REVIEW AND PROPOSING EFFICIENT ALGORITHM FOR AGGREGATING AND ANALYSING STREAMING IOT DATA

Sr No.	Author/Year	Paper Title	Core Contribution	Algorithm Methodology (algorithmic approaches such as greedy, dynamic programming divide and conquer etc.)	Parameters if any	Strength	Limitations	Remarks (such as complexity, scope of improvement etc..)
1.	2016 Furqan Alama , Rashid Mehmoodb, Iyad Katiba , Aiiad Albeshria	Analysis of Eight Data Mining Algorithms for Smarter Internet of Things (IoT)	Comparative analysis of 8 data mining algorithms including Support Vector Machine (SVM), K-Nearest Neighbours (KNN), Linear Discriminant Analysis (LDA), Naıve Bayes (NB), C4.5, C5.0, Artificial Neural Networks (ANNs), and Deep Learning ANNs (DLANNs).	This paper applies eight different data mining techniques on three datasets and enunciates their merits. and demerits. It compares these methodologies on parameters such as confusion matrix, accuracy rate and elapsed time. The computations are performed on the Aziz supercomputer.	Computation on Aziz supercomputer which is Fujitsu made and is able to deliver peak performance of 230 teraflops. It has a total of 11,904 cores in 496 nodes. Simulations are done on R platform. Datasets are from the UCI data repository and are real time sensor and accelerometer readings on human postures and movements.	The study provides a concise and clear comparison of the most common and easy to understand algorithms based on essential parameters . It judges the C4.5 algorithm as the best among those considered.	The study does not introduce any novel approaches towards the fields of IoT data analysis. It presents the best choice among the available eight according to user needs, but does not provide a means to optimize the rest.	Each of the eight algorithms is dominant in one parameter. For instance, KNN is simple and has good accuracy but is slow in execution. C4.5 is complex but performs better in the other fields. Thus, methods need to be developed to come up with an algorithm that has considerable accuracy, fast speed and is also simplistic in approach. This can be done by applying optimization techniques to existing algorithms or by merging the advantageous components of them all into a single procedure.
2.	2020 Erwin Adi, Adnan Anwar, Zubair Baig, Sherali Zeadally	Machine Learning and Data Analytics for the IoT	This study highlights the correlation of machine learning algorithms and IoT data analysis. It explains various analysis methods utilized including predictive, descriptive, prescriptive and adaptive analyses. It also compares the solutions based on infrastructural requirements with respect to speed and accuracy of the application.	Descriptive Analysis: Detailed insights into past events by studying large amount of raw data through algorithms such as clustering. Predictive Analysis: Predicting the outcome of future events by studying historical data through classification and regression. It is implemented through Decision trees or Neural Networks. Prescriptive Analysis: Provides recommendations for the handling of future events	Applications: Smart healthcare, home, vehicles and energy efficiency Protocols: AMQP, HTTP, MQTT, CoAP Machine learning algorithms: Random Forest, Azure Machine Learning, K-NN etc	The study provides a wealth of data regarding the usability of various component of the IoT architectur e for different purposes. It considers a broad range of fields, protocols as well as infrastructu re models including cloud, fog and edge computing.	The paper is presented more as a literature review than a research project. Although it curates and summarises developments over the ages, it does not propose any original solutions. Also, since it considers so many applications, it lacks in-depth analyses of its sections.	The study can be improvised by considering a single parameter and working to optimise the same. For instance, choosing the protocol as a variable factor with hardware and infrastructure constant would enable thorough study and may yield better application specific results. Also, it might prove beneficial to consider various machine learning algorithms and arrive at a general consensus on the best approach for a particular pool of

