Syntax Literate: Jurnal Ilmiah Indonesia p–ISSN: 2541-0849
e-ISSN: 2548-1398
Vol. 8, No. 6, Juni 2023

PARETO PRINCIPLE TO IMPROVE ANOMALY DETECTION ON SOFTWARE ASSET MANAGEMENT

Aa Iksan Aripin, Antoni Wibowo
Binus Graduate Program
Email: aa.aripin@binus.ac.id, anwibowo@binus.edu

Abstract

Software Asset Management (SAM) is essential for a large company with a centralized software distribution system. Unfortunately, the operationalization of SAM has various problems. These problems become even more complicated when it comes to managing big data. This research proposes the Pareto Principle to reduce data dimensions to solve the problem of large data sizes without losing dataset characteristics before conducting anomaly detection. This anomaly detection is mandatory to identify and reduce invalid data due to misalignment or misclassification. Therefore, this study compares the state-of-arts anomaly detection algorithms: I-forest, KNN, and SVM. As a result, we found that SVM is the best algorithm, with an accuracy rate of 78.4%. In addition, using Pareto for the total population and software name variation effectively reduces the number of observations to 20% data instances with only 16.5% features without compromising the dataset characteristics. In the best algorithm based on experimental results, the use of Pareto increases accuracy by 3.2% and processing time efficiency by 20%.

Keywords: *Anomaly Detection, Data Reduction, Machine Learning, Pareto Principle, Software Asset Management (SAM)*

Introduction

Software Asset Management (SAM) plays an essential role in the cost efficiency and effectiveness of Information Technology (IT). Operations and maintenance of software are complex in a large company that has a large number of computers and employees (Thompson, 2015). The IT budgeting requires valid software license data and a valid usage requirement. Ultimately, the outcome is a proper capacity planning of software licenses and maintenance.

Unfortunately, the operationalization of SAM has various problems. One of the crucial ones is the misalignment of software distribution with organizational functions, indicating inefficiency and unwanted cost potential (Vion, 2018). Classifying the organization based on installed software as its features can identify this misalignment. Removing this misalignment has benefits such as improving asset recording consistency, reducing the complexity of internal service support, and other positive outcomes. This research aims to identify and mitigate this misalignment through anomaly detection.

However, the amount of data generated from the SAM process poses problems (Nouh, 2016a). It needs a data reduction process without eliminating the relevance of the data

How to cite:	Aa Iksan Aripin, Antoni Wibowo (2023) Pareto Principle To Improve Anomaly Detection On Software Asset Management, *Volume 8 Issue 6, Juni 2023*
E-ISSN:	2548-1398
Published by:	Ridwan Institute

(Hamidzadeh et al., 2020a). Usually, most business software distinguishes organization function. Figure 1 Software based on its usage category shows the typical distribution of software based on its functions in a computer system.

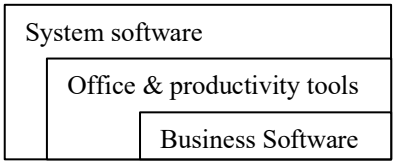

Figure 1 Software Based on Its Usage Category

By looking at the above typical software distribution, using random sampling to reduce data size may pose a risk of losing dataset characteristics. This research assumes that the business software category is the one that dominantly affects organization function classification. The Pareto principle is a well-known method for identifying the dominant cause of the problem. Developed from the thoughts of Vilfredo Pareto on income distribution (Pareto, 1897), this principle is generalized as an approach that assumes that a small portion (20%) of the population will determine the majority (80%) of the impact on the entire population (Dunford, 2014). Thus, this research will be carried out based on the Pareto principle to find the variety of software that is the dominant cause of the organization function classification.

