

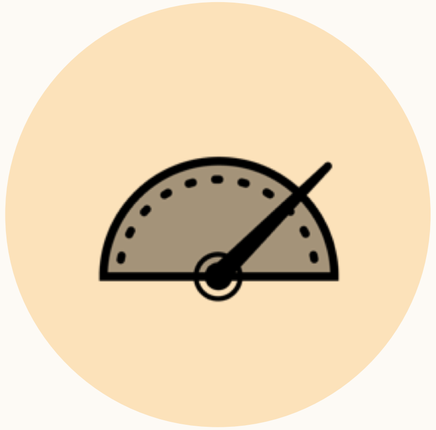
Unsupervised Outlier Detection: Isolation Forests

Isolation Forest



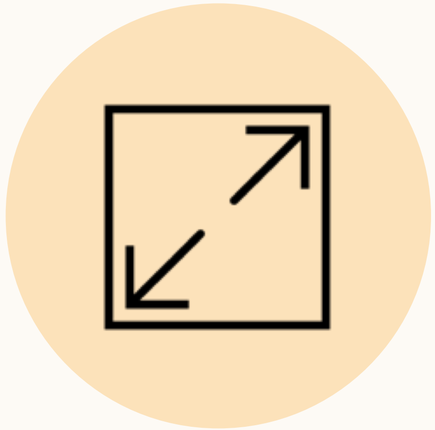
- Proposed by Liu et al. in 2008
- Ensemble of randomly created binary trees
- Tree structure captures multi-dimensional feature distribution
- Future instances can be scored

Isolation Forest Advantages



Performant

A top performer in recent benchmarks



Scalable

Low computation and memory complexity

Fewer assumptions

No assumptions about data distributions or distance metrics

Widely used

Active academic research and industry use

Isolation forest modeling process

Isolation forest modeling process

Behaviors of interest

- Use intuition of domain experts
- Leverage known examples from the intended outlier class

Feature engineering

- Create features that separate normal behavior from behaviors of interest
- Construct features carefully: defaults for missing data, value range, transformations…
- Reduce the number of unimportant features

Seed labels

- Curate seed labels
- Generate using basic heuristics on features
- Enrich with other sources of partial labels

Model training

- Use seed labels to evaluate relative performance
- Use random sampling and manual review to estimate the precision of the final model
- Estimate recall by comparing against seed labels
- Improve precision using simple filters on isolation forest output

Isolation forest modeling process

Behaviors of interest

- Use intuition of domain experts
- Leverage known examples from the intended outlier class

Feature engineering

- Create features that separate normal behavior from behaviors of interest
- Construct features carefully: defaults for missing data, value range, transformations…
- Reduce the number of unimportant features

Seed labels

- Curate seed labels
- Generate using basic heuristics on features
- Enrich with other sources of partial labels

Model training

- Use seed labels to evaluate relative performance
- Use random sampling and manual review to estimate the precision of the final model
- Estimate recall by comparing against seed labels
- Improve precision using simple filters on isolation forest output

Isolation forest modeling process

Behaviors of interest

- Use intuition of domain experts
- Leverage known examples from the intended outlier class

Feature engineering

- Create features that separate normal behavior from behaviors of interest
- Construct features carefully: defaults for missing data, value range, transformations…
- Reduce the number of unimportant features

Seed labels

- Curate seed labels
- Generate using basic heuristics on features
- Enrich with other sources of partial labels

Model training

- Use seed labels to evaluate relative performance
- Use random sampling and manual review to estimate the precision of the final model
- Estimate recall by comparing against seed labels
- Improve precision using simple filters on isolation forest output

Isolation forest modeling process

Behaviors of interest

- Use intuition of domain experts
- Leverage known examples from the intended outlier class

Feature engineering

- Create features that separate normal behavior from behaviors of interest
- Construct features carefully: defaults for missing data, value range, transformations…
- Reduce the number of unimportant features

Seed labels

- Curate seed labels
- Generate using basic heuristics on features
- Enrich with other sources of partial labels

Model training

- Use seed labels to evaluate relative performance
- Use random sampling and manual review to estimate the precision of the final model
- Estimate recall by comparing against seed labels
- Improve precision using simple filters on isolation forest output

Isolation forest output

We use outliers identified by an isolation forest model in multiple ways

Measurement

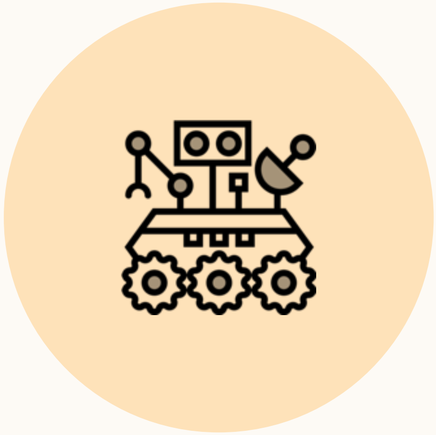
Outliers are used as the foundation of our anti-automation metrics

Labels

Outliers identified with high-confidence are used as training labels for supervised models