				through cloud or edge computing. Adaptive Analysis: Updating outcomes of predictive analysis based on latest inputs received.	Knowledge outcome: Energy consumption, Optimum traffic route, Personal health prediction	It clearly explains the discretion behind utilising these for particular purposes		applications. This would provide a generalised solution for smaller applications though it might prove insufficient for specialised purposes.
3.	Mohammad Saeid Mahdavineja, Mohammadreza Rezvan, Mohammadamin Barekatain, Peyman Adibi, Payam Barnaghi, Amit P. Sheth	Machine learning for internet of things data analysis: a survey	The main aim of the paper is to analyse how machine learning algorithms can be applied to IoT data and to bifurcate these as per applications. Further, it also tests the use of IoT in real time situations including the construction of Smart Cities. The researchers have extensively reviewed over 70 papers written by 20 authors and categorised existing data mining algorithms into eight distinct groups based on their structural similarities and the type and amount of data they can handle.	K Nearest Neighbours: It promotes the idea of classifying an unseen data point by identifying k nearest data points in terms of input features. Thus, it utilises distance metrics like Mahanalobis distance, Eucledian distance etc. However, it requires the entire training data to be stored which makes it unscalable for large data applications. Naïve Bayes Approach: It is a probabilistic model that categories an input vector by applying Bayes theorem. It assumes that the parameters of the input are independent, thus the name Naïve Bayes. Support vector machine: It is one of the best supervised algorithms for binary classification of data. It aims to find a hyperplane that is a linear function of the input variables and divide the data on either side of the same. Linear Regression: It aims to learn a linear mapping f: x→ y inorder to predict values.	Computing Framework: Fog computing, Edge Computing, Cloud computing Machine learning approach: KNN clustering, regression, Naïve Bayes, SVM, Combined models (Classification and Regression trees), Random Forests Applications in smart cities: Traffic Optimisation, Autonomous cars, Smart Citizens, Urban Planning	The study considers a large number of technologies and is based on renowned works in the domain. It offers solutions to numerous IoT data analysis application including fast processing(Fog Computing), spam filtration and text categorization(Naïve Bayes) and smart cities(SVM)	As the researchers themselves note, Smart City development has been chosen as the focal point of the research since 60 of the reviewed papers are based on the same. However, the paper does not extend the analyses to broader applications nor does it significantly conclude the most suitable framework, protocol or algorithm for the said topic. It identifies more as an academic text explaining machine learning approaches in IoT but does not provide any inputs of its own, being a literature review.	The paper highlights the unscalability of the KNN algorithm in big data analyses. Thus, the procedure needs to be optimised to extend its usability since it is one of the most simplistic and easy to comprehend. The Naïve Bayes approach has considerable accuracy except for the fact that it makes arbitrary assumptions. It can be improvised by considering the dependence of the most important parameters on the others instead of labelling all as independent. SVM is one of the most popular models but its parameters are difficult to interpret. Therefore, it can be simplified to make its results concise and accessible.
4.	Bhavesh Gawri, Anirudh Kasturi,	An efficient approach to kNN algorithm	KNN is one of the most used methods of classification, but	The study uses the concept of Breadth First Search (BFS) traversal of a tree to reduce time	Data sets: Iris	The solution addresses the dual	The model does not describe any use cases of the approach. It	The model needs to be tested against real time IoT data to analyse how it

	Lalitha Bhanu Murthy Neti, Chittaranjan Hota	for IoT devices	it suffers from high space complexity. For this reason, it cannot be scaled to big data applications. Methods such as the kd tree have been developed to address this issue. This work aims to provide a novel solution to the complexity of the algorithm by analysing the spatial arrangement of the data points in addition to the nearest neighbour properties. Thus, processing is made more effective.	complexity of the KNN algorithm. Kd tree is a k dimensional binary tree in which neighbours can be retrieved with logarithmic complexity in contrast to the linear complexity of the regular approach. It also uses spherical coordinate system to represent the points which are then divided into sectors. These sectors are assigned labels by identifying the K nearest neighbours through BFS from the tree. Thus all data points need not be stored in the testing phase since these labels can be utilised now to classify new data.	Size 150, 4 dimensions, 3 classes Wireless Indoor Localization, Size 2000, 7 dimensions, 4 classes HTRU2, Size 17898, 8 dimensions, 2 classes Statlog, Size 58000, 9 dimensions, 7 classes Poker Hand, Size 1025010, 10 dimensions, 10 classes	concerns of time and space efficiency. The BFS approach reduces time complexity to logarithmic from linear while the spatial approach negates the requirement for complete storage of data. Thus, scalability of the model is improved.	is theoretical study and is not extrapolated for specific use in real time IoT data Analyses.	performs in that scenario. Also, an approach is essential for deciding k(the number of neighbours) and also the number of sectors in the spatial partition. The fixing of hyperparameters is of utmost importance in algorithms such as these since they can make or break the accuracy of the model.
5.	2016 Bill Karakostas	Event prediction in an IoT environment using naïve Bayesian models	The paper explains in detail the Naïve Bayes model of prediction. It adopts a predictive analyses of IoT data. It collects varying IoT data from airports through devices like sensors and systems for air traffic management. It forms a database of the percentage of delays occurring in flights. It then develops a model for predicting future delays in connecting flights due to the connecting flight departing late as a consequence of	Naïve Bayes approach is utilised in this model where the probability of the connecting flight departing late (CFDL) is calculated given the probability of delay in departure of incoming flight (IFDL) or the probability of the incoming flight arriving late (IFAL) as an input. It also categorizes these delays as short, medium and long in order to offer deeper insights. The relationships are established as follows: IFDL→ CFDL IFAL→ CFDL IFDL→ IFAL→ CFDL	Data set: Flightstats.com For a period of 30 days from September to October 2015 Inputs: IFAL and IFDL classified as Short(S), medium(M) and large(L). Where a short delay corresponds to less than 30 mins, a medium delay 30-45 mins and a long delay >45 mins. Output: Probability of	The Naïve Bayes is simplistic and quite accurate. Also, utilising it for the practical application of predicting flight delays is quite apt. It addresses a real life problem and provides the outputs as simplified probabilities. Thus the system can be utilised by a layman	The study is aimed at a very specific issue. Also, predicting the delays in connecting flights is not as useful, since its passengers will already be aware of any delay in the incoming flight and might have already estimated the outcome well in advance. Extending the capability to other flights including non-stop connections would prove very useful for travellers and airport personnel alike.	The model can be extended to predict delays in all types of flights. The historical data of past flights, weather, passenger inflow and other parameters may be utilised for the same. An adaptive analysis model may be developed to integrate current inputs of traffic into the equation to make it even more accurate. This will result in the development of a truly real time delay prediction system, thus avoiding unnecessary hassles.