Following are several key takeaways from previous research that are relevant to anomaly detection using datasets related to software or IT asset and the practice of using Pareto principles in Data Mining / Machine Learning:

1. Organizations widely use data mining to improve IT services performance. Anomaly detection application is in supervised, unsupervised, and semi-supervised techniques. The problem domain spans from IT incident tickets to determine the quality of SLA to log analysis for predicting IT equipment maintenance. All of these applications intend to increase the maturity of IT services for the better. However, no research utilizes software distribution datasets in the problem domain of anomaly detection (Nguyen et al., 2019a; Taha & Hadi, 2019).
2. The choice of anomaly detection technique is highly dependent on the dataset's characteristics and the processing's purpose. Researchers agree that, to date, there is no best anomaly detection technique for all datasets. Each technique has its advantages and disadvantages when faced with different datasets. Anomaly detection on artificial datasets may produce different performances and accuracy on real-life datasets. This difference results from noise, uncertainty, size, and complexity in real-life data (Li et al., 2018; Nguyen et al., 2019b).
3. Anomaly detection in categorical data highly depends on the similarity measure approach. The selection of similarity measures depends on the required information from the dataset (Hu et al., 2019; Pang et al., 2016).
4. Many researchers use the Pareto principle and its generalizations to solve MOP (Multi-objective Optimization Problems). The goal is to reduce the data space while maintaining the statistical characteristics of the data that will affect the model's performance. This method is a common thing to do in the context of data reduction. Some researchers even

recommend using Pareto as an initial step before processing big data because it can significantly reduce the computational inefficiency of the model (Hamidzadeh et al., 2020b; Luo et al., 2019; Spolaôr et al., 2017).

5. Using Pareto will not necessarily improve the model's performance in the context of its accuracy. Some researchers state that using Pareto is more focused on reducing the data space so that relatively similar accuracy is good enough (Gallego et al., 2018; Safari et al., 2018a; Tao et al., 2020).
6. Generally, anomaly detection from previous researchers who used IT asset data used a random sampling technique (Hubballi & Dogra, 2016a; Sokolova et al., 2017a; Thangamani et al., 2017a). This research proposes to reduce initial data for anomaly detection models using the Pareto principle, especially in software distribution datasets.

Figure 2 Research Contribution to SAM's Cycle shows that each study has a specific contribution from the SAM cycle's point of view (Nouh, 2016b). The previous researches were on distributing (Sokolova et al., 2017b; Wang et al., 2019), maintaining (Thangamani et al., 2017b), and monitoring (Hubballi & Dogra, 2016b; Landthaler et al., 2018) phase in the SAM cycle. Meanwhile, this research is on procuring, distributing, and disposing phase. Then the question arises:

1. Can the Pareto principle be applied in selecting and reducing data for anomaly detection in this dataset?
2. What is the best state-of-art anomaly detection algorithm for this dataset?
3. How does the Pareto influence anomaly detection performance in this dataset?

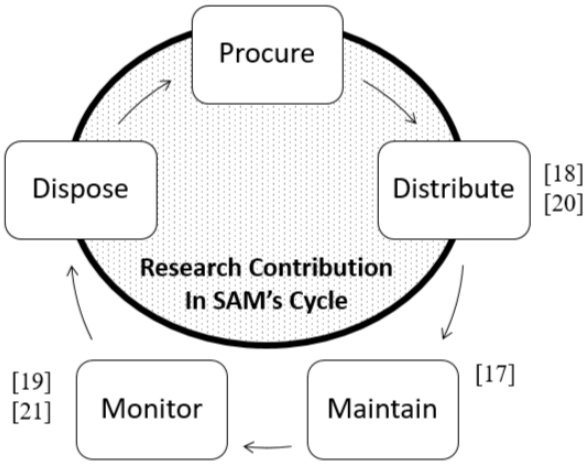

Figure 2 Research Contribution to SAM's Cycle

Method

The object of this research is software distribution data in XYZ company. We obtained the data from Microsoft System Center Configuration Manager™ (SCCM) and joined it with employee placement data from the Human Resource department. This dataset was normalized to eliminate naming errors or redundancies. Table 1 Data Attributes shows the data attributes used in this research.

