Linked Anti-Abuse Al Preventing Abuse Using Unsupervised Outlier Detection KDD 2021: Outlier Detection and Description Workshop 2021-08-15



James Verbus jverbus@linkedin.com



Labels

Few "ground truth" labels for model training or evaluation







Labels

Few "ground truth" labels for model training or evaluation Signal Limited signal for individual fake accounts







Labels

Few "ground truth" labels for model training or evaluation

fake accounts



Signal Limited signal for individual

Adversarial

Attackers are very quick to adapt and evolve



Unsupervised Outlier Detection: Isolation Forests

Linked in Anti-Abuse Al 5



Isolation Forest

Outliers are easier to isolate

> Inliers are harder to isolate

• Proposed by Liu et al. in 2008

- Ensemble of randomly created binary trees
- Tree structure captures multidimensional 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

Linked in Anti-Abuse Al₈



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



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



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

11

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 antiautomation metrics



Labels

Outliers identified with highconfidence 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



Search

InMails



Results

Linked in Anti-Abuse Al 15



score
forest
ation
SO

]												
										۲								
	 	· · · · ·	· · · · · · · · · · · · · · · · · · ·							ъ.	9							
					P			ئىرى ئىز	₽.°°			<u>T</u> Provi	,)	

Normal Day



	score	
J	torest	
•	ation .	
	SO	

Normal Day

A cluster of real members using automation tools with similar behavior



Behavior of two example real accounts using automation tools





_	





score
forest
ation
SO

]												
										۲								
	 	· · · · ·	· · · · · · · · · · · · · · · · · · ·							ъ.	9							
					P			ئىرى ئىز	₽.°°			<u>T</u> Provi	,)	

Normal Day



Φ
Ŭ
S
St
$\underline{\Theta}$
O
Ō
Ē
\overline{O}
0

Fake Account Attack Day



Q	
Ō	
С О	
St.	
Ð	
0	
Ĵ	
Iti	
$\frac{1}{2}$	
SC	



Fake Account Attack Day

A major fake account attack using abusive automation



Behavior of two example fake accounts from the attack



Time

 -

22

Fairness and Transparency

Linked in Anti-Abuse Al 23



Ensuring model fairness

- 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

Disparate treatment: No protected variables in the model; behavioral features only



Ensuring model transparency

• Use data visualizations for explainability



• Feature importance



Conclusions

Linked in Anti-Abuse Al 26





Labels

Few "ground truth" labels for model training or evaluation

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







Few "ground truth" labels for model training or evaluation

Solution: Unsupervised outlier detection is a natural fit for challenges with few labels Limited signal for individual fake accounts

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



Signal





Labels

Few "ground truth" labels for model training or evaluation

Solution: Unsupervised outlier detection is a natural fit for challenges with few labels Limited signal for individual fake accounts

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





Signal

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



linkedin / isol	ation-fo	rest		⊙ Unw	atch - 13	🚖 Unstar 150	양 Fork 31
🗘 Code 💿 Is	sues 1	រិ Pull requests 🙎	 Actions 	凹 Projects	🖽 Wiki (③ Security	
🖓 master 🛨	រឹ វ 1 branch	🛇 14 tags	Go to file	Add file -	⊻ Code -	About	ø
🥑 jverbus [skip	ci] Fixed mi	nor typo in readme.	231e	05a on Apr 8	• 48 commits	A Spark/Scala implementation	of the
.github/workfl	lows	Using verbose mode f	Using verbose mode for CI. 4 months ago			unsupervised outlier	
gradle		Increased timeout for maven central publishing. 4 months ago				detection algorithm.	
isolation-fore	st	Move off of JCenter / TravisCI to Maven Central / 4 months			4 months ago	machine-learning	scala
.gitignore		Updated Travis CI config to use multiple stages f 14 month			14 months ago	spark linkedin	
	NG.md	First commit containing our Spark/Scala impleme 2 years ago			unsupervised-learning		
LICENSE		First commit containing our Spark/Scala impleme			2 years ago	anomaly-detection isolation-forest □ Readme ↓ View license	
		First commit containing our Spark/Scala impleme			2 years ago		
README.md		[skip ci] Fixed minor typo in readme.			4 months ago		
build.gradle		Move off of JCenter / TravisCI to Maven Central / 4 m			4 months ago		
gradle.proper	ties	Artifacts now have sca	ala version suffi	x. Also, ena	2 years ago		
🗋 gradlew		First commit containing our Spark/Scala impleme			2 years ago	Releases 14	
🗋 gradlew.bat		First commit containing our Spark/Scala impleme			2 years ago		
settings.grad	le	First commit containing our Spark/Scala impleme 2 yea			2 years ago		
version.prope	orties	Added Maven Central	release to CI. B	umped IF v	4 months ago		
E README.md	README.md					Packages	
isolatio	n-fore	est				No packages publish Publish your first pac	ed skage
CI passing r	elease v2.0.4	License BSD 2-Clause				Contributors 3	
We have move	d from Bin	trav to Mayon Contral				jverbus Jam	es Verbus

As of version 2.0.0, we are only publishing artifacts to Maven Central instead than Bintray. Bintray is approaching its end of life.

Introduction

This is a Scala/Spark implementation of the Isolation Forest unsupervised outlier detection algorithm. This library was created by James Verbus from the LinkedIn Anti-Abuse Al team.

This library supports distributed training and scoring using Spark data structures. It inherits from the Estimator and Model classes in Spark's ML library in order to take advantage of machinery such as Pipeline s. Model persistence on HDFS is supported.



📰 shipkit-org shipkit.org ..

Languages

Scala 100.0%

eisber Markus Cozowicz

Open-source isolation-forest Library @ LinkedIn

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

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









Thank you



James Verbus jverbus@linkedin.com

Linked in Anti-Abuse Al 31

