A DNS Anomaly Detection Model to Identify Unusual Lagging in Zone Updates

Speaker: Kundana Pillari, UC Irvine and Salesforce
Collaborators: Allison Mankin and Baula Xu, Salesforce

Background

- In large-scale DNS deployments, zone updates are made by DNS hosting services on short timescales to large numbers of servers.
- Normal for the updates to be somewhat asynchronous from each other (DNS is “eventually consistent” by design)
- For our organization, customer sensitivity to stale information is high
- Take action by asking the hosting services to check for server problems when there are anomalous lags

Background

Question: Which lags are anomalous? What threshold of difference in the SOA serial numbers indicates a problem?

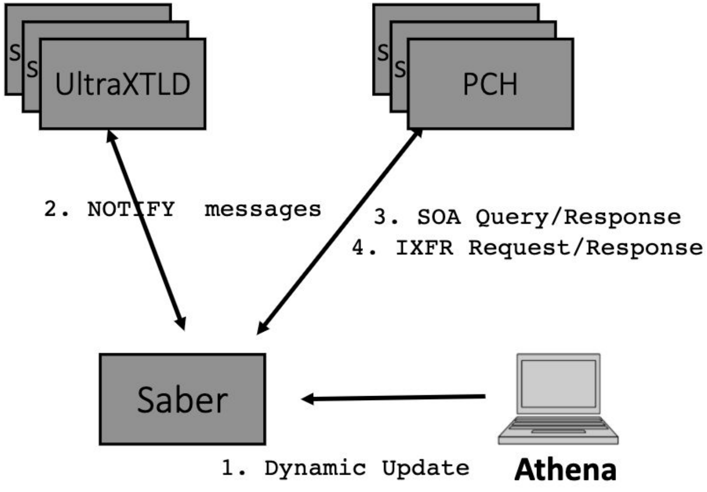
Approach: Use an unsupervised machine learning algorithm to identify the anomalous points in monitor logs to help with this question of what is actionable.

Scope of Anomaly Detection in DNS

- Hidden internal server is Saber
  - Updates external servers

- External servers:
  - UltraDNS
  - PCH

- The scope of work is to find update latency anomalies in external servers

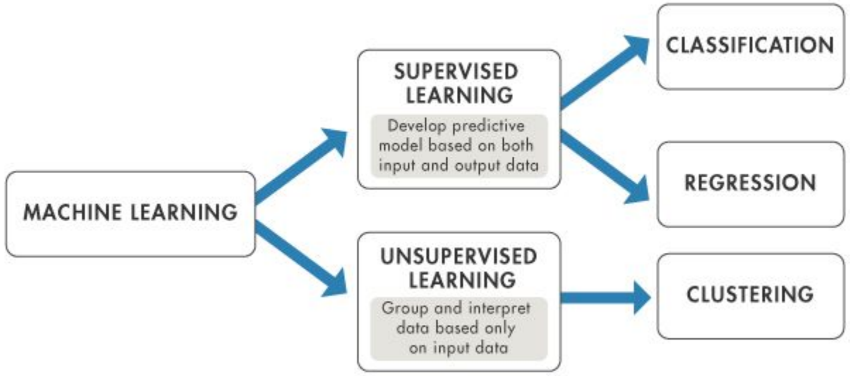


Exploring Anomaly Detection Models

- Explored multiple platforms and algorithms
  - TensorFlow (Autoencoders)
  - Pytorch (LSTM)
  - PyCaret

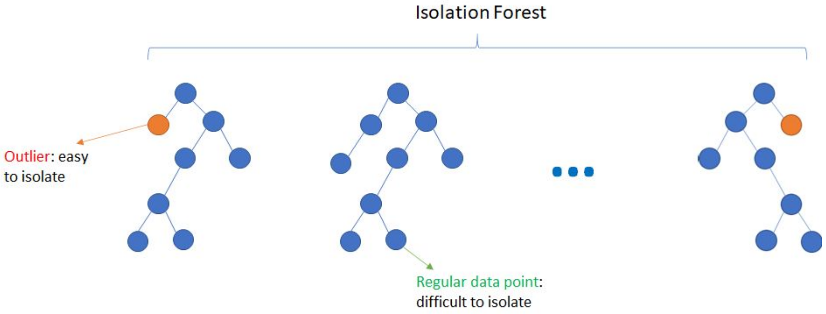
KNN, SVM, PCA, Iforest...