Table 1 Data Attributes

Attribute	column	Use as

Computer Name	*pc*	Index
Organization	*dvs*	Classification target
Software name	*sw*	Features

The raw data used in this study is a dataset containing the software distribution at company XYZ within five years (2015-2019). This raw data is then mapped/tuned and normalized. There are 12,416 software names in the raw data. This data is then mapped manually into 4426 software names. The resulting dataset is a software package or suite that represents the software and has also been able to reduce standard and redundant software. Table 2 Software Mapping shows an illustration of this mapping process.

Table 2 Software Mapping

Raw	**Tune**
AutoCAD 2012 - English	*AutoCAD*
AutoCAD 2012 - English SP1	*AutoCAD*
AutoCAD 2012 Language Pack - English	*AutoCAD*
Autodesk Inventor Fusion plug-in language pack for AutoCAD 2012	*Inventor Fusion*
Autodesk Inventor Fusion plug-in for AutoCAD 2012	*Inventor Fusion*
Microsoft Office Professional Plus 2016	*Microsoft Office*
Microsoft Office ?? ? 2016 - ???	*Microsoft Office*
Contiki ECM	*Contiki*
ContikiManager-5.2-EN-SK-01.01	*Contiki*

A total of 1109 software names resulted from this manual mapping. However, this assumption is not necessarily correct. So in the dataset exploration process, these assumptions will be tested and then readjusted if necessary.

Data reduction by the Pareto principle is trying to reduce the data dimensions. Thus, working only with the collection of software that has the most influence on installation variations. The resulting dataset (Pareto data) is an input for the algorithm selection process. The evaluation focuses on accuracy and processing time. The best algorithm has the best processing time without reducing accuracy based on the best hyperparameter tuning results. The results of Pareto data processing are compared with the result of initial data (dataset prior to Pareto reduction) to get the Pareto effect on processing time and accuracy.

Following are the stages of creating Pareto data:

1. Identifying whether the data distribution each year follows the Pareto distribution.
2. Testing with statistical tests.
3. Grouping of each software using the quartiles of its frequency distribution and color rules as shown in Table 3 Data Categorization Based on Quartile. This method is also used by (Walkinshaw & Minku, 2018) in grouping program code files to detect defects/bugs. However,

they depict the distribution using a box plot per quartiles, and this study uses a histogram with color as a visual differentiator.

Table 3 Data Categorization Based on Quartile

Category	Percentage	Color
Common	100% - 75%	Grey
Often	75% - 50%	Blue
Seldom	50% - 25%	Green
Sparse	25% - 1%	Orange
Rare	= < 1%	Red

4. Performing minimal–redundancy–maximal –relevance analysis (Peng et al., 2005) to get how much influence each software name (features) has on the installation variations that occurred per organization's function (classification target). What will be seen is whether there is an anomaly in each organization's function. The redundancy and relevance score determines the magnitude of the influence. This study calculates all features to get an overall MRMR score, and the Pareto principle will decide the selection of features.
5. Create Pareto Diagram from MRMR result to assess and select dominant data as Pareto data.

Results and Discussion

The conclusions of data property exploration for five years (2015-2019) are:

1. No null values exist.
2. No imputation step is required.
3. All data can be considered categorical data.
4. The initial data dimension is 244,082 rows with information for five years contained; 24 organizational functions, an average of 2945 computers per year, and 2945 types of software for further processing. Normalized data is available online at [24].

The ground truth survey is using 2019's data. Figure 3 2019's Data Histogram shows the histogram plot of this data.

