International Journal of Engineering
Journal Homepage: www.ije.ir

The Ensemble of Unsupervised Incremental Learning Algorithm for Time Series Data

D. Beulah*[a], P. Vamsi Krishna Raj[b]

[a] *Aditya Engineering College, Surampalem, AP, India*
[b] *CIIS, Swarnandhra College of Engineering. & Tech. (Autonomous), Narsapur, AP, India*

P A P E R I N F O

Paper history:
Received 14 May 2020
Received in revised form 19 October 2021
Accepted 06 November 2021

Keywords:
Data mining
Data Stream Mining
Unsupervised Learning
Incremental Learning

A B S T R A C T

Data mining is one of the key concepts to discover hidden knowledge from available data. Along with the data mining, data analytics is a field to analyze and process data in a scientific and cognitive angle. It is more helpful to convert knowledge to actionable knowledge for accurate decision making. Data Stream Mining is another challenging area than normal Data Mining due to its dynamics. Dynamics of data in a stream includes changes in data frequency, volume and nature. This paper focuses on the behavior of data mining of machines in process/manufacturing industries. In general, such data is continuous numerical and time series data captured by various industrial sensors. By nature, equipment or machinery behaviour can change over time. It requires calibration/replacement before failure of machinery. By analyzing data, one can find the behavior or state change. To identify that, dynamic models are required to be built using data mining and data stream mining. Thus, we are following a semi-novel approach for building such models using "*Ensemble of Unsupervised Incremental Learning*" method. Results show how the existing methods are different from the proposed method. This method can be applied for any other domain like image/audio/video or text mining.

doi: *10.5829/ije.2022.35.02b.07*

NOMENCLATURE

D	Dataset	$\|C_i\|$	Density of i^{th} Cluster
$\|D\|$	Number of samples in the dataset (Dataset size)	Q_i	Quality of i^{th} Cluster
X_i	i^{th} Sample of Dataset D	Q	Quality of Cluster method
X_{ij}	j^{th} dimension of i^{th} Sample	SC	Representation for Sub Cluster
C	Cluster Set	F	Field Set
C_i	i^{th} Cluster	F_i	i^{th} Field
C_{ih}	Head or Center of i^{th} Cluster	$\|F\|$	Number of Fields in Dataset
$\|C\|$	Total Clusters		

1. INTRODUCTION

In general, data mining is a collection of methods for knowledge discovery on static data. Data stream mining is a little bit advanced operating on live or stream data. So, existing data mining methods need to be altered to support stream and concept drifts in data streams. Finally, data analytics is used to analyze large historical data and data on highway (live data) with various advanced mathematical, statistical and logical methods. Apart from that, incremental learning is an approach in data stream mining in which incoming data is continuously fed to the model. But the current model with pre-defined parameters or constraints may or may not give accurate results if incoming data behavior is changed. To cater to those requirements for a long time, existing models must be rebuilt or extended. Applications of such models/learning include predictive maintenance, anomaly detection, and pattern recognition.

This paper focuses and describes what are the challenges/limitations of some of the traditional methods and try to integrate them in a systematic way by considering their strengths and try to overcome the weaknesses. This integration improves the robustness of

*Corresponding Author Institutional Email: *beulah@aditya.ac.in*
(D. Beulah)

Please cite this article as: D. Beulah, P. Vamsi Krishna Raj, The Ensemble of Unsupervised Incremental Learning Algorithm for Time Series Data, ***International Journal of Engineering, Transactions B: Applications***, Vol. 35, No. 02, (2022) 319-326

the learning model to increase the reliability and accuracy with considerable positive results in performance metrics.

Paper Organization: Rest of the paper contains the following sections. "*Related work*" section describes some of the work that was done. "*Proposed work*" section describes the definitions, notations, algorithms and their description in detail. Finally, the "*Results"* section describes results comparison with some of the existing methods and performance gains of proposed methods.