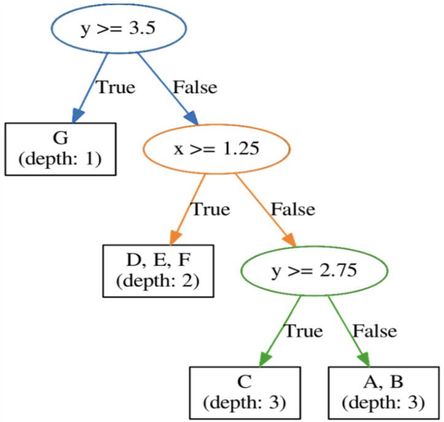
- PyCaret open-source python machine learning library
  - Unsupervised learning
- Used Isolation Forest machine learning algorithm
  - Less memory
  - Faster in performance

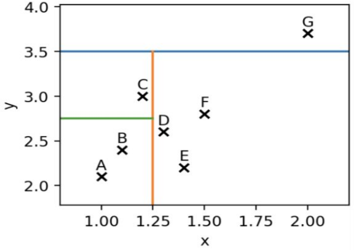


Isolation Forest machine learning algorithm

- Identifies anomalies by isolating outliers in the data
- Works on the principle of the decision tree algorithm
- Randomly selecting a feature from the given set of features
- Forms decision trees using combination for these features
- Outliers will be closer to the root node







Model Training

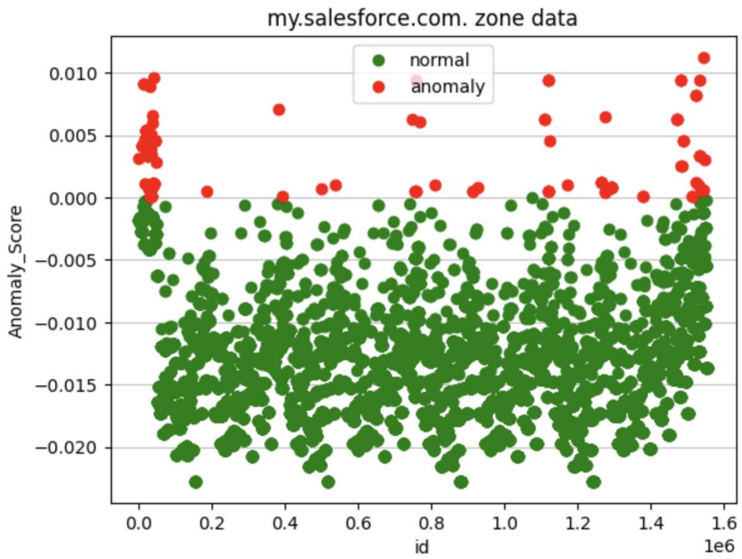
- May and June monitor logs
- Multiple anycast locations for three production zones: my.salesforce.com, salesforce.com, and force.com
- SOA (state of authority) records
- These logs provide timestamp (probe_time) and each server represented by serial numbers that indicates the zone version
- Fine-tuned with a supervised_target (probe_time) to control the learning process

Sample Input:

	id	probe_time	zone	type	status	details
0	40	2021-06-01 00:00:03.919788	my.salesforce.com.	SOA	0	{"UDNS1":"2128885586"\t"UDNS3":"2128885586"\t"...
1	93	2021-06-01 00:02:01.401629	my.salesforce.com.	SOA	0	{"UDNS1":"2128885640"\t"UDNS3":"2128885640"\t"...
2	166	2021-06-01 00:04:01.384063	my.salesforce.com.	SOA	0	{"UDNS1":"2128885695"\t"UDNS3":"2128885695"\t"...
3	235	2021-06-01 00:06:01.300033	my.salesforce.com.	SOA	0	{"UDNS1":"2128885745"\t"UDNS3":"2128885745"\t"...
4	312	2021-06-01 00:08:01.877665	my.salesforce.com.	SOA	0	{"UDNS1":"2128885806"\t"UDNS3":"2128885806"\t"...