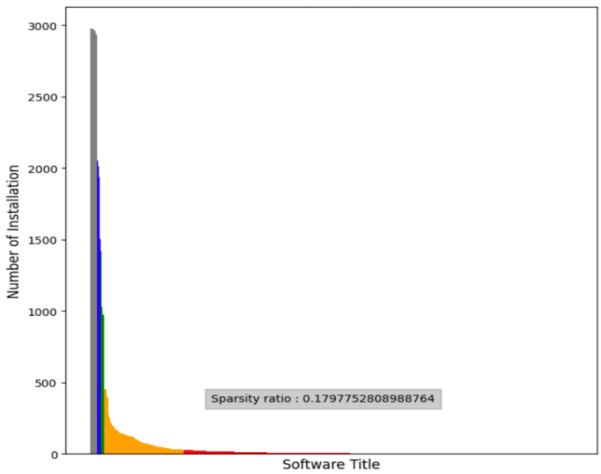

Figure 3 2019's Data Histogram

The histogram depiction of the frequency distribution data shows that the distribution graph has the characteristics of a Heavy-Tail Distribution or can be assumed to be statistically similar to the Pareto distribution. Figure 4 Distribution Fitting Evaluation shows the fitting result. The data can fit with the Pareto distribution model. The raw data has metadata as follows; the number of software names = 534, the number of computers = 2977, and the number of functions = 24.

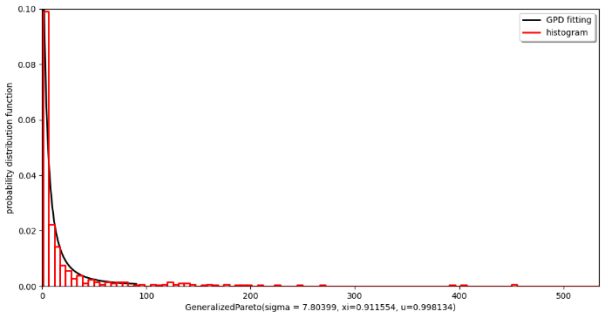

Figure 4 Distribution Fitting Evaluation

Table 4 Frequency Category shows that software with the "rare" frequency category has the most software types, followed by the "sparse" frequency category. Meanwhile, the frequency of the "common," "often," and "seldom" categories is minimal in terms of the number of software types.

Table 4 Frequency Category

Category	Number of Sw	Ratio	Freq.	Ratio Freq.
Common	8	1.50%	23693	49.72%
Often	4	0.75%	7502	15.74%
Seldom	4	0.75%	4396	9.23%
Sparse	88	16.48%	9400	19.73%
Rare	430	80.52%	2659	5.58%
Total	534	100%	47650	100%

Then an MRMR analysis is carried out to get the magnitude of the effect of each type of software, with the classification target being an organization function. The scoring results are shown in Table 5 MRMR Score.

Table 5 MRMR Score

Rank	Software Name	Category	MRMR Score
1	*Petrel*	*sparse*	167903
2	*WellView*	*sparse*	49846
3	*WRFView*	*sparse*	46434
4	*Brava Desktop*	*sparse*	42138
5	*Contiki*	*sparse*	35617
...	...	...	…
533	*Vbsedit*	*rare*	14.216
534	*skillpipe Reader*	*rare*	13.546

Table 6 Calculation Table for Pareto Diagram shows the cumulative and its percentage of MRMR score per category. Figure 5 Pareto Diagram Results show its Pareto diagram with an 80/20 rule threshold.

Table 6 Calculation Table for Pareto Diagram

Category	Total MRMR Score	Cumulative	Cumulative %
sparse	657357.544	657357.544	73.88%
rare	218301.601	875659.146	98.42%
common	9511.326	885170.472	99.49%
seldom	3237.326	888407.798	99.85%
Often	1343.307	889751.106	100.00%