Current isolation forest production use cases at LinkedIn

Identification of abuse across a wide variety of product surfaces



Profile scraping

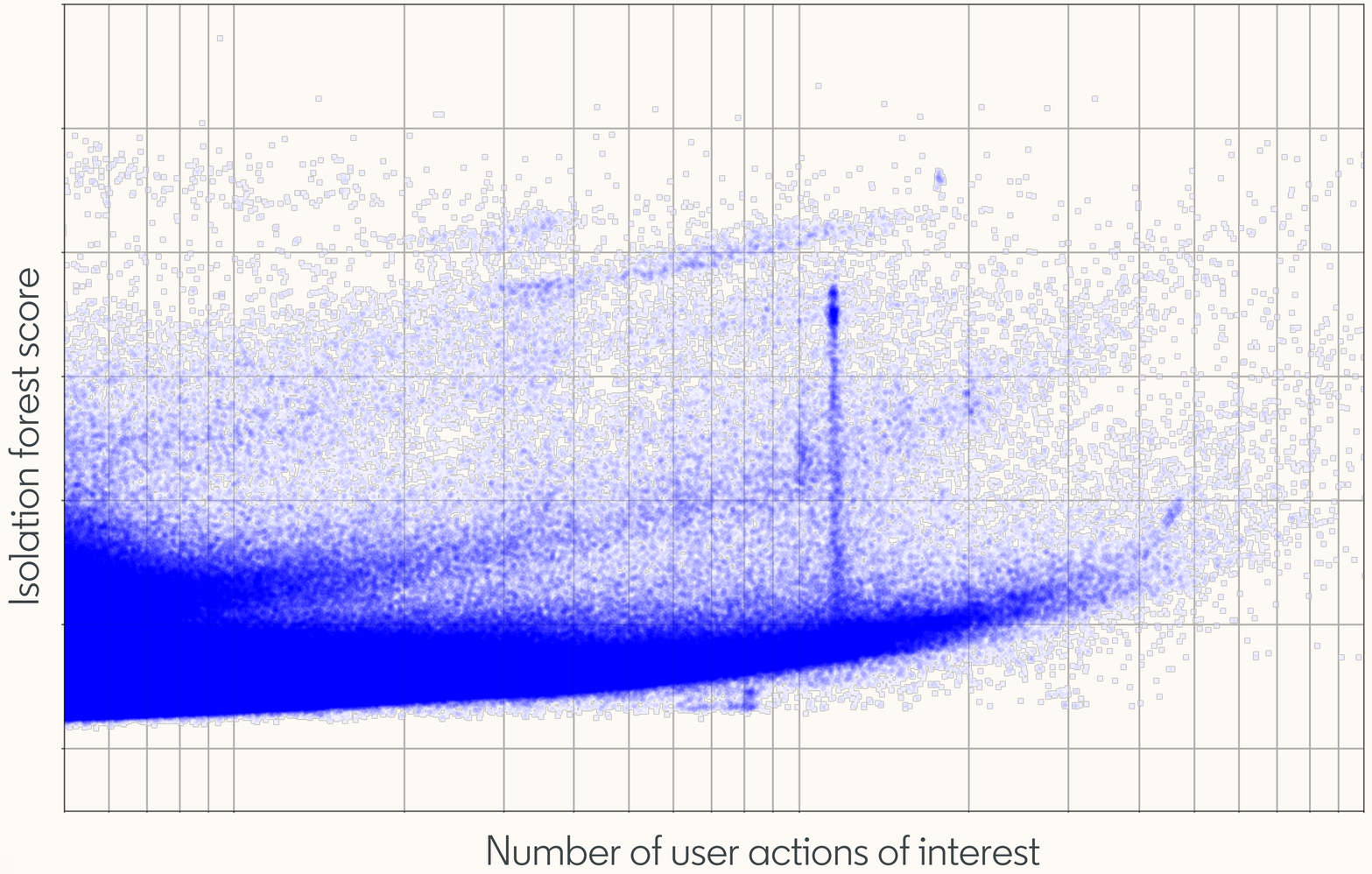
Invitations

Messages

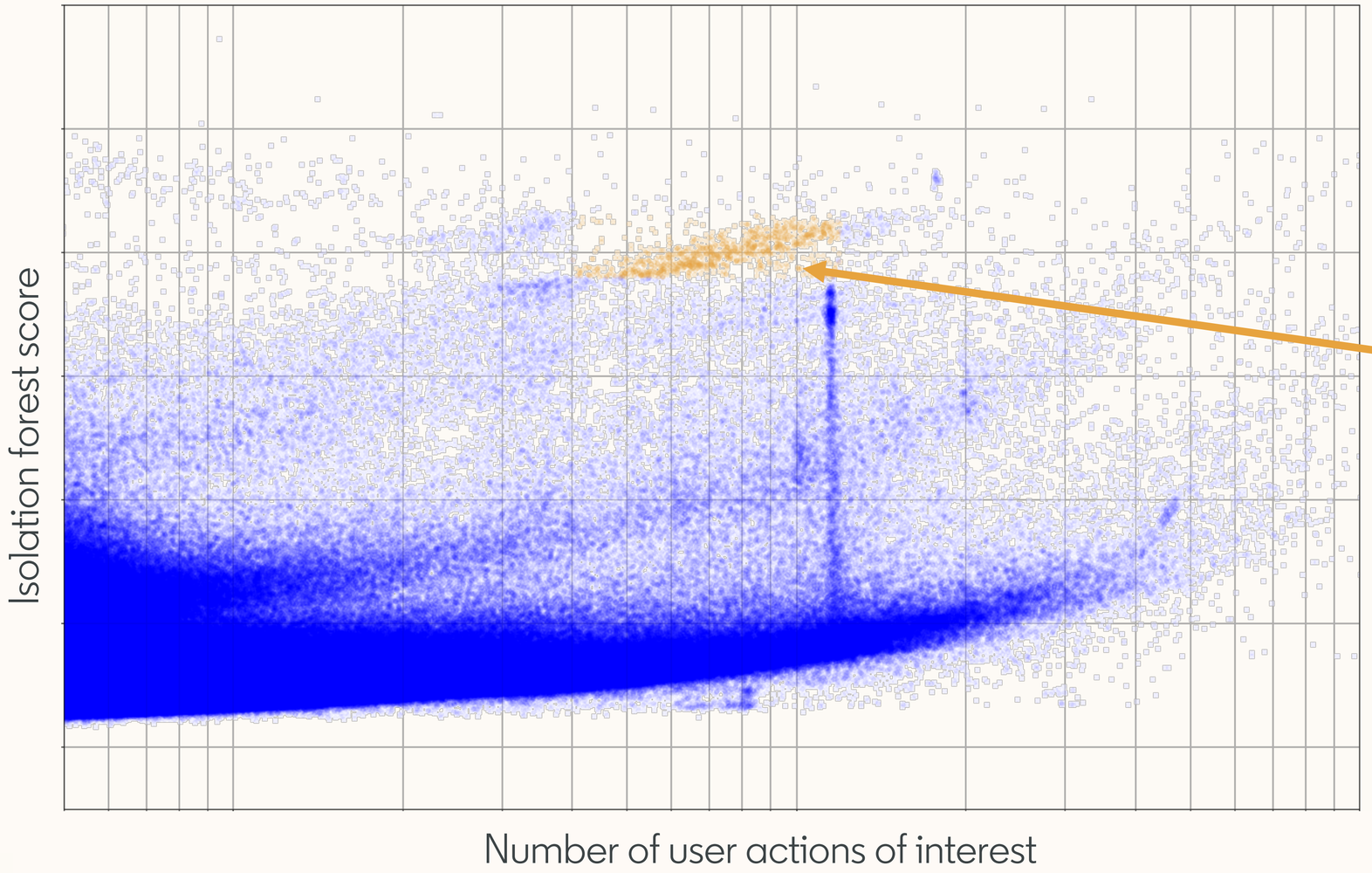