			the incoming flight arriving late.	The model is then constructed based on the Bayes theorem equation of the above constraints.	CFDL for short, medium and long delays.	once developed as its intricacies are not reflected in the output.		
6.	2020 Chaomin Li	Information Processing in Internet of Things using big data analytics	This paper proposes the model design which uses the Fog computing for the smart and real time healthcare information processing. This paper aims to use the big data tools with the holistic platforms for the monitoring and processing of the real time data. It also discusses the plan to analyse its efficiency in terms of transmission cost, storage cost, specificity, sensitivity, accuracy and F-measure.	This processing model is composed of three layers: IoT body sensor network layer which collects the data, Fog processing and computing layer which processes informs, analyse it, and classify it using Naïve Bayes classifier, cloud computing layer which performs data analysis, store the data and decision-making classification. The detailed flow consists of: data accumulation model, data encryption and compression model, data aggregation model, data pre-processing, information extraction, data normalization, data filtration, data analysis, cloud computation layer and remote health monitoring.	Dataset: Healthcare data set from UCI dataset repository Environmental dataset from data repository of US EPA IoT Sensor network layer: utilized for data accumulation, aggregation, compression and encryption Spark and Handoop Ecosystem: utilized for information extraction, data normalization, rule engine, data filtration, data processing Naïve Bayes classifier: for data classification Kalman filter (KF): utilized for the filtration of irrelevant and noisy data For the cloud computation layer they have utilized Apache	The proposed scheme attains result as: less SRHIP transmission cost up to 40%, improved transmission ratio up to 15%, Accuracy, specificity, sensitivity, and F-measure of the proposed NB based SRHIP system are 96.5%, 96.7%, 93.6%, and 94.3% respectively. While the other classifiers like SVM, ANN and KNN has the accuracy of 90.1%, 91.3%, and 94.4%, has specificity of 91.1%, 92.3%, and 91.3% and f-measure of 91.5%, 91.2% and 90.3% respectively.	While Fog computing might be the efficient way, this study adds the task of the encryption of the large amount of data. Also, the cost to set up and maintain the fog computing might go higher.	This paper can be extended by adding the algorithms which can efficiently work on the reduction of the data redundancy, as it increases the processing load on the encryption. It also increases the energy usage if the data redundancy is not taken into the consideration. It focuses on the reduction of transmission cost but should also focus on the way it collects data and processes it which can significantly reduce the cost by reduction in data.

