



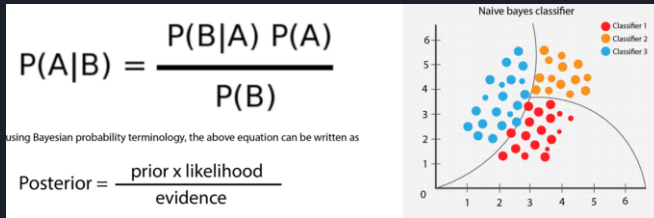
Algorithms

Alex, Jung, Muhamed, Pallavi

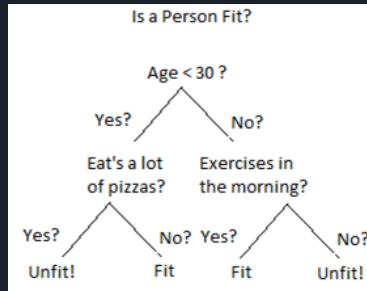
Anomaly detection by Classification

Finding anomalous category of data using data classification.

- Rule Based - if/then -> Rule tree - conditions(T/F)
- Bayesian - calculates statistical probability

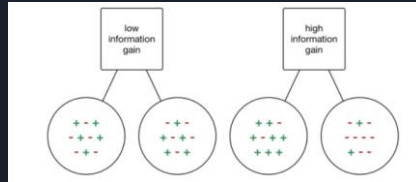


- Decision Tree



Decision Tree

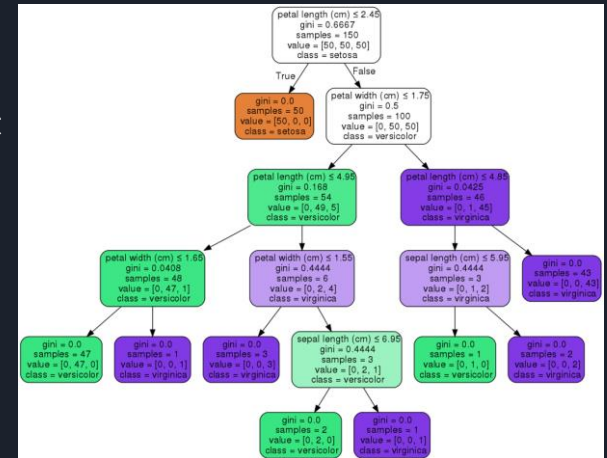
- Induction
 - Creating the tree based on the data.
 - Using greedy algorithm, select best feature to split the tree.



- Pruning
 - Removing the unnecessary branches.
 - Use greedy algorithm to evaluate each rules by removing one, and select the most effective ones.

- Pros
 - Easy to understand.
 - Requires very little data preparation.
 - Cost of creating a tree is $\log(N)$ which is little.
- Cons
 - Overfitting is common
 - If training dataset is biased, the result will be biased too. Needs class balancing to mitigate the effect.
 - Suitable to machine created logs. Low performance for Natural Language.

- Iris dataset



Anomaly detection by Categorization

- In Kibana, there is a categorization wizard to help you create this type of anomaly detection job.

Using the composite decision tree algorithm called **Boosted tree regression** model

- Analyzes the content in the message field to find out the machine generated log's category.
- Can customize the categorization_analyzer to have better results.

Create job: Categorization

1 Time range 2 Pick fields 3 Job details 4 Validation 5 Summary

Pick fields

Categorization detector

Count
Look for anomalies in the event rate of a particular category.
 Selected

Rare
Look for categories that occur rarely in time.
 Select

Categorization field

Specifies which field will be categorized. Using text data types is recommended. Categorization works best on machine written log messages, typically logging written by a developer for the purpose of system troubleshooting.

Categorization field: message

Selected category field is valid

274 field values analyzed, 100% contain valid tokens.
Analyzer used: ml_classic
[Edit categorization analyzer](#)

Examples

```
888 8808 : db-10.16.1.63 [svc-prod-oid-ibadent] pid=8888 ETC-17 [Microsoft][1090 SQL Server Driver][TCP/IP Sockets] SQL Server does not exist. Access denied. -hostname=observer.acme.com;physicalhost=esxserver1.acme.com;vmhost=app1.acme.com
```

Top Influencers

micategory

micategory... 98 381

micategory... 91 95

Anomaly timeline

Overall

View by micategory **Limit** 10

2

4

Anomalies

Severity threshold: warning Interval: Auto

time	severity	detector	found for	influenced by	actual	typical	description	category examples	actions
February 8th 2019, 03:00	98	count by micategory	micategory:2	micategory:2	49	1	49x higher	REC NOT INSERTED...	