2. RELATED WORK

A number of papers are published on incremental learning. According to literature [1-3], it is a process of adopting new behavior on arrival of an incoming data stream. It becomes important to improve the accuracy of the model prediction in interactive environments with the help of human feedback in a timely fashion as described Rico-Juan and Inesta [4]. Specifically, data stream processing must be done in a single pass and cannot mine repeatedly according to Morales and Samoa [5]. Apart from these methods some of the investigators [6-8] paid attention and publicized on unsupervised learning for data stream using cluster methods. Because this paper also depends on clustering, it is worth mentioning some of the existing works with respect to this topic. One of such work is STREAM [6]. It will cluster given data stream in two phases. In the first phase, this algorithm divides incoming stream into various sets/batches of samples and apply an efficient clustering algorithm named as LOCAL SEARCH. After this phase whatever the number of cluster points created, are sent to the next phase. In the second phase, all these cluster points can be re-clustered and generate a global set of clusters. But it is an NP-hard (non-deterministic polynomial-time hardness) problem. Space requirement is proportional to the time. Some other cases include both online and offline scenarios to support live and historical data mining. Even our paper also belongs to this category. It is also having two stages/phases. Offline phase to create an initial set of clusters and online phases are treated as incremental learning. This was proposed by Aggarwal et al. [9].

Other sets of algorithms focus on sliding windows. It was based on the principle that new or recent data samples are more important than the samples arrived long back. Sliding window can be viewed as a set of N recent samples in the data stream. And each set can be viewed as a block. This sliding window method is further divided into fixed length and variable length windows. Each block summary or statistics can be maintained by calculating boundaries or sketches to that block. The sketch can be calculated based on the algorithm described by Greenwald and Khanna [10]. Further, some of the applications are described by Arasu and Manku [11]. An agglomerative hierarchical clustering with random swap was proposed by Hautamaki [12] in Euclidian space. Other approach proposed by Lian and Lei [13] essentially stated that all types of time series data applications need an efficient and effective similarity search over stream data.

2. 1. Challenges of Incremental Learning for Data Stream

Parameter tuning and adaption: Parameters are the most affecting elements in any algorithm. They need to change based on the type of data, domain, signature etc. It is difficult to tune the parameter even for a single model to give accurate results with static or historical data. In that case, it will be far more difficult to tune the parameters in models built for data streams.

Dynamic Normalization: Normalization is a transformation process of raw data from different scales to the same scale. It is very simple in historical data processing but very challenging in data stream processing.

Concept drift detection and handling: Data stream is a collection of samples coming sequentially and changes its frequency and behaviour over time. In that condition, if the model was built based on some set of samples, the prediction coming out of that model may not be accurate when the incoming set of recent samples does not fit in current scale or range. At that time model must be intelligent enough to detect that change and update itself and must be able to give accurate or decent results. But, this is not that easy.

Boundary Samples/ Points analysis: This is a special case in clustering and applicable for some type of clustering methods. It leads to misinterpretation of samples if boundary points are selected as cluster heads. In turn, it leads to wrong tagging or labelling. There should be a proper method to detect boundary points and not to make them as cluster heads, but in those methods this situation may probably occur.

Space and Time Complexity Management: Most of the big data management and data stream applications are memory and time intensive. It is difficult to achieve a balance between space and time complexity. If we try to achieve performance in terms of space, then time must be compromised and/or vice versa. Efficient data structures are required to balance this situation.

3. PROPOSED APPROACH

By considering some of the above challenges, we are proposing certain functionality to handle dynamic normalization, concept drift handling, boundary point interpretation with good space and time complexity management.

3. 1. Definitions

Dataset: It is a collection of samples and fields. Each sample in the data set must have different values for each dimension.

Time series: If each sample in a dataset is associated with a timestamp then it will be treated as Timeseries dataset. Moreover, each value of each dimension in each sample is not random value. It must relate to a concept or context.

Unsupervised Machine Learning: If the samples in the given dataset are not assigned to any predefined labels or classes then processing or learning of such dataset is called unsupervised machine learning.

