CHALLENGES AND SOLUTIONS OF REAL-TIME CLUSTERING FOR NETWORK ANOMALY DETECTION

Jagatheesan Kunasaikaran¹, Roslan Ismail², Abdul Rahim Ahmad³

¹Software Engineering Program, Department of Computing, University Tenaga Nasional, Malaysia.

²Associate Professor, Software Engineering Program, Department of Computing, University Tenaga Nasional, Malaysia.

³Associate Professor, Systems & Networking Program, Department of Computing, University Tenaga Nasional

Email: ST22489@utn.edu.my, Roslan@uniten.edu.my, Abdrahim@uniten.edu.my

Corresponding author: Roslan Ismail
Email: Roslan@uniten.edu.my

https://doi.org/10.26782/jmcms.2019.12.00078

Abstract

The escalating number of novel network attacks warrants an approach where network data is processed in real-time for anomaly detection. Clustering is one of the foremost unsupervised learning algorithms in this domain that can detect outliers without prior knowledge of the data. However, cluster analysis precludes with it many challenges that need to be overcome for it to be adapted for real-time computation. This research paper outlines these challenges and the possible solutions to mitigate these challenges. We have also explored on a brief overview of clustering algorithms to give a high-level idea of cluster analysis.

Keywords: Clustering methods, Intrusion detection, Network security

I. Introduction

In 2015, the British government conducted a survey in association with PricewaterhouseCoopers (PwC) [1] to understand the nature of cyber security incidents happening in the United Kingdom (UK). The cost sustained on average by a small business is from £75k - £311k and for a larger organization, the cost increases to £1.46m - £3.14m. For the past few decades, researchers have advanced the field of anomaly detection. The main goal of this paper is to study the challenges of applying clustering
which is a popular data mining algorithm to solve real-time anomaly detection. Possible solutions to overcome the challenges are also presented in this paper.

Anomalies are classified as any behavior in a network that is outside the normal behavior of the network traffic. In this process of detecting anomalies, live network traffic is collected and labeled as either normal or anomalous. Then, a model of normal network behavior is trained on this data and the model is deployed to network environment to detect any traffic behavior that is outside the trained model. However, to detect new forms of attacks, the model needs to be retrained to detect the new attacks. Real-time network anomaly detection is heavily researched now to address these shortcomings. In the computing context, real-time is defined by Oxford Dictionary as a system in which input data is processed within milliseconds so that it is available virtually immediately as feedback to the process from which it is coming. One of the ways of achieving real-time network anomaly detection is to use a category of machine learning known as unsupervised learning. In unsupervised learning, inferences are obtained from input data without the need of any labeled response. Cluster analysis is one the most widely used algorithm for unsupervised learning. In cluster analysis, objects that have similar characteristics are grouped together. This feature of cluster-based algorithms enables anomalies to stand out from the normal traffic group that is constructed.

The rest of this paper is organized as follows. Section II provides an overview of clustering algorithm classified by categories. In Section III, the challenges of applying clustering algorithms to detect anomalies in real-time are discussed and the possible solutions to overcome the challenges are also elaborated. This paper concludes with Section IV where a summary of the work done in the paper is given and future research directions are stated.

II. Clustering Algorithms

Cluster analysis involves the process of grouping similar objects into groups which is called as clusters. Cluster analysis is a numerical method for classification that is able to classify a set of data objectively and in a stable manner [2]. The objective here means that when the computation to classify is repeated, the analysis remains the same and it should be stable in a sense that when new characteristics are added to the data points or new data points are added, the classification remains the same.

Clustering based algorithms can be classified into five categories, hierarchical, density, fuzzy, grid, and model based clustering [3]. In hierarchical clustering, clusters are made from successive iteration over the data through agglomerative or divisive methods. In the agglomerative method, clusters are made by joining multiple clusters into bigger clusters. On the other hand, in the divisive method, clusters are made by a
successive division of clusters into smaller clusters. Some well-known algorithms in this category are BIRCH, CURE, and ROCK. Density-based algorithms rely on the structure of data whereby every point in a data space would have a collection of points that are closer to each other which makes the region denser than the others. Criteria such as the number of points that should be in a region for it to qualify as a dense region and then a cluster is formed over these dense regions are the basis for this category of algorithms. DBSCAN which uses the idea of density reachability and density connectivity is an algorithm that is used extensively in this category.