Search

InMails

Results



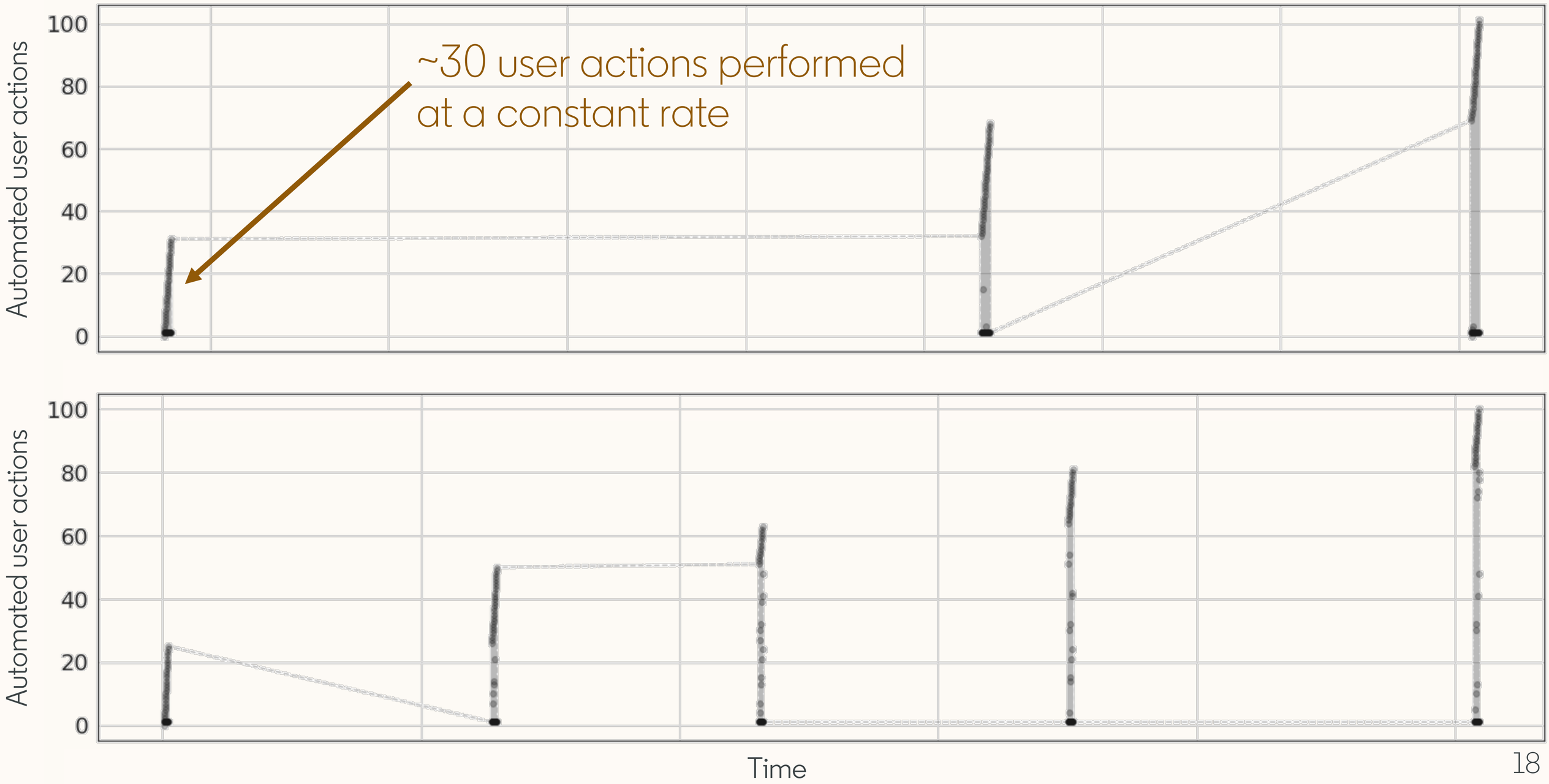
Normal Day

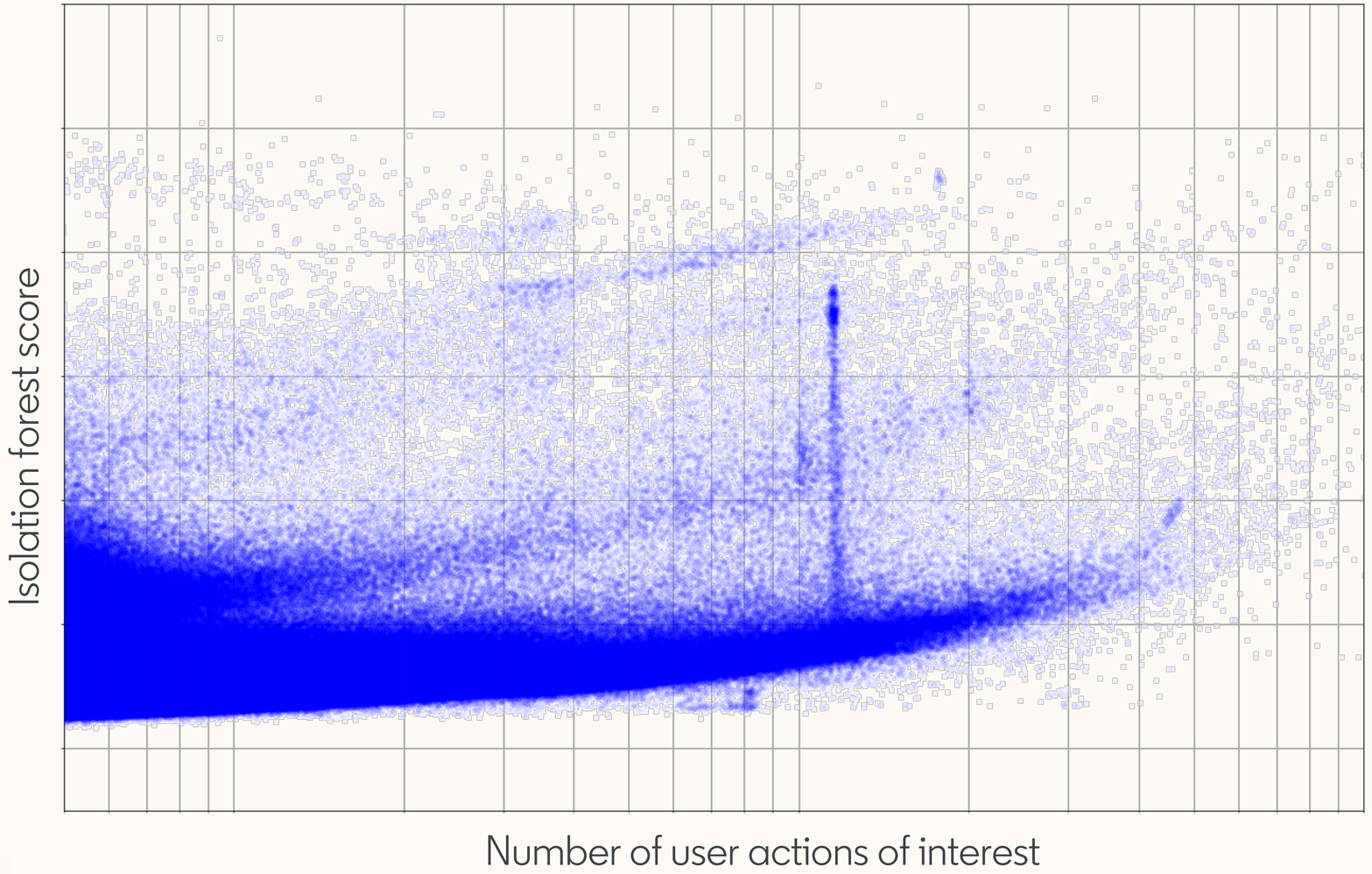