Cluster: It is a collection or group of samples exhibiting with at most similar behavior.

Crisp/Hard Clustering: Every point/sample in the dataset belongs to only one cluster.

Fuzzy/Soft Clustering: Every point/sample in the dataset can belong to more than one cluster with membership value or weight.

Incremental/Dynamic Learning: In static learning, once model building is over, it cannot be updated or learned. This type is suitable only for historical data mining. To process or mine the data stream, a trained model must be retrained or learn from new samples on a continuous basis.

Signature: It is a collection of independent variables with a single dependent variable. Dependent variable means it is the variable most affected with any small change in any one of the related independent variables. Signature, in other words is the logical representation of an equipment or section in the industry.

3. 2. Dataset

Nature of Dataset we are considering for selection of ensemble algorithm is time series with numerical values and bad data as well. This is a real-time data set retrieved from an Oil and Gas company and belongs to some oil wells. Details of the wells and company are hidden due to a privacy agreement.

3. 3. Proposed Ensemble of Algorithms

The proposed algorithm must run in two phases, offline and online with hierarchical clustering. But we choose the parent level cluster algorithm as different from the child level algorithm by understanding the nature of the data. Parent algorithm is used to divide or partition the given input data into meaningful groups in such a way each group or some set of groups belongs to one state and another set of groups belongs to another state. State here refers to various operating conditions or zones such as off, on, start-up, steady, transient and stale. Off means, the value of certain sensor is equal to 0 or less than minimum "on" range. On means greater than or equal to some value. Start-up range means state between on and steady state or general operating condition. Steady state means variance must be maintained between +/- certain percent. Stale means variance of last n-samples equals 0. Transient means irregular variance among last n samples. State calculation requires domain knowledge and thus out of the scope for this paper. We are using some variables to specify thresholds for state ranges. Parent algorithm uses incremental learning.

Child/Sub Level algorithm is any traditional algorithm such as K-Means. We are also using the K-means for sub level clustering.

One can get the question in their mind, why to choose 2 different algorithms in parent and child levels. In our case, if we choose K-means as a parent or only algorithm for both levels it leads to inaccurate results due to the value of K, because K-means must cluster all the available samples into K-clusters at any cost, whether their behavior really belongs to that cluster or not. It leads to mislabelling or misinterpretation. But we adopted and extended the idea, leader-follower algorithm in parent level clustering which was proposed by Parthasarathy et al. [14] Davies and Bouldin [15] fits the need of the paper.

Apart from the Clustering algorithm, normalization in pre-processing step also refined to adaptive normalization to get more accurate results for test and live streaming data. We are assuming that data with text values are treated as bad data and cannot be part of clustering. We are calculating the running statistics for the last n samples to find the ‘stale' state. Values or samples belong to stale are also not suitable for prediction.

Off-line phase creates multiple signatures and each signature has multiple clusters. Each signature specifies a partial representation of an equipment or machinery in the selected domain. As already told, the proposed model can fit into any domain with a little bit of parameter tuning. On the other hand, online phase utilizes the signatures created in the offline phase and makes the model robust with incremental learning method. Figure 1 shows the proposed system and steps involved in the process flow chart.

Without any doubt above mentioned model is not new in any data mining application but the approach is fresh at Pre-processing, Signature Building and Online Processing stages.

Each Stage is described in detail in the following paragraphs.

System Input: Models must be built on historical data or data at rest. The data source can be anyone like files, databases or historians. As mentioned previously, here data is numerical time series data extracted from various industrial sensors. The proposed model is part of Industrial IoT and analytics.