					Spark for real time stream data processing			
7.	2019 Klemen Kenda, Blaz Kazic, Eric Novak, Dunja Mlandic	Streaming Data Fusion for the Internet of Things	This paper proposes the framework using the feature vector which can be applied in the machine learning algorithms for the data fusion of the heterogeneous data streams. Feature vector consists of all the necessary information required for prediction. Its rich feature vector facilitates accurate predictive modelling. It provides real-life applications for the heterogeneous multi-sensor data streaming. It has also demonstrated its use in cloud and edge infrastructure. This work aims to provide a generic framework for data fusion of a set of heterogeneous data streams. It mainly focuses on the three types of data: sensor, weather and static data.	The framework proposed in this paper consists of three parts: pre-processing, fusion and modelling. The pre-processing step generates the partial feature vector using resampling time by updating the value in the vector and using data stream configuration. It will generate the feature vector for each collected data and pass it forward at a specific time (the time of data collected from different sensors might be different so it needs to taken into consideration) for complete fusion. In the fusion process authors propose this algorithm: If the data is available from all the partial feature vector of sensors, store the data, if data is unavailable then store the data and trigger the new instance for the data. If data is available only from one of the partial feature vectors, then empty the complete feature vector and trigger for new instance. Now this feature vector can be used in various ML algorithms. For the Integration it provides two different options: Cloud Infrastructure and Fog/Edge Infrastructure.	For the data processing this study utilizes the lambda Architecture, which is used for big data processing. For the integration in the cloud layer in the communication layer they use message queue which here was Apache Kafka. It has implemented stream fusion system in QMiner stream processing engine which enables fast prototyping and rich ecosystem of implemented stream aggregate operators. For the edge/fog infrastructure, the messaging queue can be omitted and data adaptors can be connected directly to the data source via HTTP API. Modelling and framework implemented using QMiner framework. And the predictions are made available using WebSocket Protocol via GUI.	Here the given model can store the historical data in form of feature vectors and also provide prediction based on that. The paper also studies its use cases regarding electricity distribution like cloud infrastructure for smart grid and edge/fog infrastructure for the public trains. For the smart grid it utilizes the electricity records of the past 2 years of every hour and gives its prediction accordingly showing its capability to handle, aggregate and utilize large amount of data with the inclusion of historical values.	It provides an overall generic framework, but doesn't present the option for the real-time data streaming.	Overall, this paper presents the work along with the tested analysis.

8.	2022 Vani S. Badiger, Ganashree T. S	Data aggregation scheme for IOT based wireless sensor network through optimal clustering method	This paper aims to address resource constraints like data redundancy, high energy consumption, packet collision which arises due to huge amount of real-time data generated. In this paper authors propose the efficient data aggregation scheme (EDAS) which considers improved low energy adaptive clustering algorithm to form optimal number of cluster head by considering node residual energy and average network energy. It also aims to eliminate data redundancy using network coding technology which integrates linear XOR operation which reduces the traffic load and improves network lifetime.	Clustering scheme is proposed in way which balances node energy and integration data aggregation function. The network model consists of the sensors and clusters are formed among these. Each cluster head will collect the data and transmit it to the base station which is placed away. This base station will collect the data and send it to the cloud for further processing. Each sensor node is assigned unique ID. Initially each node is assigned same energy. And the distance between each node is calculated using Euclidean distance. Here the cluster head is selected considering following parameters: Energy consumption of given node, distance from the base station, energy dissipation and optimal cluster head is decided unlike conventional methods where threshold is considered and random cluster head is generated. These conditions will be checked for every node of the sensor and then the best will be selected as the cluster head. For the data aggregation linear XOR operation is used during inter cluster data transfer which will ensure replication free data transfer which will minimise energy consumption of cluster head. During data transmission two things are taken into consideration stable link and sequence in which data is transferred.	The stimulator used for the efficient data aggregation scheme is NS2. Machine learning approach: Linear XOR operator	While the traditional researched focuses on the data aggregation ignoring the energy consumption, data redundancy and packet collision parameters, this research aimed to provide the efficient way to do this. Here the packets received at the base station through the EDAS is more compared to traditional approaches due to its efficient way of selecting cluster head. Also, it consumes less energy and extends network lifetime due to the same reason. Also, number of the alive nodes are more compared to the traditional methods.	This paper summaries only the generalized theory for the energy consumption per number of packets received. It doesn't provide the case-studies and studies the real-time data available.	This paper proposes the way to select the appropriate node, but it uses the iterative method to do the same. It might be possible that there are multiple number of nodes in a provided space which might cause redundancy. E.g., if we got the competent node at the first iteration itself but still, we will have to compare it with the other node available to make sure it is competent enough.