Normal Day

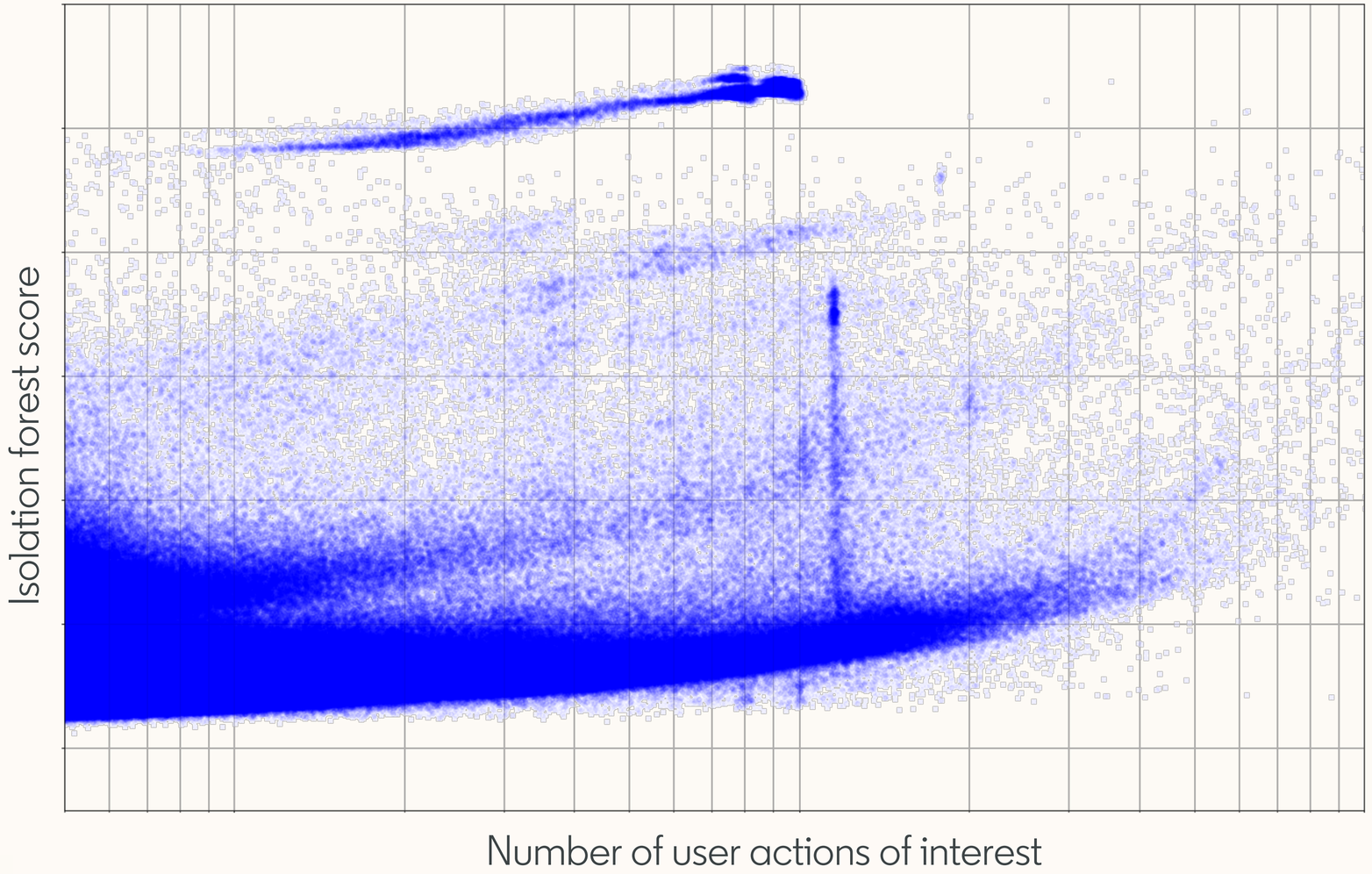
A cluster of real members using automation tools with similar behavior