Results and Analysis

- Each data point is marked as normal or anomaly
- Analyze Serial Numbers (Zone versions)
- Finding the average differences between normal and anomalous data points
- Finding average differences between normal data points
- Learning trends and patterns in zone updates
- Thresholds for the zones
  - Improve the monitoring system



my.salesforce.com.: 1570.9815442561203

salesforce.com.: 62.941269841269836

force.com.: 569.0604288499026

my.salesforce.com.: 115.96783645507104

salesforce.com.: 0.2705122010774978

force.com.: 9.802812446759296

Results and Analysis

- Provide good numeric thresholds for how much lag was normal and how much anomalous.
- Previously, thresholds were guesstimates made by the team and were the same for all three zones
- Thresholds mark where anomalous behaviors begin
- Thresholds were extremely different for the three zones
- Lagging depended on zone sizes and how frequently the zones were updated

my.salesforce.com.: 1570.9815442561203
salesforce.com.: 62.941269841269836
force.com.: 569.0604288499026

my.salesforce.com.: 115.96783645507104
salesforce.com.: 0.2705122010774978
force.com.: 9.802812446759296

Next Steps

- Dynamic testing
  - Python script
  - Logs to preserve previous thresholds
- Sending alerts for highest anomalous data points
  - Email
  - PagerDuty

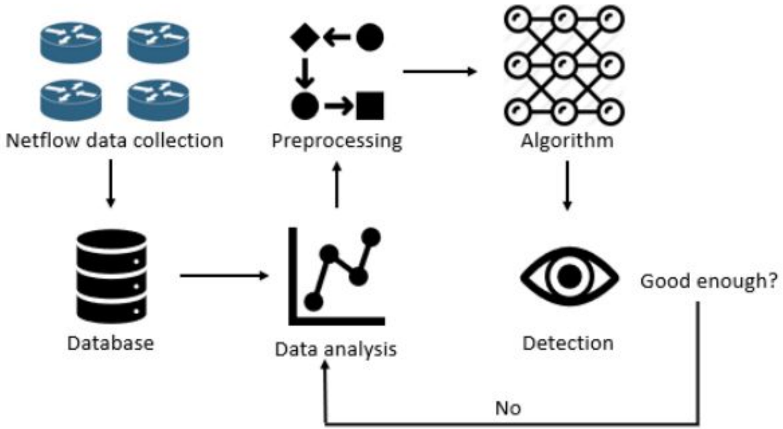
Refocus Dashboard:

salesforce_com										
	PCH1	PCH2	Saber	Saber1B	Saber2A	UDNS1	UDNS2	UDNS3	UDNS4	zone_OK_Percent
SOA	5753	5753	5753	5753	5753	5753	5753	5753	5753	100
TCP	57	136	136	144	3	44	13	119	16	100
UDP	54	29	60	69	1	22	8	60	9	100

siteforce_com										
	PCH1	PCH2	Saber	Saber1B	Saber2A	UDNS1	UDNS2	UDNS3	UDNS4	zone_OK_Percent
SOA	0801	0801	0801	0801	0801	0801	0801	0801	0801	100
TCP	42	42	133	136	3	43	15	123	19	100
UDP	54	21	74	71	2	22	8	59	10	100

Applications of ML in DNS

- ML has been used on DNS data frequently, especially focusing on DNS attacks and security.
  - Anomaly detection in DDOS mitigation
- However, this work covers an area in DNS availability
- ML models, both unsupervised and supervised, can be valuable to many other areas where DNS operators have data in realtime in future.



Thank You