Details [Category examples](#)

Terms

REC NOT INSERTED TRAN Table hostname=dbserver.acme.com;physicalhost=esxserver1.acme.com;vmhost=app1.acme.com

Regex

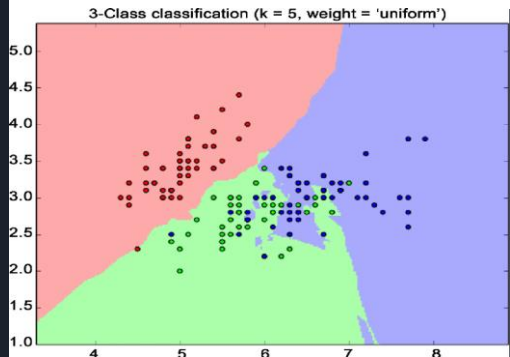
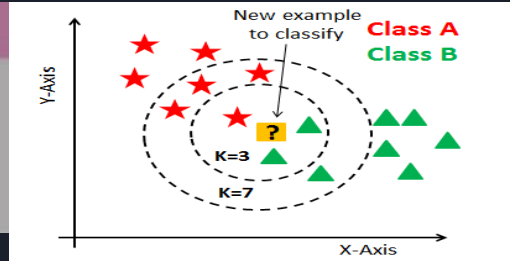
7REC.?Nnot.*?NINSERTED.*?TRAN.*?TTable.*?hostname.*?dbserver1.acme1.com.*?physicalhost.*?esxserver11.acme1.com.*?vmhost.*?app11.acme1.com.*

Examples

REC NOT INSERTED [DB TRAN] Table:hostname=dbserver.acme.com;physicalhost=esxserver1.acme.com;vmhost=app1.acme.com

Distance of K-nearest neighbors (KNN)

Algorithm	Pros	Cons
K-NEAREST NEIGHBOR: K-NN	<ul style="list-style-type: none">• VERY EASY TO UNDERSTAND• GOOD FOR CREATING MODELS THAT INCLUDE NONSTANDARD DATA TYPES SUCH AS TEXT	<ul style="list-style-type: none">• LARGE STORAGE REQUIREMENTS• COMPUTATIONALLY-EXPENSIVE• SENSITIVE TO THE CHOICE OF THE SIMILARITY FUNCTION FOR COMPARING INSTANCES



- Density-based anomaly detection methods
- Continuous data uses Euclidean distance
- Discrete data uses Hamming distance
- Creates models for non-standard data
- It's a supervised algorithm
- Great for fraud detection
- Business and Finance
- Little to no training phase
- Benchmark for other algorithms(Support Vector Machine or Artificial Neural Network)
- Good only for prediction and Quick calculation time



Local Outlier Factor

- Density based method
- Based on this approach, a metric is computed called local outlier factor (lof) for each data point.
 - \hat{LOF} = data point is more outlying
- Computed by taking the average ratio of the local reachability density of a point and its k-nearest neighbors.
- Contamination : proportion of the most isolated points (points that have the highest LOF scores) to be predicted as anomalies.

Pros:

- Well-known and good algorithm for local anomaly detection

Cons:

- Only relies on its direct neighborhood
- Performs poorly on datasets with global anomalies

Osti.gov: Research by IEEE members: an event detection LOF is proposed to detect abnormal events in power systems.

Local Outlier Factor

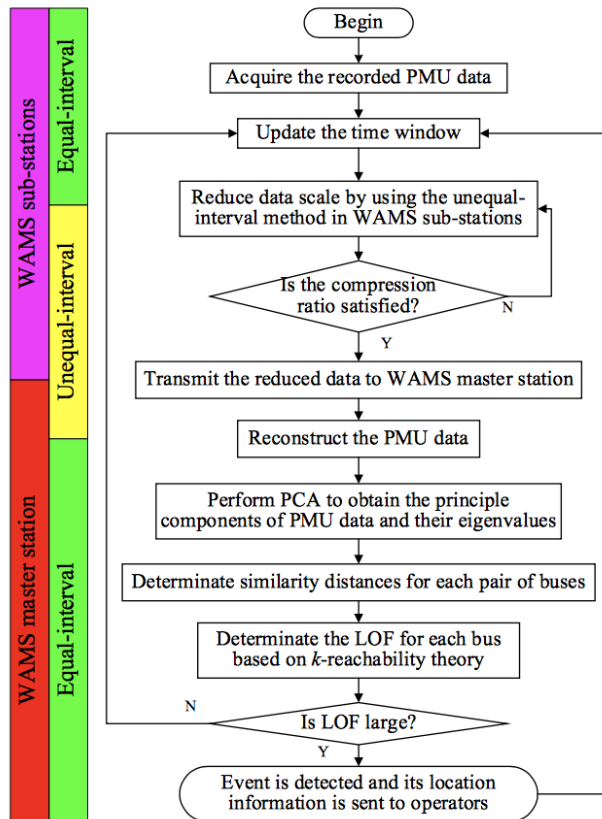


Fig. 5. Flow chart of the proposed SS-LOF algorithm.