Besides that, fuzzy-based algorithms assign a membership function to each of the data points. This membership function determines whether data point belongs to a cluster or otherwise. GOM is an algorithm that belongs to this category and is widely used. In grid-based algorithms, the objects in the data space are partitioned into grids. This partitioning of data allows for a reduction in the processing time as computation are performed against the characteristics of the grid instead of individual objects in the data space. STING and CLIQUE are commonly used algorithms in this category. Model-based clustering differs from the other clustering methods which lean towards heuristics as this type of clustering is based on describing the subsets of data in terms of some form probability distribution. MCLUST, EM, and COBWEB are some of the widely used algorithms in this category.

III. Challenges and possible solutions

Clustering algorithms in real-time network anomaly detection faces many challenges such as the accuracy of the anomaly detected, the curse of high dimensionality, scalability, the adjustment of parameters and the continuous processing of data. In anomaly detection, the clustering algorithm that is used should be able to handle noises as these noises will give false-positive outliers which will in turn affect the accuracy of the anomaly detected. Besides, the curse of dimensionality also affects clustering algorithms whereby algorithms perform poorly in cases where the dimensionality of the objects in the data space is of high dimensionality. In high dimensionality space, the vectors containing a list of points are orthogonal to each other or the distance between the vectors are almost equally distanced [4]. This causes the formation of the cluster to be difficult as no measure of “closeness” can be satisfied.

As problem domains that are applicable in clustering algorithms are moving towards real-time processing where data that needs to be processed is growing exponentially and the speed of computation needs to be increased in tandem, commonly used clustering algorithms suffers from the inability to scale with these new challenges posed. In addition to that, most types of clustering algorithms are based on heuristics. There are multiple parameters that needs to be adjusted in each algorithm. These adjustment of parameters poses a challenge as it takes multiple trial and error to
determine whether the parameters suffice to create meaningful clusters. To solve the problem domains where data is continuous, clustering algorithms should also be able to consume the data in a continuous fashion and adjust the clusters formed continuously. This is not the case with all widely used clustering algorithms now as these algorithms were mostly developed under the perception that data is aggregated beforehand.

Detection of anomaly

The noise in a data space is identified as anomalies. These noises are characterized as points in the data space that do not belong to any cluster. Algorithms such as k-Means algorithm do not have the notion of noise. All points in the data space belong to a cluster. Hence, it is harder to detect anomalies. As a means to overcome the inherent limitation of algorithms such as k-Means, Density-based spatial clustering of applications with noise (DBSCAN) is a clustering algorithm that can be applied to detect anomalies in the data as this algorithm comes with the notion of noise [5]. Two parameters are defined by the user in DBSCAN, a minimum number of points in a dense region, $M$ and the radius of the dense region, $\varepsilon$. A cluster contains core and non-core objects. Core objects are objects which satisfy the condition of having $M$ objects within the radius $\varepsilon$. For objects that does not satisfy the condition but is density-reachable to core objects, it is known as non-core objects. An object $p$ is density-reachable to $q$ if there is a chain link of objects to $q$ such that all the objects in the chain link are density-reachable to each other. These non-core objects are also included in the cluster. One of the key requirements that DBSCAN intends to solve compared to the other algorithms is to be able to detect clusters with arbitrary shape. This feature of DBSCAN enables it to detect noise separately than other clusters. In [6], the authors introduced a novel algorithm to detect anomalies by setting a threshold that determines whether a behavior is outside the normal behavior. A similarity calculation is done to calculate the difference between the normal and abnormal behavior. If the similarity calculation is more than the set threshold, then the behavior is considered as an anomalous behavior.

Curse of dimensionality

When the number of dimension of the points in a data space increases, four problems arise [7]. Firstly, it will get harder for a complete enumeration of all the data space to be created as visualization of the data space becomes more complex. Common ideas attached to clustering such as proximity, distance or neighborhood also becomes convoluted in meaning. Furthermore, not all the dimension might be relevant to problem domain and these irrelevant dimensions may cause unnecessary noise in the clustering result. The higher number of dimensions also yield a subset of attributes being correlated to one another. Feature reduction that is used on this data sets may result in only a few attributes considered as not redundant and this causes the loss of knowledge on the correlation among the subset of attributes.
One of the technique to solve this problem is to remove dimensions that of no use to the domain of the problem being solved. This reduces the dimensionality of the data. As classified in [8], there are three broad categories of feature reduction methods that can be used. The categories are filter, wrapper, and embedded method. In filtering methods, the data is preprocessed and a ranking of features is generated. On the other hand, in wrapper methods, the criterion for feature selection is based on the performance of the predictor. The last category, embedded methods, make the selection of variables as a phase in the training process without separating the data into training or testing data. Besides that, there are techniques to project the data to a lower dimension. An algorithm such as Principal Component Analysis (PCA) or Singular Value Decomposition (SVD) can do these higher to lower dimension projection. In PCA, it tries to find the direction having the greatest variance in the data and each point is mapped to these directions.