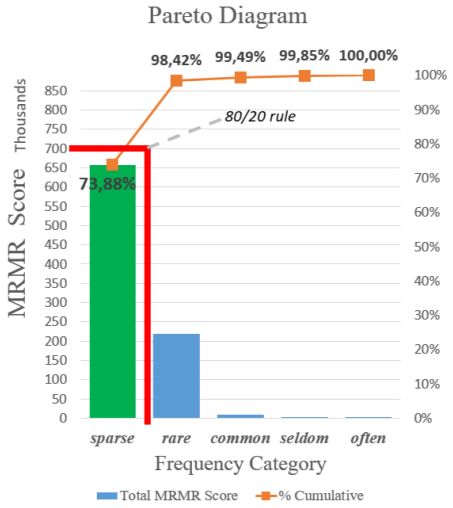

Figure 5 Pareto Diagram Result

According to the diagram, the data group in the most influential category (far left) is the "sparse" frequency category, which has an influence percentage of 73.88%. As shown in Table 4 Frequency Categories, this category has a ratio of software types of 16.48%. So it can be concluded that meeting the Pareto 80/20 principle, where in this case, the "sparse" category with the type of software 16.48% of all types of software in the initial data affect 73.88% of the variation/combination of software installations in the dataset according to the (classification target) organization function. Following the Pareto principle, software with the "sparse" category is the most relevant data. At the same time, this also confirms the first research question that the Pareto principle applies.

Figure 6 Relevant Subspace Frequency Distribution shows the overall distribution of “sparse” categories. With the application of Pareto as described previously, it is possible to select a subspace that determines the classification variation, which is only for software with the “sparse” category. Thus it will reduce the features of anomaly detection processing. This selection will also reduce the number of computers (data instance) with no features in the group "sparse."

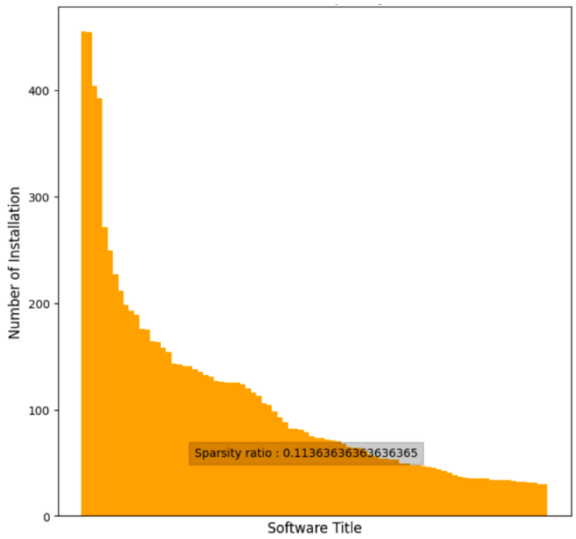

Figure 6 Relevant Subspace Frequency Distribution

We have the relevant subspace for the modeling process up to this stage. The Pareto approach has reduced more than 80% of the data instance from the original 47,650 rows to 9,400 (19.73%). Thus the manual process for labeling anomaly data can work with 20% of the initial data and with 88 (16.48%) software titles from the previous 534 software titles as features.

Ground Truth Survey

The labeling process is manual. The binary number 1 in the software column indicates that the software exists on a particular computing device. While the binary number 0 indicates the opposite. An anomaly label at each line indicates a computer and organization's function mapped to the type of software considered anomalous. This process is carried out with an iterative approach until a consensus achieves between three domain experts. The result of this iterative process is a complete mapping of the existence of software for each computer, complete with descriptions of the anomalies.

Modeling

This research is semi-supervised because the modeling uses an unsupervised approach, while the test uses a supervised approach with ground truth comparisons resulting from the survey. There are three candidate algorithms chosen for modeling, namely Isolation Forest (i-forest), K-Nearest Neighbor (KNN), and Support Vector Machine (SVM).

The i-forest algorithm hyperparameter is the n-estimator parameter. The maximum samples parameter is "auto," considering the variation in the amount of data in each organization's function varies greatly. The n-estimator values in this study are 3, 4, 5, 6, 7, 8, 10, 15, 20, and 30.

The KNN algorithm hyperparameter variation is K. The hyperparameter tuning for the K value is { 2, 3, 4, and 5}. These values are selected considering that the K parameter indicates the number of neighbors formed in the modeling. Given that each organization's function has different records, the highest K value follows the smallest computer population in one organization function, which is 5.