9.	2021 Shamim Yousefi, Hadis Karimipour, Farnaz Derakhshan	Data Aggregation Mechanisms on the Internet of Things: A Systematic Literature Review	This paper aims to present the approach for data-aggregation from both client-server based and mobile agent-based in IOT systems. It provides clearer view by dividing it into two different categories. Further client-server-based data aggregation system is studied under three mechanisms: cluster-based, tree-based, and centralized. It compares the approaches in detail. It describes the architecture, applications based on IOT. It also discusses the challenges, and future research possibility of research in IOT.	For the cluster-based mechanism it discusses some of the already existing works which focuses on efficient data transmission while reducing energy consumption during transmission, improve transmission delay time, failure tolerance, low traffic load, improving IOT lifetime, security and privacy in IOT. In the tree-based data aggregation mechanism it discusses several works which has used tree-based techniques to overcome challenges like reduce transmission delay of sensitive data, improve energy consumption by providing shortest path, improve network lifetime, robustness which will help to improve the quality of data. For the centralized data aggregation mechanism, it discusses the work which are capable of providing the secure data transmission between the IOT devices and base stations.	It mainly discusses and compared the idea and advantages of the existing works, thereby having no such parameters.	This study compares the approaches based on the energy, security, lifetime, delay, computational cost, and reliability. This can be useful for the future researches to identify the best existing methods according to the required application.	This article doesn't present any novel approach rather lists out the existing. While this study reviews the mechanisms, it doesn't list out or differentiate them in the applications they can be utilized.	Each of the provided mechanisms are efficient in their own ways and lists out its pros, but their real-time use cases might have been more useful for the researchers for the further advances. It could be improved if it adds use cases in each of its presented mechanisms with specifying the algorithms.
10.	2015 Sadia Din, Hemant Ghayvat, Anand Paul, Awais Ahmed, M. Mazhar Rathore, Imran Shafi	An Architecture to Analyze Big data in the Internet of Things	With the generation of the massive amount of heterogeneous data in the IOT, the problem arises to extract the useful information from the high-speed aggregated data generated. This paper proposes an architecture to analyze big data. It divides the data into different	First the paper describes the requirement of the architecture like the infrastructure should accept the heterogeneous data, should have the capability to forward the data for processing, have proper communication protocols and should have mobility. In healthcare the requirement is that the same sensor should transfer the data to the multiple systems. So, the	6LoWPAN sensor attached to the human body used for the collection of data like temperature, ECG, running, walking. Handoop server node setup on UBUNTU 14.04 LTS core i5 machine with	The proposed architecture significantly reduces the processing time. It processes the GBs of data into few seconds.	It states that the proposed architecture might take the larger amount of time for the smaller data compared to the larger datasets as the Handoop uses the Map and Reduce function which requires the	The paper doesn't illustrate its finding in detail nor does it provide the complete information and specific features. Also, it has redundancy in terms of the size of data sets. It could have been more précised and can be improved like by providing specific algorithms. The proposed

			subsets based on the complex magnitude of the data. The paper also presents the model for the fusion of the data using Handoop server to improve computational efficiency. It also tests the model on the healthcare data set.	system must be equipped with it. Here the data is collected from 6LoWPAN sensor attached to human body which collects the data like temperature, ECG, walking, running. Data preprocessing is done using the techniques like data integration, data cleaning and data redundancy elimination. After which in the aggregation block the data block is prepared of the collected data. In the storage server data is stored, it filters the data, it can also be shared and it queues up the data enhancing its data processing capabilities. After this data is transferred to data processing server. It uses the Handoop processing server. In this big data block are divided into the small data blocks for processing by data fusion technique. It also compares the data block. It also fills the empty values with the dummy values. Then it moves on to the decision-making block where it is divided into two blocks, fusion storage and decision making. After the decision is made it is stored into the fusion storage device.	3.2 GHz processor and which has 4GB memory. Analyzed data of patients with 4GB memory and also 2 GB memory.		large amount of data. It has mainly illustrated its ability and use case in the healthcare monitoring and moreover has the generalized data rather than the specific algorithms.	architecture should work independent of the size of the dataset provided.

Research Gap (Your observations based on the limitations of these 10 papers)

Based on the research papers reviewed, it may be concluded that there are many options of data analysis in the Internet of Things domain. However, they each have their own limitations. Moreover, there doesn't exist a general consensus on the methodology to be adopted in the case of specific applications.

The most frequently observed research gaps in the field of IoT data analyses are as follows:

Scalability: The digital world experiences a constant inflow of information. Thus, researchers often face the need to comb through enormous amounts of data. This process is both time and memory intensive. Simple algorithms like KNN classification are unable to handle such datasets. Thus, specialised techniques need to be developed specifically for big data analysis.

