Optimizing Preprocessing & Parameters for Cluster - Based Anomaly Detection

Jeremy Perez & Bethany Danner | REU Mentor Dr. Strnadová-Neeley | Montana State University | August 2019

Abstract

An important component of cyber security is intrusion detection. One way to perform intrusion detection is through anomaly detection: using algorithms to differentiate between normal and abnormal, potentially malicious, network usage in network traffic data.

Many types of anomaly detection algorithms exist. Algorithms such as Local Outlier Factor, DBSCAN, and K-Means use distance measures to identify anomalies, while algorithms such as Isolation Forest use a decision tree to accomplish the same task. Choosing the best algorithm is hard enough; however, each algorithm requires user-input parameters, and often algorithms do not offer straight-forward, objective methods for choosing the best parameter. Thus, in this project, we focused on parameter selection and, after many experiments, verified the best parameters for using anomaly detection algorithms on the NSL-KDD dataset.

However, the network traffic data to be analyzed presents its own set of issues. For example, the IDS 2017 dataset contains missing values that must be filled, and the NSL-KDD dataset has categorical data that must be converted into numerical data before a distance-based clustering algorithm can be implemented. Thus, in this project, we further focused on experimentally evaluating the most effective preprocessing techniques.

Despite focusing on dataset preprocessing and algorithm parameters, the larger goal of this project is evaluating clustering-based anomaly detection techniques, and this project we accomplished by comparing the clustering results K-Means and DBSCAN experiments.

Methods

UNSUPERVISED ANOMALY DETECTION

- vs. Misuse Detection: Anomaly detection has the "theoretical potential" to identify unknown attacks
- vs. Supervised: Unsupervised methods do not require training labels

ALGORITHMS

CLUSTERING:

K-Means (centroid-based)

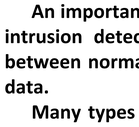
Assumes data clusters are hyper-spherical

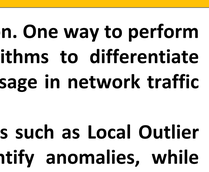
Parameters: user specifies numbers of clusters "K"

DBSCAN (density-based)

Density-Based Spatial Clustering of Applications with Noise

Parameters: user specifies epsilon (esp) and minimum samples (ms)





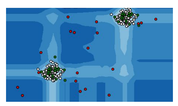
NON-CLUSTERING COMPARISON:

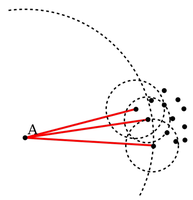
Isolation Forest (IF)

Generates a "forest" of decision trees

Local Outlier Factor (LOF)

Compares distances to K-Nearest Neighbor for each point





Experimental Evaluation

DATASETS

NSL-KDD: An "effective benchmark dataset" [1] and an improved version of KDD 99

Snapshot of the dataset:

	1	3	4	5	...	42	43
Instance	Duration	Service	Flag	Src_bytes	...	Difficulty Label	Attack
1	0	ftp_data	SF	491	...	20	normal
2	0	private	S0	0	...	19	neptune
3	25950	private	RSTR	1	...	15	portsweep
4	0	ftp_data	SF	334	...	11	warezclient
...	...	...	...	...	...	...	...
125973	0	ftp_data	SF	151	...	21	normal

Attack Class	Attack Type
DoS	Back, Land, Neptune, Pod, Smurf,Teardrop,Apache2, Udpstorm, Processtable, Worm
Probe	Satan, Ipsweep, Nmap, Portsweep, Mscan, Saint
R2L	Guess_Password, Ftp_write, Imap, Phf, Multihop, Warezmaster, Warezclient, Spy, Xlock, Xsnoop, Snmpguess, Snmpgetattack, Httptunnel, Sendmail, Named
U2R	Buffer_overflow, Loadmodule, Rootkit, Perl, Sqlattack, Xterm, Ps

IDS 2017

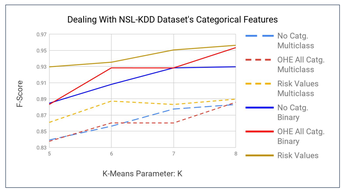
Contains "most up-to-date common attacks" [1]

Snapshot of the dataset:

	1	2	3	4	...	79
Instance	Destination Port	Flow Duration	Total Fwd Packets	Min Packet Length	...	Label
1	80	38308	1	6	...	BENIGN
2	80	3	2	0	...	DoS Hulk
3	80	5011127	8	0	...	DoS GoldenEye
...	...	...	...	...	...	...
692703	80	99999734	2	0	...	DoS slowloris

PREPROCESSING

- Scale data
- Remove true labels & difficulty level from data
- Binary & multiclass-encode attacks
- Deal with missing data
- Deal with categorical data