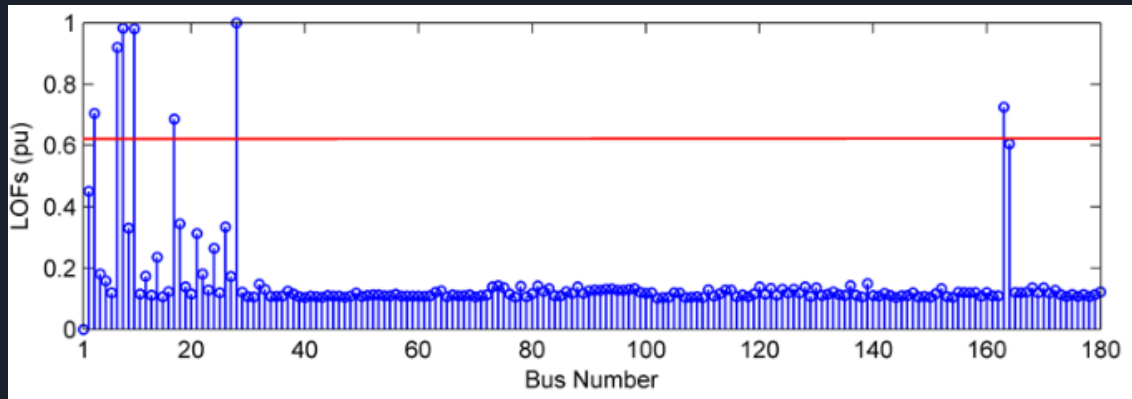


TABLE VIII
COMPARISONS OF THE PROPOSED SS-LOF ALGORITHM WITH THE OTHERS

Functions	VMD [14]	MW-PCA [16]	RQA [21]	MVEE [23]	ST-LOP [35]	Proposed SS-LOF
Event detection	✓	✓	✓	✓	✓	✓
Multiple events		✓	✓	✓		✓
Event location			✓		✓	✓

Support Vector Machine

- Classification via hyperplane
- Finding the optimal hyperplane
- Supervised or unsupervised
- One-class SVM

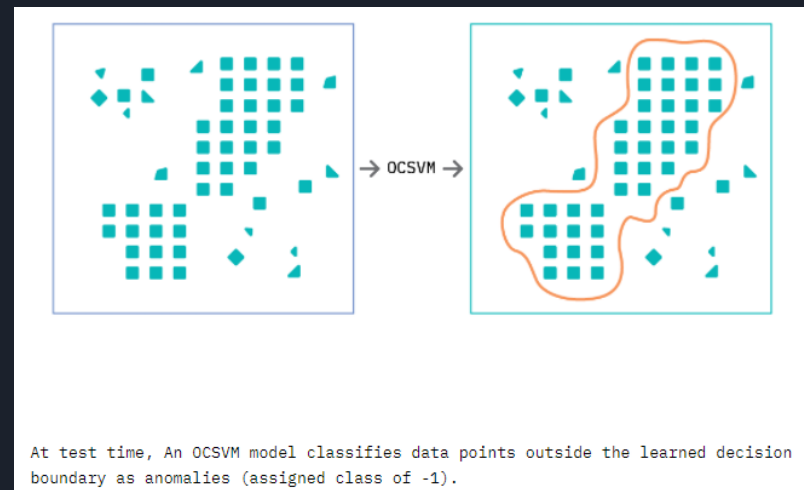


Table 4: Comparing the AUC of SVM based algorithms against other anomaly detection algorithms

Dataset	One-class	Robust one-class	Eta one-class	k-NN	LOF	COF	INFLO	LoOP	Histogram	CBLOF	u-CBLOF	LDcoF
<i>ionosphere</i>	0.9878	0.9956	0.9972	0.9933	0.9178	0.9406	0.9406	0.9211	0.7489	0.3183	0.9822	0.9306
<i>shuttle</i>	0.9936	0.9597	0.9941	0.9208	0.6072	0.5612	0.5303	0.5655	0.9889	0.8700	0.8739	0.5312
<i>breast-cancer</i>	0.9843	0.9734	0.9833	0.9826	0.9916	0.9888	0.9922	0.9882	0.9829	0.8389	0.9743	0.9804
<i>satellite</i>	0.8602	0.8861	0.8544	0.9003	0.8964	0.8708	0.8592	0.8664	0.8862	0.4105	0.9002	0.8657

Autoencoder

	False Positive Rate(%)	Detection Accuracy(%)
k-NN	38.02	88.91
SVM	6.91	92.98
TANN	3.83	96.91
AE	4.09	95.85
CAE	3.44	96.87

- Deep Learning
- Dimensionality reduction
- Reconstruction error as AS
- Noise removal
- Importance of training