Discrete wavelet transforms (DWT) is another technique that can be used to reduce the dimension of data. DWT works by transforming a time series data into lower dimension by applying a mathematical function to the data. The coefficients of the function that gives the highest energy yield to the function are retained and the other coefficients are dropped [9]. Neural networks can also be used to reduce the dimensionality of the data. By using the neural network, high-dimensional data can be low-dimensional data. A neural network is composed of multiple layers where each layer in the neural network shows a correlation between the activities of the units in the layer below. In [10], the authors proposed a method to pre-train the neural network by automatically finding weights to initialize the neural network with weights that are near to good weight values. The method proposed in the paper shows better in reducing the dimensionality of data compared to PCA.

Scalability

With the advent of Internet, data has been continuously generated and this data is available in an irregular manner. According to a study done by Cisco Systems, 278 Exabytes (EB) of global IP traffic will be generated per month by the year 2021 [11]. Mining this data streams has become a hot topic in recent years. It is concerned with finding structures of knowledge that could be found in models and patterns of the non-stopping streams of data [12]. A multi-step clustering algorithm could also be used to process the data efficiently. In [13], the authors have proposed to use a multi-step clustering algorithm which is split into two steps, coarse-grained clustering, and fine-grained clustering. In the coarse-grained clustering, the BIRCH algorithm is used to cluster the network traffic based on simple statistical features. The clustering of this simple statistical features reduces the complexity of the clustering in the next step which uses hierarchical clustering. In this step, the clusters formed in the first step is further
split into different clusters based on the structure of the HTTP queries. The simplification of the data space after each clustering process speeds up the overall clustering.

An ensemble of machine learning techniques which have cluster analysis as part of the training process is proposed in [14]. In this research, the authors have built a hierarchical cluster using the dynamically growing self-organizing tree (DGSOT) algorithm. Support vector machines (SVM) algorithm is then used to perform operations on the nodes to the computed nodes. SVM is used to reduce the training set and approximate the support vectors. Recent advancements in streaming algorithms also gave rise to software computing platforms such as Apache Storm, Apache Spark and Apache Kafka which can process streams of data in real-time. In [15], the authors have used Apache Kafka to collect the network data and channel the data to the Spark stream processing framework. Then, Spark’s stream processing framework is used for the model training and prediction.

Adjustment of parameters

Clustering algorithms usually need multiple parameters to be adjusted for the clustering to give the desired result. This causes the intrusion detection system administrators to spend a lot time on tuning and adjusting the parameters of the system to optimize the clustering algorithm for anomaly detection. In [16], the authors have proposed a method to automatically split the data space into clusters without the need to adjust the parameters. The proposed uses a splitting method which splits a dense group into different clusters. This is done by calculating the average of data instances in a dense group and this average becomes the cluster center. Then, the average of the data instances in the dense group to the cluster center, is calculated. If the data instances that is the furthest in the dense group is having a distance that is more than , a new cluster center is formed from that data instance. Data instances which are below in distance to the new cluster center are then assigned to this new cluster. Later, the mean of the members in the new cluster replaces the initial center of the new cluster. This process is repeated until no data instances are having a distance more than from its cluster center.

Continuous processing of data

A major challenge in real-time clustering is that data needs be continuously collected and clustering needs to be done continuously to detect anomalies in real time. In [17], the architecture of the network intrusion detection includes a window-time sliding algorithm which precedes the clustering step. In this pre-processing phase, a set of network data is continuously captured in a preset timeframe. Network flow information is then extracted from this set of data. By not using the raw network packet itself, the feature space is reduced tremendously. Each progressive step of the architecture reduces the feature space which speeds up the anomaly detection process.
Moreover, the authors of [18] have introduced an incremental grid density-based clustering algorithm. This algorithm can update the feature space for cluster analysis continuously. The proposed clustering algorithm is then used to do clustering in the feature space that is being updated. As the clustering algorithm does not use the points in the feature space directly and instead uses grids which could contain a total number of points more than a certain threshold, the feature space is greatly reduced for cluster analysis. This in turns speeds up the cluster analysis and allows it to process the feature space continuously in a reasonable amount of time.

IV. Conclusion

Real-time detection of network anomalies is vital to ensure the safety of the network. Cluster analysis which can learn from unlabeled data sets to identify anomalies is well-suited for this task. However, there are inherent challenges to each clustering algorithm. In this paper, we have outlined the common challenges and possible solutions to mitigate this problem. Further research can be done to evaluate the advantages and disadvantages of each solution outlined.

