# HaaS in Environmental Computing: Hadoop-as-a-Service for Big Data Mining in Environmental Computing Applications

APARNA S. VARDE
1. School of Computing
2. Clean Energy & Sustainability Analytics Center
Montclair State University, Montclair, NJ, USA
(vardea@montclair.edu) — ORCID ID: 0000-0002-3170-2510

This article addresses the importance of *HaaS (Hadoop-as-a-Service)* in cloud technologies, with specific reference to its usefulness in big data mining for *environmental computing* applications. The term environmental computing refers to computational analysis within environmental science and management, encompassing a myriad of techniques, especially in data mining and machine learning. As is well-known, the classical MapReduce has been adapted within many applications for big data storage and information retrieval. Hadoop based tools such as Hive and Mahout are broadly accessible over the cloud and can be helpful in data warehousing and data mining over big data in various domains. In this article, we explore HaaS technologies, mainly based on Apache's Hive and Mahout for applications in environmental computing, considering publicly available data on the Web. We dwell upon interesting applications such as automated text classification for energy management, recommender systems for ecofriendly products, and decision support in urban planning. We briefly explain the classical paradigms of MapReduce, Hadoop and Hive, further delve into data mining and machine learning over the MapReduce framework, and explore techniques such as Naïve Bayes and Random Forests using Apache Mahout with respect to the targeted applications. Hence, the paradigm of Hadoop-as-a-Service, popularly referred to as HaaS, is emphasized here as per its benefits in a domain-specific context. The studies in environmental computing, as presented in this article, can be useful in other domains as well, considering similar applications. This article can thus be interesting to professionals in web technologies, cloud computing, environmental management, as well as AI and data science in general.

DOI: 10.1145/3704991.3704995 http://doi.acm.org/10.1145/3704991.3704995

## 1. INTRODUCTION

Advances in many technologies as well as continued reduction in data storage and processing costs have led to data explosion, including human data in the form of emails, photos, messages, blogs and tweets on social media and digital data generated by sensors, such as telescopes, cameras and GPS (global positioning systems) to name a few. Data volumes have expanded from terabytes and petabytes to zettabytes and yottabytes today. The available data can be huge in volume and complex in terms of the number of data sources and interrelationships, posing challenges in storage, querying, sharing and analysis. Distributed File System and the MapReduce programming model originally introduced by Google have remained very popular as the technology for big data storage and processing.

MapReduce is a programming model introduced and successfully used at Google in order to perform various computations, such as inverted indices and Web crawl data summaries over large volumes of data distributed across thousands of machines.

Cloud computing has been proposed to operate business on the Web using a pay-as-you-go model. Computing service providers offer on-demand network access to configurable computing resources, data management, analytics and knowledge discovery;. Apache Hadoop, Hive and Mahout are available as cloud technologies. These fall in the paradigm of *HaaS*, i.e. *Hadoop-as-a-Service* [Hadoop 2021]. Some of the leading companies as HaaS providers are Amazon Web Services, HP Cloud, Google Cloud, HortonWorks, Cloudera, Microsoft Azure - HDInsight, and IBM BigInsight. More than a thousand companies today use HaaS including giants such as Ebay, Amazon, Facebook, Google, IBM, LinkedIn, Twitter, Yahoo, Adobe and Alibaba. The use of cloud services for data storage, information retrieval and knowledge discovery has numerous advantages from the perspective of greenness and energy saving as well. This is particularly because the cloud providers have a better PUE (Power Usage Effectiveness) than most server based systems in data centers of individual companies, universities and other institutions as observed in the literature [Pawlish et al. 2014], [Liu et al. 2020]. Accordingly, there are trends to consider the DevOps paradigm [Mclvor 2016] in many organizations today in conjunction with cloud technologies. These trends have their advantages and limitations [Pawlish and Varde 2018]. Cloud services can be helpful in scientific domains using languages such as MML (Medical Markup Language) [Araki et al. 2000] developed in Japan. Its applications along with concerns are important to address [Tancer and Varde 2011]. On the whole, using the cloud with HaaS is viewed in a positive light and forms the focus of many works.

In general, some of the other reasons for using HaaS can be listed as follows.

(1) The usage of Hadoop clusters does not require learning complicated details of Unix/Linux system administration, but instead can be rather seamless.

(2) Minimal installation and configuration can be sufficient, e.g. Amazon Web Services (AWS) offers machine images with Hive, cmd3s, Maven already installed on EC2, thus facilitating the development of applications.

(3) Multiple HaaS technologies, e.g. Hadoop, Hive, Mahout, are open-source, hence making their overall adaptation more convenient

Over the years, research in AI and data science has delved much into predictive analytics on large and complex datasets in multiple scientific domains such as medical & health informatics [Karthikeyan et al. 2020], physics [Ko et al. 2023], chemistry [Tetko et al. 2016], earth science [Gevaert 2022], materials science [Varde et al. 2007] and others. The knowledge of the domain coupled with the technologies available can help propose solutions to challenging domain-specific problems and can be extremely useful in a plethora of applications. The availability of cloud technologies can further enhance the adaptation of techniques in data mining and machine learning, thereby augmenting solutions in predictive analytics. HaaS can play a vital role here, via providing technologies such as Hive and Mahout to use “as a service” for big data management and knowledge discovery, with the typical pay-as-you-go model, without huge investments for server-based data storage and / or acquisition of expensive customized software tools for analysis.