Pre-processing: This is a filtering and transformation stage to ignore the text and missing values. Handling the missing values itself is a separate concept and not part of this paper. Next step is converting or transforming given data of multiple scales to a single scale. We are using min-max normalization. It requires a single pass of entire

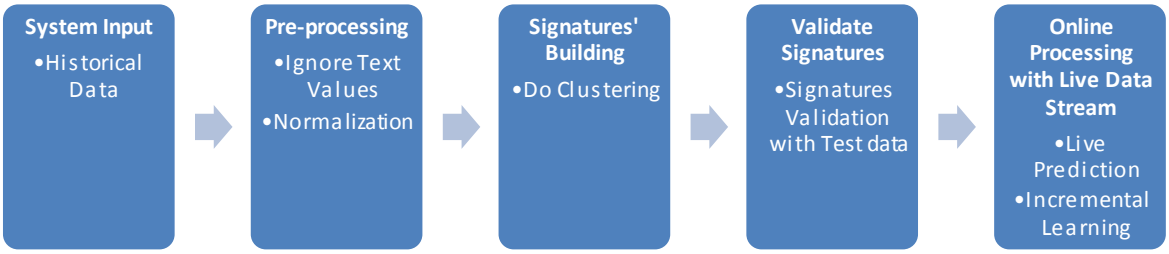

Figure 1. Proposed System Process Flow

dataset to find min and max values of each dimension. But, here the issue arises whenever new min-max comes into the picture with arriving sample.

Finally, we are also calculating the statistics of each dimension to check the quality of input data. If input data for any dimension is having text%>30 and if that dimension is part of any signature, then it is not possible to build the model for that signature.

Signature Building: This is the actual area of our research interest. The proposed unsupervised incremental learning is as follows.

Algorithm:

Procedure *mainProc*

Input: Signature File
Dataset File
Distance threshold (th),
Maximum number of results of recent iterations to match for k-means (noi)
The minimum density of any cluster to give prediction (minden)

Output: Cluster Set (C)

Steps:

1. Build in memory dataset D
2. Calculate each field statistics such as min, max, mean, var, sd, text% at the time of dataset building itself
3. Do min-max normalization on D
4. Build Signature Set (S) from Signature File and sub divide the given dataset D into multiple datasets based on a number of signatures.
5. For Each Signature S_i in S
    a. Call BuildClusters (D,th,noi,minden)
    b. Output Clusters and Save Signature Output with predictions

Description:

Above mainProc is entry point and control point of all other procedures that contain algorithm implementation steps. Step 1 is used to read the data from the input file and create a logical in-memory dataset for next processing. If the size of the dataset is large and if it cannot fit in memory, there are two solutions. First one is to increase the heap size. Second one is to divide the actual file again into multiple files. Next apply algorithm to cluster individually and merge results to achieve global result. The second option is not part of the paper and it is an extension of current work and left as future work. Thus, we are assuming that the input file size can fit in current memory by increasing the default heap size. Step 2 is used to calculate Field/Column/Dimension wise statistics to understand the quality of the data. If data is not having quality, then there is no point to proceed further. Step 3 is used to find the normalization based on min and max values of each dimension. One strong point here is min and max are not min and max of the entire dataset; they are min and max of each dimension separately. Step 4 is used to build the logic in memory signature list from given signature file. As told in the definition each signature has a name, dependent and some set of independent variables. Thus, each signature must have a separate set of data samples i.e. given data set D is divided into multiple data sets based on a number of signatures. Finally, step 5 is used to invoke a clustering algorithm in procedure '*BuildClusters()*' for each signature.

Procedure *BuildClusters* (D,th,noi,minden)

Steps

1. Initialize Cluster Set C
2. For each X_i in X
   do
3. If X_i is the first sample
   then
    a. Create new Cluster newclu and Append newclu to C
    b. Make X_i as cluster head to newclu and add to the cluster
4. Else
    a. Initialize min-dist-clu=null, cnt=0, min-dst=0
    b. For each C_j in C
       do
        i. Calculate ed=EuclDist(X_i,C_j)
        ii. If ed<=th and cnt=0
            Then