The SVM algorithm hyperparameter is the Gamma parameter and the coefficient (c). The gamma value is "auto." Values with auto settings are equivalent to 1/ n-features. The value of c used in this study is c = { 0.01, 0.1, 1, 10, 100}. The value of c is set by referring to the standard used by previous studies.

The result with the best combination of hyperparameters for each tested algorithm with non-Pareto (initial data) and Pareto data is shown in Tables 7 Results Using Non-Pareto Data and Table 8 Result Using Pareto Data, respectively.

Table 7 Results Using Non-Pareto Data

Algorithm	***Hyper Parameter***	***Accuracy (%)***	***Time (s)***
Iforest	30	75,281%	15,391
KNN	3	78,889%	5,141
SVM	0,01	75,190%	3,969
Average		76,343%	

Table 8 Result Using Pareto Data

Algorithm	***Hyper Parameter***	***Accuracy (%)***	***Time (s)***
Iforest	30	77,955%	14,219
KNN	3	77,485%	3,781
SVM	0,01	78,384%	3,156
Average		77,956%	

The above results show that Pareto implementation impacts the result of accuracy and processing time. The impact shows that Pareto can solve the MOP problem for the I-forest and SVM algorithms on the software dataset but not for the KNN algorithm. The highest accuracy increase is in the SVM algorithm, which is 3.194%. In comparison, the highest processing time efficiency is in the KNN algorithm at 26.444%. The test results on the KNN algorithm prove that Pareto does not constantly improve detection accuracy. In fact, in this case, there is a decrease in accuracy of 1.40%. Overall, because this study tries to answer the MOP problem, it is concluded that the SVM algorithm is the best algorithm based on processing time and accuracy.

Discussion

In scoring calculation with MRMR, the software ranking order is not always correct according to the frequency category when viewed according to its ranking. This overlap is due to the low average correlation (low redundancy). So it has a reasonably high MRMR score even though the relevance is also low. Shifting the grouping of frequency categories based on quartiles may minimize this overlap. For example, the outlier filter 1.5 * (Q3 – Q1) is a typical formula in data mining. If applied to this dataset, it will produce a minimum frequency limit of 1.5 * (19-1) = 27. The “sparse” category with the smallest frequency value for the current grouping will become 30. Research related to the determination of the Pareto tail may optimize this result (Safari et al., 2018b).

We also found "common" overlap with "sparse." Filtering the "relevance" results may anticipate this overlap, for example, by the standard deviation of the scoring results as shown in Table 9 Relevance Scoring Characteristics. This filter results in the minimum "relevance" value of 21.1.

Table 9 *Relevance* Scoring Characteristics

mean	***std***	***min***	***25%***	***50%***	***75%***	***max***
5,554	15,6	0,138	0,54	1,636	4,014	167,903

Based on Table 10 Pareto VS Non-Pareto Method Accuracy and Processing Time, the processing speed increase in I-forest is the most minor compared to others. Overall, with initial and Pareto data, i-forest takes the longest time. The accuracy of KNN decreases. This accuracy can be caused by taking the average of several tests using $K \in \{2 \ldots 5\}$. At $K \neq 3$, KNN has increased accuracy from processing initial data to Pareto data. However, at K=3, where K=3 is the best hyperparameter (on initial data), the accuracy decreases in Pareto data. The best hyperparameter for KNN on Pareto data is K=2.

Table 10 Pareto VS Non-Pareto Method Accuracy and Processing Time

Algorithm	**Accuracy (%)**	**Time (s)**
Alg KNN	-1,404250%	26,44377%
Alg I-forest	2,6735045%	7,614213%
Alg SVM	3,1937095%	20,47244%

Conclusion

The conclusions from this research are as follows:

1. The Pareto principle successfully reduces initial data. The similarity of data distribution between the initial data to the Pareto data and the performance results of the model based on the Pareto data confirm that data characteristic is identical.
2. The best anomaly detection algorithm for the software asset dataset in this research is the SVM algorithm, with an accuracy rate of 78.384% on Pareto data.
3. The use of Pareto is effective in reducing the number of observations. In the best algorithm based on experimental results, Pareto can increase accuracy by 3.194% and increase processing time efficiency by 20.472%.