Behavior of two example real accounts using automation tools

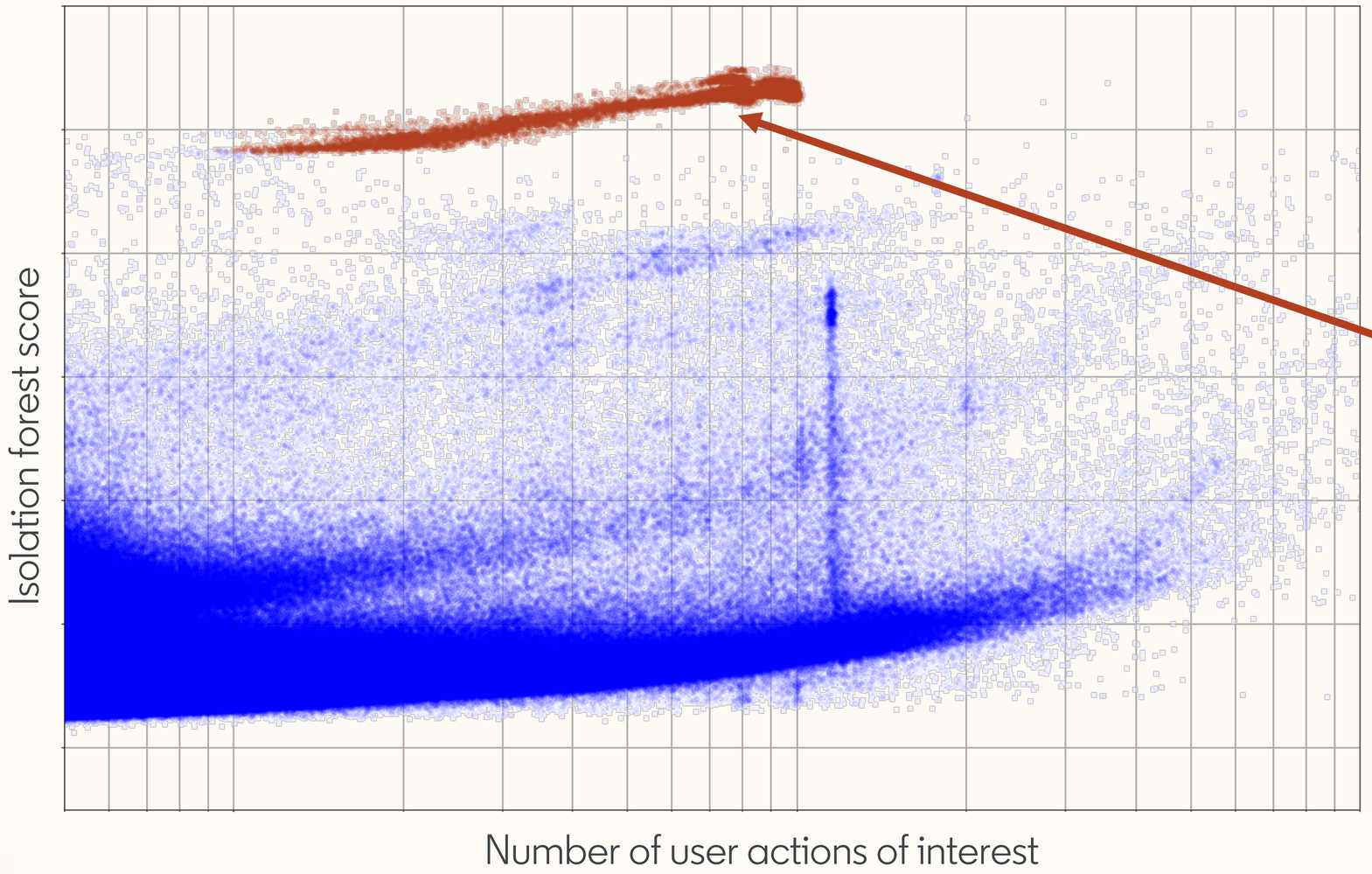




Normal Day



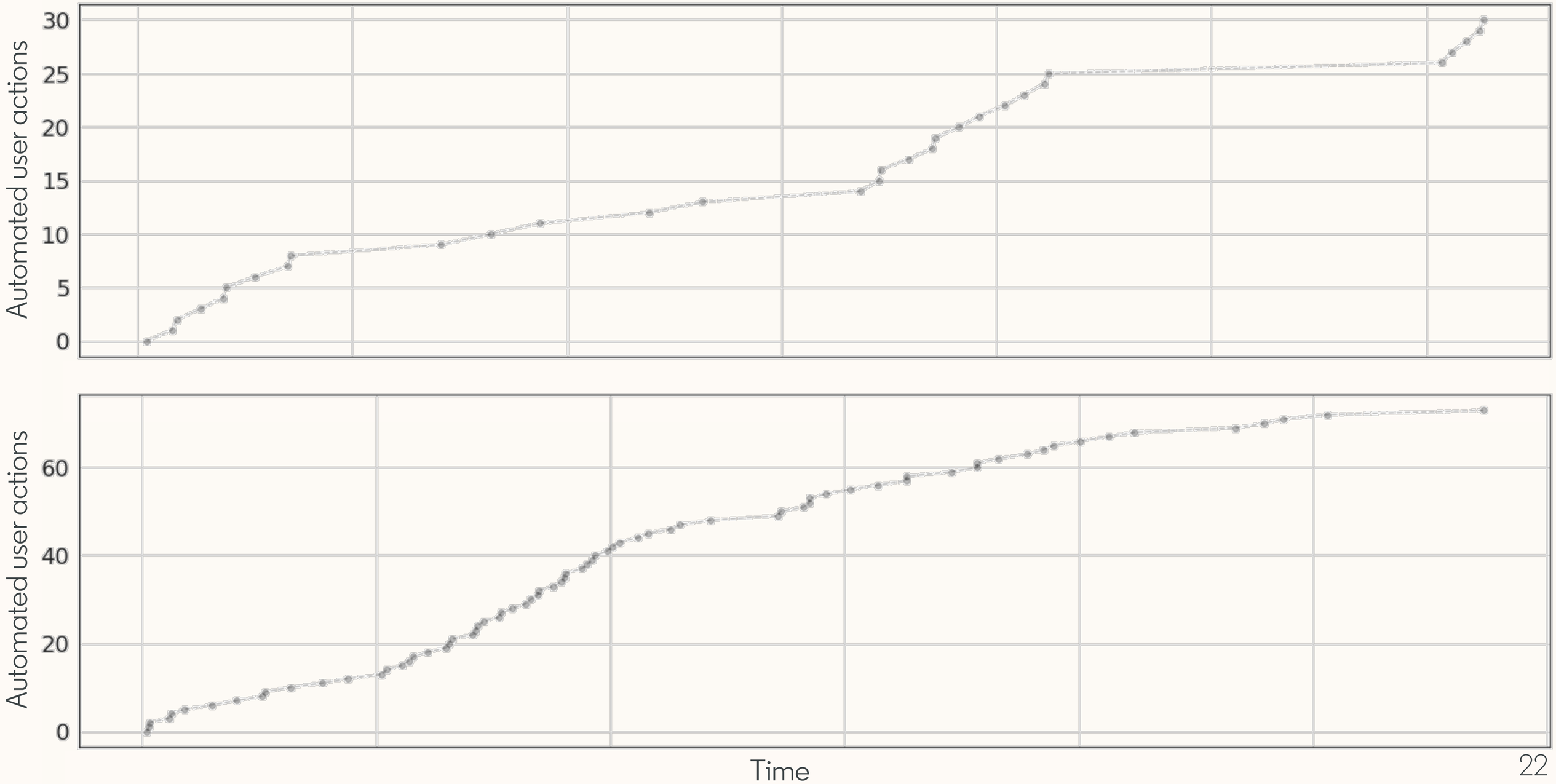
Fake Account Attack Day



Fake Account Attack Day

A major fake account attack using abusive automation

Behavior of two example fake accounts from the attack



Fairness and Transparency

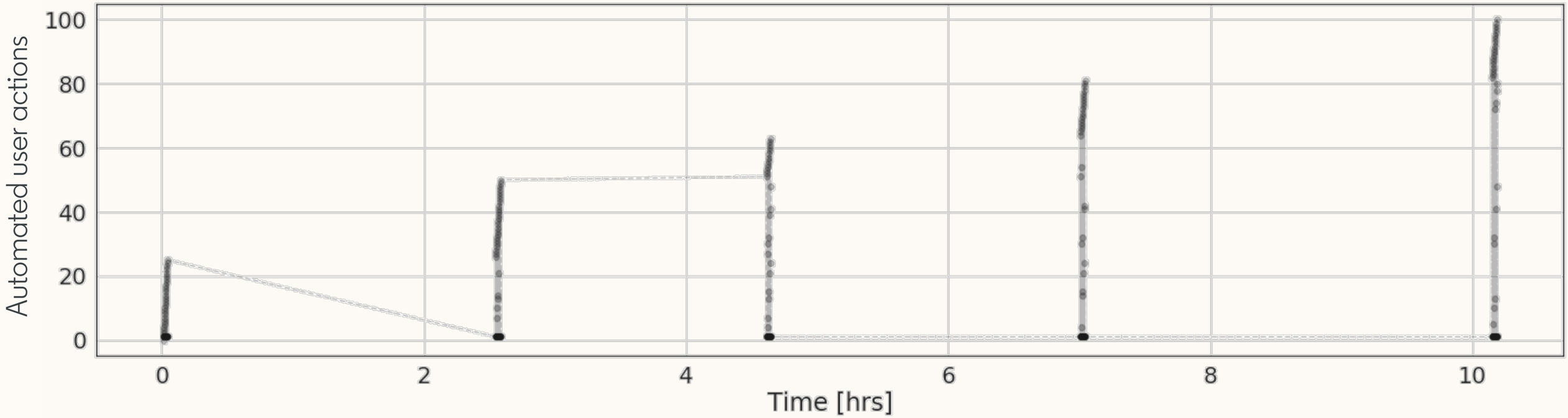
Ensuring model fairness

- **Disparate treatment:** No protected variables in the model; behavioral features only
- **Statistical parity:** Bad actors can control the distribution of protected variables in the abusive population -> **statistical parity is not always a good fairness measure**

- **Model fairness**
  - Use the direct output of the model for measurement only
  - Establish a process for precision monitoring
  - Monitor downstream supervised models and member appeals for fairness

Ensuring model transparency

- Use data visualizations for explainability



- Feature importance

Conclusions

Unique Challenges in Anti-Abuse

Unsupervised outlier detection is a solution

Labels

Few “ground truth” labels for model training or evaluation

Solution: Unsupervised outlier detection is a natural fit for challenges with few labels

Unique Challenges in Anti-Abuse

Unsupervised outlier detection is a solution

Labels

Few “ground truth” labels for model training or evaluation

Solution: Unsupervised outlier detection is a natural fit for challenges with few labels

Signal

Limited signal for individual fake accounts

Solution: Our techniques catch groups of camouflaged bad accounts due to shared behavioral patterns