1. minclu=C_j
2. mindist=ed;
3. Increment cnt by 1

iii. else if ed<=th and ed<mindist then
1. mindist=ed;
2. minclu=C_j

iv. End if

c. End for-2

5. If minclu =null
Then
a. Create new Cluster newclu and Append newclu to C
b. Make X_i as cluster head to newclu and add to the cluster
6. Else
a. Add X_i to Cluster minclu
b. Update member list(minclu)
7. End if
8. End for-1
9. For each C_i in C
do
a. If $|C_i|$>=minden
b. Then
i. MakeSubCluster(C_i)
c. End if
10. End for
11. For each C_i in C
do
a. If SC(C_i) is not empty
b. Then
i. For each SC_j in SC(C_i)
1. Append SC_j to C
ii. End for
c. End if
12. End for
13. Return C;
14. End *BuildClusters*

Description

Above procedure is the heart of the entire process. As discussed previously no single algorithm outperforms and each will have some limitations. Hence, we introduced two separate algorithms in parent and child level of hierarchical clustering. Parent algorithm strength is, it can well separate samples into multiple groups based on their actual nature. Moreover, it is very much extent suitable for incremental learning. When coming to Child level K-means is used. It becomes very robust when optimal 'K' is found. By combining the strength of two algorithms we designed and implemented the solution. To find the optimal K again we followed the same parent algorithm. But it only restricted to find the maximum number of clusters to be generated. The clusters formed in this stage cannot be used or merged to original cluster list. Only clusters generated from K-Means are appended to the final cluster list.

Coming to the steps, outer for loop is used to find the cluster of each sample X_i in dataset D. If the current sample is the first sample then, itself forms a new cluster and marks itself as cluster head to that new cluster. Then add the new cluster to already initiated cluster set C.

From the second sample onwards, each sample X_i must find nearest clusters among available clusters. Step 4 is used to find that minimum distance cluster "*minclu*". Unlike K-means, the nearest cluster must obey two rules here. The first distance from current sample X_i to current cluster head C_{jh} must be less than or equal to given distance threshold (th). Now the minimum distance of all such clusters will give final minimum cluster (minclu). Here there is no rule like K-means that current sample must fit in any one of the clusters. If it is unable to find the cluster of its nature, it will form a new cluster and marks itself as head to that cluster. Otherwise, it will add to the existing cluster and update the cluster statistics. Steps 5 and 6 describe one more point here, that is, final clusters created from these clustering algorithms are micro-clusters means it maintain only cluster head and cluster statistics to find the quality of the cluster. Another thing is values of a dependent variable must not be included in the distance calculation. It is only used to calculate prediction. Prediction of an incoming sample is mean of values included in the current cluster of respective dependent variable so far. There is a provision also to give the prediction based on density. If the density of the given cluster exceeds a certain number, then only that cluster is eligible to give a prediction. But here we didn't place such restriction. So, from second member onwards prediction is given by each cluster.

According to the above discussion, clusters will be created dynamically based on the distance threshold. Here distance threshold is a tuneable parameter and specifies the radius of the cluster. Thus, this clustering is suitable for incremental learning and type is crisp or hard clustering scheme. Finally, there is no need to create sub clusters for all clusters. The cluster those met minimum density threshold i.e. density of the cluster greater than given '*minden'* parameter, sub clusters are built as described in step 9.

Step 10 is used to merge existing clusters with sub-clusters to generate the final set of clusters to end the model building. At this point, it is possible to calculate cluster quality and cluster scheme quality on whole. This quality measure is used to specify intra and inter cluster similarity. For that purpose, we used Davies-Bouldin Index measure. According to the value of this measure, this index value is inversely proportional to the quality. i.e., increase in the index means a decrease in quality. But < 0.2 also not treated as good quality because it will specify at most one cluster for one or 2 samples. Then it's of no use. In that case, do the above procedure by changing the distance threshold parameter. Auto tuning of this parameter is also possible but it is out of scope for this paper.