Security: Along with the permission to collect and store data comes the responsibility of protecting it. Especially when the data is forwarded to external servers, there arises a need to encrypt it before any kind of transmission can occur. Thus, analysis algorithms need to be intelligent enough to first decrypt information while also maintaining the users' privacy.

Speed: IoT is often connected with real life applications. There are some situations where a decision might need to be made instantly and acted upon. For instance, autonomous cars can afford only milliseconds to avoid an obstacle and avert collisions. Thus, the server may not be located far from the collection source. Also, the algorithm utilised should have the ability to recognise and organise its priorities.

Simplicity: IoT devices are most often used by consumers and end-users. Thus, the analysed data needs to be presented in such a manner that it is easily understood by the user. A simple example is a weather prediction system, where the output is graphical and in the format of probabilities.

Preprocessing: The data sent from the users' end might not always be in digitally usable format. Thus, it needs to be cleaned and pre-processed in order to be of use.

Noisy and incomplete data: Since sensors are susceptible to the external environment, they are affected by noise. This gives rise to outliers and missing fields. Most of the machine learning algorithms are affected by such incomplete data, due to which the signal may need to be filtered before passing for analysis.

Objective:

Our objective is to provide a generalised algorithm for the analyses of IoT data. The main concern is to ensure that the approach is simple, scalable and efficient.

Your proposed algorithm/approach:

- Let us consider there are n IOT devices/sensors from which we collect the data for our model distributed in M X M size area.
- Now we collect the data from this and send it to the base station and pre-processed and then sent to the cloud infrastructure for the further decision making and analysis.

For data-aggregation:

- Now we divide the M X M region into let us say x parts. From each region we will have one cluster head.
- So, now we have x cluster heads which will collect the data from the region and will transmit it to the base station.
- Before considering the node for the cluster head it is better to know its energy retaining capacity and distance from the base station which will save the time required for selecting cluster head.
- While choosing the cluster head we would consider the following things:
    - Threshold energy
    - Its distance from the base station which can be calculated using Euclidian distance
    - Its energy retaining capacity as it could raise the system in the problem if the battery dies out often and changing battery in the node can be difficult task.
- Once the cluster head is selected it will transfer the data to base station for pre-processing.
- While transferring the data we need to take into consideration like the traffic load, we could set the particular frequency on which data travels which will prevent collision and smoothen the process.

For data pre-processing:

- Now once the base station receives the data from the cluster heads, it should start dividing it into the different subsets which could be in form of the vectors. For example: V = {(x1, t1), (x2, t2), …..} where x1 represents the data collected at time t1. Similarly, different vectors can be formed based on the similarity of the information.
- While forming the vectors we need to take into consideration the data redundancy.
- By forming groups, we can also remove the duplicate values.

- Different vectors can be formed for different type of information based on the requirement. Some of its use cases are as follow:
    - If we have the specific requirement like there is some sensitive data which needs to be transmitted as soon as received these subgroups can really be useful. While, transferring the data priority queue can also be utilized.
    - Now we have some information like we want to predict the weather then we would require multiple data like humidity, moisture, air quality index. So, for which we can set the algorithm as we have to forward the packet that is the partial vector only if we have all the available information. If it has some missing information then it will have to trigger it for new instance and wait to receive all the necessary data.
- Now these vectors will transfer the data to the cloud for further prediction.

For Cloud- Infrastructure:

- For the cloud we could consider two parts:
    1. Which will predict the results.
    2. Which will store the result.

Time Complexity:

The given approach divides the data aggregated into smaller sizes. Also, if we collect some information beforehand and provide it as static input, it will reduce the amount of time required for the computations. So, the time required for selecting cluster head will only be required for the first time and later we can have the same cluster head. So, its time complexity is number of nodes available in particular region. Also, this method can be helpful in many ways like helping to save energy, providing accurate results and improve lifetime of the network.

Also, the formation of the vectors at the data pre-processing step will help to differentiate the data and will only send the required data. This also supports the heterogeneous data. So, its complexity will depend on the data required for prediction. Also, by forming the vectors we can solve the problem of data redundancy and filtration of data.

Overall, this generalized approach can help us and serve us the purposes like: energy efficiency, improve network lifetime, managing heterogeneous data, addressing problem like data redundancy, filtration of data and managing large amount of data efficiently.