References

Dunford, R. (2014). The Pareto Principle. *The Plymouth Student Scientist*, *7*(1), 140–148. Https://Doi.Org/10.1016/J.Jacr.2018.02.026

Gallego, A. J., Calvo-Zaragoza, J., Valero-Mas, J. J., & Rico-Juan, J. R. (2018). Clustering-Based K-Nearest Neighbor Classification For Large-Scale Data With Neural Codes

Representation. *Pattern Recognition*, *74*, 531–543. Https://Doi.Org/10.1016/J.Patcog.2017.09.038

Hamidzadeh, J., Kashefi, N., & Moradi, M. (2020a). Combined Weighted Multi-Objective Optimizer For Instance Reduction In Two-Class Imbalanced Data Problem. *Engineering Applications Of Artificial Intelligence*, *90*(January), 103500. Https://Doi.Org/10.1016/J.Engappai.2020.103500

Hamidzadeh, J., Kashefi, N., & Moradi, M. (2020b). Combined Weighted Multi-Objective Optimizer For Instance Reduction In Two-Class Imbalanced Data Problem. *Engineering Applications Of Artificial Intelligence*, *90*(January), 103500. Https://Doi.Org/10.1016/J.Engappai.2020.103500

Hu, X., Wang, Y., & Cheng, L. (2019). Multi-Hierarchy Attribute Relationship Mining Based Outlier Detection For Categorical Data. *Proceedings Of The International Joint Conference On Neural Networks*, *2019-July*(July), 1–8. Https://Doi.Org/10.1109/Ijcnn.2019.8852383

Hubballi, N., & Dogra, H. (2016a). Detecting Packed Executable File : Supervised Or Anomaly Detection Method ? *2016 11th International Conference On Availability, Reliability And Security*, 638–643. Https://Doi.Org/10.1109/Ares.2016.18

Hubballi, N., & Dogra, H. (2016b). Detecting Packed Executable File : Supervised Or Anomaly Detection Method ? *2016 11th International Conference On Availability, Reliability And Security*, 638–643. Https://Doi.Org/10.1109/Ares.2016.18

Landthaler, J., Uludag, O., Bondel, G., Elnaggar, A., Nair, S., & Matthes, F. (2018). A Machine Learning Based Approach To Application Landscape Documentation. In R. A. B. K. Kirikova (Ed.), *Ifip Working Conference On The Practice Of Enterprise Modeling* (Vol. 335, Issue 2019, Pp. 71–85). Springer Nature Switzerland Ag. Https://Doi.Org/10.1007/978-3-030-02302-7_5

Li, J., Zhang, J., Pang, N., & Qin, X. (2018). Weighted Outlier Detection Of High-Dimensional Categorical Data Using Feature Grouping. *Ieee Transactions On Systems, Man, And Cybernetics: Systems*, *Pp*, 1–14. Https://Doi.Org/10.1109/Tsmc.2018.2847625

Luo, J., Jiao, L., Liu, F., Yang, S., & Ma, W. (2019). A Pareto-Based Sparse Subspace Learning Framework. *Ieee Transactions On Cybernetics*, *49*(11), 3859–3872. Https://Doi.Org/10.1109/Tcyb.2018.2849442

Nguyen, T. H. T., Dinh, D. T., Sriboonchitta, S., & Huynh, V. N. (2019a). A Method For K-Means-Like Clustering Of Categorical Data. *Journal Of Ambient Intelligence And Humanized Computing*, *Berkhin 2002*. Https://Doi.Org/10.1007/S12652-019-01445-5

Nguyen, T. H. T., Dinh, D. T., Sriboonchitta, S., & Huynh, V. N. (2019b). A Method For K-Means-Like Clustering Of Categorical Data. *Journal Of Ambient Intelligence And Humanized Computing*, *Berkhin 2002*. Https://Doi.Org/10.1007/S12652-019-01445-5

Nouh, F. (2016a). *Sam Software Asset Management*. *January*, 0–24.