Procedure *MakeSubClusters(C)*

Steps

1. Apply Same BuildClusters() logic on Members(C), only to find optimal K
2. Now Call Traditional K-mean on members(C)
3. Return Sub Clusters set SC
4. End *MakeSubClusters*

Description:

The above procedure is used to do sub-clustering. As discussed, the parent cluster algorithm is very intelligent enough to separate given dataset into meaningful state groups. But the limitation is, if boundary point is elected as cluster head then the result will be inaccurate for high-density clusters. Because, once the cluster head is selected, it cannot be changed. We can overcome this limitation through K-means, but difficult in this is getting optimal K. This can be achieved by applying parent algorithm one more time and find that K. Now high-density clusters can be further divided into subclusters and parent clusters of these clusters are of no use after forming sub clusters. Thus, the final cluster set contains sub-clusters and non-dense parent clusters.

Validate Signatures:

This stage is used to validate the signatures based on testing data. But clusters must not be updated because intra cluster statistics will be updated. This is only to understand and evaluate generated signatures and respective clusters' performance.

Online Processing with Live Data Stream:

Finally, this stage is used to read real-time data from live data sources and predict the values based on existing signatures. The advantage of this approach is, this also performs incremental learning. If new sample fits the existing cluster, then its statistics will be updated otherwise form a new cluster and update the cluster set C in the respective signature. Thus, concept drift can be efficiently handled without modifying existing logic. Even though the domain or data type is changed logic will be the same. Normalization part is also dynamically modified. Figure 2 illustrates the data distribution of each dimension. Table 1 summarized the comparative results.

4. RESULTS AND DISCUSSION

Dataset description:
Training Instances: 47200
Testing instance: 65000
Attributes: 4
RTOR (Rotor Torque)
MCUR (Motor Current)
MSP (Moto Speed)
MTOR (Motor Torque)
Where n = number of samples
m = part of n (because, no need to apply sub-clustering for all the samples)

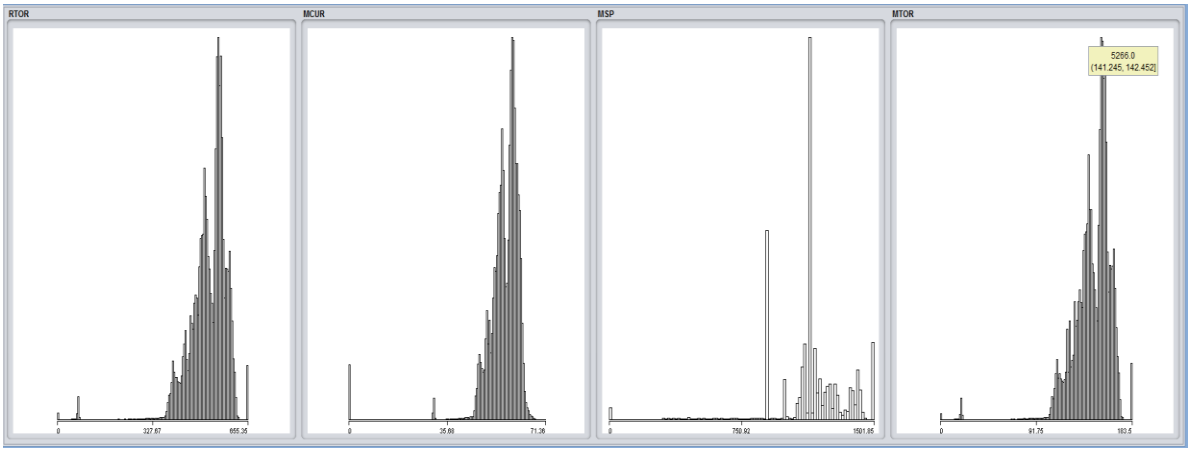

Figure 2. Data Distribution of Each Dimension

TABLE 1. Results Comparison

Measure	Traditional K-Means	Simple Incremental Learning	Hierarchical Incremental Ensemble
No of Cluster	10	12	46
Outlier Clusters	N/A	4	15
Cluster Quality	6.4497	4.1742	2.0723
Time Complexity	$O(n^2)$	O (n log n)	O (n log n) +$O(m^2)$

Even though time complexity of Proposed approach (Hierarchical Incremental Ensemble) is a little bit high when compared to other two methods, quality is good. Both 2nd and 3rd columns are part of the proposed approach; we merged that approach with the ensemble approach to get more accurate results. Outlier clusters are decided based on the density of the cluster. i.e., if the cluster is having a density less than given threshold it is treated as outlier cluster. It is not shown in the algorithm in the procedure because these clusters' density might be increased in the online process when new samples might fit into these clusters. Thus, these clusters not ignored as well.