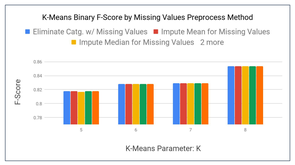
EVALUATION METRICS

F-Score: harmonic mean of precision and recall

Binary: Normal vs. Abnormal

Multiclass: Normal vs. 3-4 types of attacks

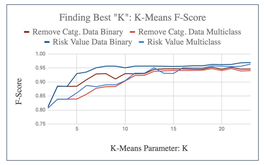
Run Time: time in seconds taken for each algorithm to process the data

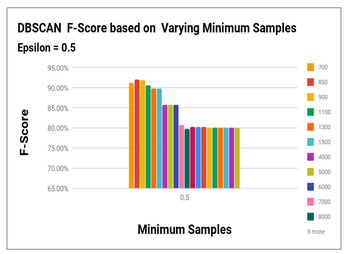


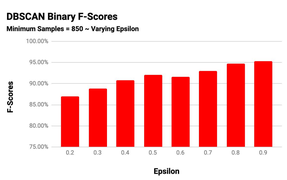
PARAMETER SWEEP

K-Means: F-Score (compared with Inertia)

DBSCAN: F-Score







ANOMALY DETECTION ALGORITHM COMPARISON

Comparison of F-Scores and Run Times: Best Algorithm Parameters

Dataset	Algorithm	Parameter(s)	Binary F-Score	Multiclass F-Score	Run Time (seconds)
NSL-KDD	K-Means	K = 8	95.63%	88.98%	**0.0000188**
	DBSCAN	Esp = 0.8 ~ MS = 650	**96.19%**	**92.09%**	214.2
	LOF	Contamination = 0.3	54.19%	53.46%	259.8
	ISOForest	Contaminatin = 0.25	63.66%	55.79%	24.3

Conclusion

After extensively testing K-Means and DBSCAN parameters, we found that the best F-Scores can be obtained by using K = 8 (K-Means) and eps = 0.8 with ms = 650 (DBSCAN). The respective F-Scores are 95.63% and 96.19%. We ran a parameter sweep on K-Means and found an F-Score asymptote as K increases beginning at K = 8. We further found that the K-Means F-Score plot results aligned with the popular K-Means sum of squares (elbow plot) results. Because of this, we are confident in using K = 8 for further tests on the NSL-KDD Dataset. However, because DBSCAN requires two parameters, one of which is boundless (namely minimum samples), we were unable to test every possible parameter combination. Thus, there possibly exists other parameters that would produce higher F-Scores and better clustering for DBSCAN.

The preprocessing experiment on the NSL-KDD dataset, containing 7.3% of categorical data, showed that the best F-Score can be obtained by encoding categorical data with the risk values given in [5]; thus, our experiment verified the risk value experiment conducted by [5]. We also ran the missing values experiment on the IDS 2017 dataset, in which we found that whether one drops the categories with missing values entirely or imputes a statistically relevant value, the F-Score remains unaffected. However, only 0.0047% of the dataset contains missing values; thus, the missing values affect the IDS dataset much less than the categorical values affect the NSL-KDD dataset.

Finally, we compared clustering algorithms K-Means and DBSCAN with non-clustering algorithms Local Outlier Factor and Isolation Forest and found the clustering algorithms to produce significantly higher F-Scores. K-Means proved to be significantly faster than DBSCAN, with an average run time of 0.0000188 seconds compared respectively to 214.2 seconds, as expected due to the complexity of the respective algorithms. As mentioned above, the best F-Score from DBSCAN is 96.19% and from K-Means is 95.63%, showing that DBSCAN is slightly more accurate than K-Means. While DBSCAN and K-Means obtained F-Scores above 95%, Local Outlier Factor and Isolation Forest both obtained F-Scores below 60%. Because DBSCAN and K-Means are both clustering algorithms, these results favoring DBSCAN and K-Means indicate a large potential for clustering algorithms in network data analysis applications.

Future Work

- Test best parameters on IDS 2017 for K-Means, DBSCAN
- Test missing value techniques on datasets with more missing values
- Dimensionality Reduction
  - Feature Selection: Low Variance, High Correlation, etc.
  - PCA & other true dimension reduction techniques

Acknowledgements

Special thanks to Dr. Veronika Neeley for mentoring us throughout this project, and for Dr. Clem Izurieta for organizing the REU program at Montana State University. This work was funded by the NSF.

[1]Canadian Institute for Cyber Security. University of New Brunswick.
[2]Pedregosa *et al.* (2011). Scikit-learn: Machine Learning in Python.
[3]Jones E, et al. (2001). SciPy: Open Source Scientific Tools for Python.
[4]Dhanabal and Shantharajah. (2015). *International Journal of Adv. Research in C&CE.*
[5]Juanchaiyaphum, et. al. (2014). *Journal of Theo. & Applied Information Technology.*
Images courtesy of: Geeks for Geeks / MathWorks / PRACTICE 2 CODE / aws / Turi Machine Learning

NSF

MONTANA STATE UNIVERSITY