Unique Challenges in Anti-Abuse

Unsupervised outlier detection is a solution

Labels

Few “ground truth” labels for model training or evaluation

Solution: Unsupervised outlier detection is a natural fit for challenges with few labels

Signal

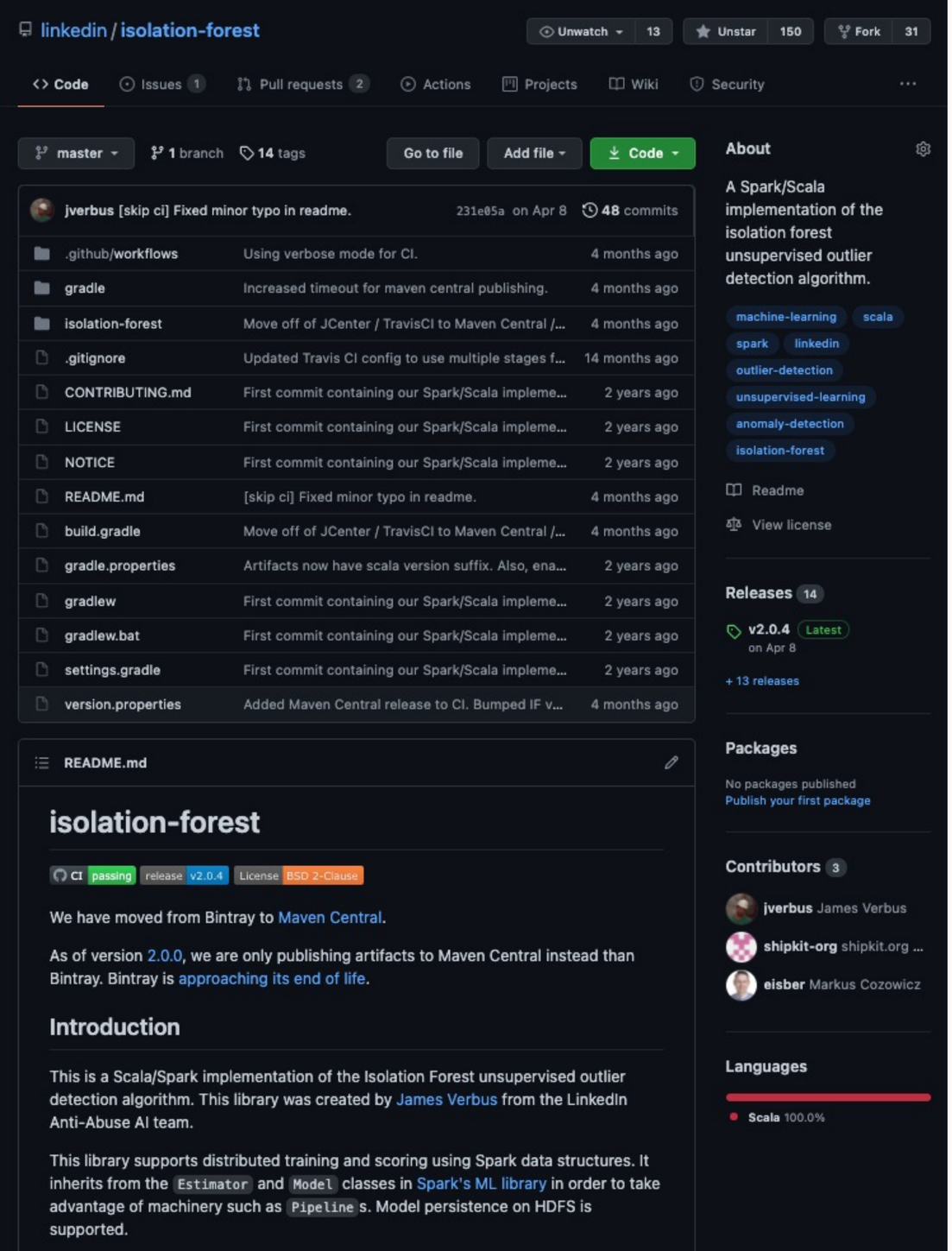
Limited signal for individual fake accounts

Solution: Our techniques catch groups of camouflaged bad accounts due to shared behavioral patterns

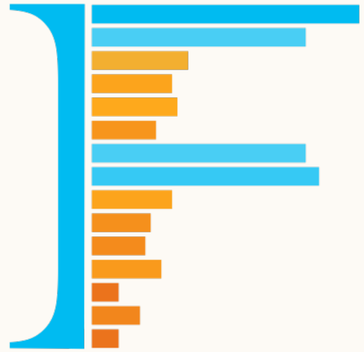
Adversarial

Attackers are very quick to adapt and evolve

Solution: As long as attacker behavior is different than normal user behavior, it can be detected



Open-source isolation-forest Library @ LinkedIn



- Scala/Spark
- Developed by LinkedIn Anti-Abuse AI
- Distributed training and scoring
- Compatible with spark.ml
- Open sourced and available on GitHub
- Artifacts in Maven Central
- Our library powers Microsoft's MMLSpark Isolation Forest

1. https://github.com/linkedin/isolation-forest
2. https://engineering.linkedin.com/blog/2019/isolation-forest
3. https://github.com/Azure/mmlspark

Thank you

James Verbus

jverbus@linkedin.com