Figure 3 depicts a scatter plot of clusters with two dimensions' distributions. This distribution will be changed for the other two dimensions. But the important thing here is X-axis must represent independent and Y-axis must represent dependent variables.

Apart from the above discussions some of the points we did not highlight due to space problem are such as dataset file format, signature list file format, output file format, cluster-wise field statistics and cluster statistics, visual representation of parent and child clusters, cluster tagging that is useful for anomaly detection and other cluster measures used in K-means like sum of squared errors. We can provide those details based on readers' interest.

Applications

The proposed Clustering Algorithm can be effectively applied in the following areas:

- Data Pre-processing
- Operating Modes Identification in Industrial IoT Sensor Data
- Outlier Detection
- Anomaly Detection
- Pattern Recognition

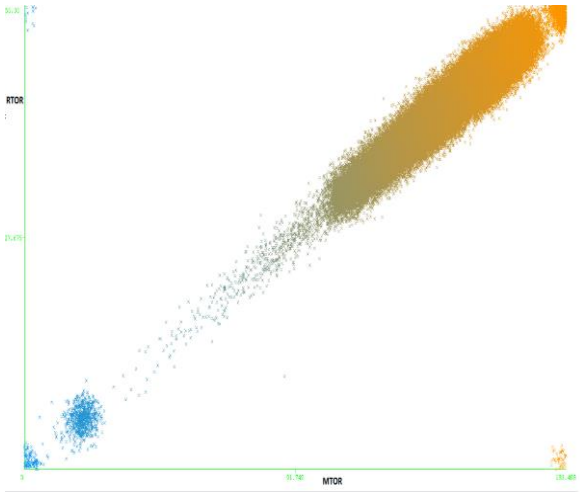

Figure 3. Cluster distributions by considering 2 dimensions

- Document Categorization
- Effective Concept Drift Handling in Data Stream Processing
- Image Processing
- Video Processing

5. CONCLUSION AND FUTURE WORK

This paper mainly focused on ensemble-based incremental learning to achieve more accurate results by using clustering methods for Industrial IoT data. Addressed some of the challenges in data streaming such as concept drift handling, dynamic normalization, incremental learning with hierarchical clustering and border point handling.

As we mentioned in earlier sections, parameter tuning can be automated, make the algorithm more robust by converting into the distributed algorithm. Anomaly detection using cluster pattern recognition is also treated as future work.