References

I. T. Report, “2015 INFORMATION SECURITY,” (2015).

II. B. S. Everitt, S. Landau, M. Leese, and D. Stahl, Cluster Analysis. Wiley, (2011).

III. A. Fahad, N. Alshatri, Z. Tari, A. Alamri, I. Khalil, A. Y. Zomaya, S. Foufou, and A. Bouras, “IEEE TRANSACTIONS ON A Survey of Clustering Algorithms for Big Data : Taxonomy and Empirical Analysis,” vol. 2, no. 3, (2014).

IV. J. Leskovec, A. Rajaraman, and J. D. Ullman, Mining of Massive Datasets, 2nd ed. New York, NY, USA: Cambridge University Press, (2014).

V. M. Daszykowski and B. Walczak, “Density-Based Clustering Methods,” Compr. Chemom., vol. 2, pp. 635–654, (2010).

VI. Z. Ruijuan, C. Jing, Z. Mingchuan, Z. Junlong, and W. Qingtao, “User abnormal behavior analysis based on neural network clustering,” J. China Univ. Posts Telecommun., vol. 23, no. 3, pp. 29–44, (2016).

VII. P. K. R. Oger and A. Zimek, “Clustering High-Dimensional Data: A Survey on Subspace Clustering, Pattern-Based Clustering, and Correlation Clustering,” vol. 3, no. 1, (2009).

VIII. G. Chandrashekar and F. Sahin, “A survey on feature selection methods,” Comput. Electr. Eng., vol. 40, no. 1, pp. 16–28, (2014).
IX. P. Chaovalit, A. Gangopadhyay, G. Karabatis, and Z. Chen, “Discrete wavelet transform-based time series analysis and mining,” ACM Comput. Surv., vol. 43, no. 2, pp. 1–37, (2011).

X. G. E. Hinton and R. R. Salakhutdinov, “Reducing the Dimensionality of Data with Neural Networks,” vol. 504, (2006).

XI. C. Visual, N. Index, C. Vni, and C. Vni, “The Zettabyte Era: Trends and Analysis,” no. June, pp. 2016–2021, (2017).

XII. M. M. Gaber, A. Zaslavsky, and S. Krishnaswamy, “Mining Data Streams: A Review,” vol. 34, no. 2, pp. 18–26, (2005).

XIII. R. Perdisci, D. Ariu, and G. Giacinto, “Scalable fine-grained behavioral clustering of HTTP-based malware,” Comput. Networks, vol. 57, no. 2, pp. 487–500, (2013).

XIV. L. Khan, M. Awad, and B. Thuraisingham, “A new intrusion detection system using support vector machines and hierarchical clustering,” VLDB J., vol. 16, no. 4, pp. 507–521, (2007).

XV. Y. Dang, B. Wang, R. Brant, Z. Zhang, M. Alqallaf, and Z. Wu, “Anomaly Detection for Data Streams in Large-Scale Distributed Heterogeneous Computing Environments,” in ICMLG2017 5th International Conference on Management Leadership and Governance, (2017), p. 121.

XVI. J. Song, H. Takakura, Y. Okabe, and K. Nakao, “Toward a more practical unsupervised anomaly detection system,” Inf. Sci. (Ny.), vol. 231, pp. 4–14, (2013).

XVII. J. Dromard, G. Roudière, and P. Owezarski, “Online and Scalable Unsupervised Network Anomaly Detection Method,” IEEE Trans. Netw. Serv. Manag., vol. 14, no. 1, pp. 34–47, (2017).

XVIII. Á. C. Ning, C. An, and Z. Long-Xiang, “An Incremental Grid Density-Based Clustering Algorithm,” vol.1313, no. 101, pp. 1–7, (2002).
Authors

Jagatheesan Kunasaikaran received his bachelor’s degree from Universiti Tenaga Nasional in the field of software engineering. Currently, he is pursuing his master’s in Information Technology. His area of research is in machine learning and network security. He is also working as a product manager in ZALORA.

Roslan Ismail received his doctoral degree from Queensland University of Technology. He received his bachelor’s degree and master’s degree in computer science from Universiti Putra Malaysia and Universiti Teknologi Malaysia respectively. Currently, he is an associate professor in Universiti Tenaga Nasional. His area of interest is in software testing and cybersecurity.

Abdul Rahim Ahmad received his doctoral degree from Universite de Nantes and Universiti Teknologi Malaysia. He received his bachelor’s degree in computer science from University of Queensland and master’s in information technology from Loughborough University. He is also a diploma holder in teaching obtained from Malayan Teachers College. Currently, he is an associate professor in Universiti Tenaga Nasional. His area of interest is in networking and machine learning.