Nouh, F. (2016b). *Sam Software Asset Management*. *January*, 0–24.

Pang, G., Cao, L., & Chen, L. (2016). Outlier Detection In Complex Categorical Data By Modeling The Feature Value Couplings. *Ijcai International Joint Conference On Artificial Intelligence*, *2016-Janua*, 1902–1908.

Pareto, V. (1897). The New Theories Of Economics. *Journal Of Political Economy*, *5*(4), 485–502.

Peng, H., Long, F., & Ding, C. (2005). Feature Selection Based On Mutual Information: Criteria Of Max-Dependency, Max-Relevance, And Min-Redundancy. *Ieee Transactions On Pattern Analysis And Machine Intelligence*, *27*(8), 1226–1238. Https://Doi.Org/10.1109/Tpami.2005.159

Safari, M. A. M., Masseran, N., & Ibrahim, K. (2018a). Optimal Threshold For Pareto Tail Modeling In The Presence Of Outliers. *Physica A: Statistical Mechanics And Its Applications*, *509*, 169–180. Https://Doi.Org/10.1016/J.Physa.2018.06.007

Safari, M. A. M., Masseran, N., & Ibrahim, K. (2018b). Optimal Threshold For Pareto Tail Modeling In The Presence Of Outliers. *Physica A: Statistical Mechanics And Its Applications*, *509*, 169–180. Https://Doi.Org/10.1016/J.Physa.2018.06.007

Sokolova, K., Perez, C., & Lemercier, M. (2017a). Android Application Classification And Anomaly Detection With Graph-Based Permission Patterns. *Decision Support Systems*, *93*, 62–76. Https://Doi.Org/10.1016/J.Dss.2016.09.006

Sokolova, K., Perez, C., & Lemercier, M. (2017b). Android Application Classification And Anomaly Detection With Graph-Based Permission Patterns. *Decision Support Systems*, *93*, 62–76. Https://Doi.Org/10.1016/J.Dss.2016.09.006

Spolaôr, N., Lorena, A. C., & Diana Lee, H. (2017). Feature Selection Via Pareto Multi-Objective Genetic Algorithms. *Applied Artificial Intelligence*, *31*(9–10), 764–791. Https://Doi.Org/10.1080/08839514.2018.1444334

Taha, A., & Hadi, A. S. (2019). Anomaly Detection Methods For Categorical Data: A Review. *Acm Computing Surveys*, *52*(2). Https://Doi.Org/10.1145/3312739

Tao, R., Gong, Z., Ma, Q., & Thill, J. C. (2020). Boosting Computational Effectiveness In Big Spatial Flow Data Analysis With Intelligent Data Reduction. *Isprs International Journal Of Geo-Information*, *9*(5). Https://Doi.Org/10.3390/Ijgi9050299

Thangamani, A., Nitta, B., Day, C., Shah, D., & Aggarwal, N. (2017a). *Anomaly Analysis For Software Distribution* (Patent No. Us 9,626,277 B2). United States Patent Application Publication.

Thangamani, A., Nitta, B., Day, C., Shah, D., & Aggarwal, N. (2017b). *Anomaly Analysis For Software Distribution* (Patent No. Us 9,626,277 B2). United States Patent Application Publication.

Thompson, M. (2015). *An Introduction To Software Asset Management* (Issue January).

Vion, A. (2018). *Software Asset Management And Cloud Computing*.

Walkinshaw, N., & Minku, L. (2018). Are 20% Of Files Responsible For 80% Of Defects? *International Symposium On Empirical Software Engineering And Measurement*. Https://Doi.Org/10.1145/3239235.3239244

Wang, H., Yang, W., Nakagawa, N., Gelman, D., & Holland, M. (2019). *System And Method For Detecting Fraudulent Software Installation Activity* (Patent No. Us 2019 / 0102545 A1). United States Patent Application Publication.