6. REFERENCES

1. Silver, D.L., Yang, Q. and Li, L., "Lifelong machine learning systems: Beyond learning algorithms", in 2013 AAAI spring symposium series. (2013).
2. Silver, D.L., "Machine lifelong learning: Challenges and benefits for artificial general intelligence", in International conference on artificial general intelligence, Springer. (2011), 370-375.
3. Luo, Y., Yin, L., Bai, W. and Mao, K.J.E., "An appraisal of incremental learning methods", Vol. 22, No. 11, (2020), 1190, doi: 10.3390/e22111190.
4. Rico-Juan, J.R. and Iñesta, J.M.J.N., "Adaptive training set reduction for nearest neighbor classification", Vol. 138, (2014), 316-324, doi: 10.1016/j.neucom.2014.01.033.
5. Morales, G.D.F. and Bifet, A.J.J.M.L.R., "Samoa: Scalable advanced massive online analysis", Vol. 16, No. 1, (2015), 149-153, doi.
6. O'callaghan, L., Mishra, N., Meyerson, A., Guha, S. and Motwani, R., "Streaming-data algorithms for high-quality clustering", in Proceedings 18th International Conference on Data Engineering, IEEE. (2002), 685-694.
7. Wankhade, K.K., Jondhale, K.C. and Dongre, S.S.J.A.S.C., "A clustering and ensemble based classifier for data stream classification", Vol. 102, (2021), 107076, doi: 10.1016/j.asoc.2020.107076.
8. Dubey, A.K., Gupta, R. and Mishra, S., "Data stream clustering for big data sets: A comparative analysis", in IOP Conference Series: Materials Science and Engineering, IOP Publishing. Vol. 1099, No. 1, (2021), 012030.
9. Aggarwal, C.C., Philip, S.Y., Han, J. and Wang, J., "A framework for clustering evolving data streams", in Proceedings 2003 VLDB conference, Elsevier. (2003), 81-92.
10. Greenwald, M. and Khanna, S.J.A.S.R., "Space-efficient online computation of quantile summaries", Vol. 30, No. 2, (2001), 58-66, doi: 10.1145/1055558.1055598.
11. Arasu, A. and Manku, G.S., "Approximate counts and quantiles over sliding windows", in Proceedings of the twenty-third ACM

SIGMOD-SIGACT-SIGART symposium on Principles of database systems. (2004), 286-296.

12. Hautamaki, V., Nykanen, P. and Franti, P., "Time-series clustering by approximate prototypes", in 2008 19th International conference on pattern recognition, IEEE. (2008), 1-4.
13. Lian, X., Chen, L.J.I.T.o.K. and Engineering, D., "Efficient similarity search over future stream time series", Vol. 20, No. 1, (2007), 40-54, doi: 10.1109/TKDE.2007.190666.
14. Parthasarathy, D., Shah, D. and Zaman, T.J.a.p.a., "Leaders, followers, and community detection", (2010), arXiv preprint arXiv:1011.0774.
15. Davies, D.L., Bouldin, D.W.J.I.t.o.p.a. and intelligence, m., "A cluster separation measure", No. 2, (1979), 224-227, doi: 10.1109/T PAMI.1979.4766909.

Persian Abstract

چکیده

داده کاوی یکی از مفاهیم کلیدی برای کشف دانش پنهان از داده های موجود است. در کنار داده کاوی، تجزیه و تحلیل داده ها زمینه ای برای تجزیه و تحلیل و پردازش داده ها در یک زاویه علمی و شناختی است. تبدیل دانش به دانش عملی برای تصمیم گیری دقیق مفیدتر است. داده کاوی زنده به دلیل پویایی آن، یکی دیگر از حوزه های چالش برانگیز نسبت به داده کاوی معمولی است. دینامیک داده ها در یک جریان شامل تغییرات در فرکانس، حجم و ماهیت داده است. این مقاله بر رفتار داده کاوی ماشین‌ها در صنایع فرآیندی/تولیدی تمرکز دارد. به طور کلی، چنین داده‌هایی، داده‌های عددی و سری زمانی پیوسته هستند که توسط حسگرهای صنعتی مختلف گرفته می‌شوند. طبیعتاً، رفتار تجهیزات یا ماشین آلات می تواند در طول زمان تغییر کند. قبل از خرابی ماشین آلات نیاز به کالیبراسیون / تعویض دارد. با تجزیه و تحلیل داده ها، می توان رفتار یا تغییر حالت را پیدا کرد. برای شناسایی آن، مدل های پویا باید با استفاده از داده کاوی و جریان کاوی ساخته شوند. بنابراین، ما یک رویکرد نیمه جدید را برای ساخت چنین مدل‌هایی با استفاده از روش «مجموعه یادگیری افزایشی بدون نظارت» دنبال می‌کنیم. نتایج نشان می‌دهد که چگونه روش های موجود با روش پیشنهادی متفاوت است. این روش برای هر حوزه دیگری مانند تصویر/صوت /ویدئو یا متن کاوی. قابل استفاده است.