In this work, we focus mainly on the domain of environmental science, emphasizing where HaaS can be useful in *environmental computing*. The term *environmental computing* can have a broad context. It refers to computational technologies being applied to studies in environmental science and management. Given the huge proliferation of AI and data science across several domains today, the term environmental computing certainly entails much emphasis on data storage and information retrieval, as well as data mining and machine learning to discover interesting knowledge. Hence, environmental computing can deploy numerous techniques in data mining and machine learning, spanning a variety of algorithms in supervised as well as unsupervised learning. It can serve a range of applications that can have impacts on addressing climate change, providing energy savings, enhancing urban planning and more.

The goal of this article is to investigate HaaS for environmental computing. It addresses large-scale Web data in environmental science that can be mined for finding new information. It considers datasets available in publicly accessible repositories on the Web. It explains how knowledge discovered in the data mining process can be used for predictive analysis based solutions in interesting applications. More specifically, it presents automated text classification on energy-related data to enhance energy management with the broader impact of energy savings; it also explains the development of recommender engine prototypes targeting ecofriendly products to help the environment; furthermore, it describes property value prediction in real estate, helpful for decision support in urban planning. It dwells upon the utilization of HaaS-based technologies for conducting data mining and machine learning with suitable algorithms such as Naïve Bayes and Random Forest, particularly in the context of targeted applications.

The rest of this paper is organized as follows. Section 2 provides an overview of HaaS foundations by discussing MapReduce, Hadoop and Hive, mainly aimed for relatively new readers on this subject. Section 3 delves deeper into the data mining and machine learning facets of HaaS technologies. Sections 4, 5 and 6 respectively present interesting studies in the applications of automated text classification for energy savings, recommender engine development for ecofriendly products, and property prediction for decision support in urban planning, all deploying adequate HaaS-based methods and adapting them for these environmental computing applications. Section 7 gives the conclusions and future work.

## 2. OVERVIEW OF HAAS FOUNDATIONS

In this section, we provide a brief introduction of the classical HaaS foundations. We first explain Hadoop built using the MapReduce framework and then present Apache Hive as the data warehousing system for Hadoop that facilitates data storage and information retrieval. This is intended mainly for an audience of relatively new readers in the area.

### 2.1 Hadoop and MapReduce

Apache Hadoop is an open source implementation of the MapReduce framework and distributed file system. The MapReduce framework is designed to automatically divide applications into small fragments of work, each of which can be executed on any node in the cluster, handle node failures as well as "schedule inter-machine communication to make efficient use of the network and disks" [Dean and Ghemawat 2004], [White 2012]. Hadoop

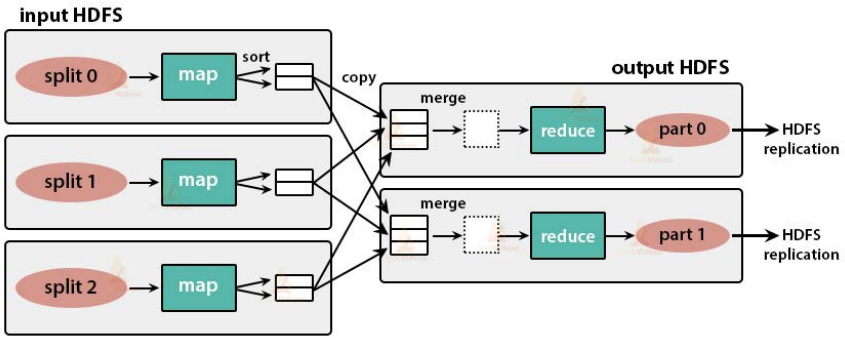


Fig. 1: A basic overview of Hadoop and MapReduce

MapReduce solution has been referred to in the literature as "one of the best solutions for batch processing and aggregating flat file data in recent years" [Gruenheid et al. 2011]. In [Dean and Ghemawat 2010] they emphasize high fault tolerance of MapReduce model for large jobs; its usefulness for handling data processing and data loading in a heterogeneous system; and ability to handle the execution of more complicated functions than are supported directly in SQL in parallel database systems. They also make several suggestions for working with MapReduce, such as avoiding using inefficient textual format for the data focusing on using HDFS binary format instead; taking advantage of natural indices such as timestamps in log file names; and leaving MapReduce output unmerged, as there is no benefit to merging it if next consumer is another MapReduce program. The overall architecture of Apache Hadoop based on the MapReduce framework is depicted in Fig. 1, adapted from [Dean and Ghemawat 2004] and [White 2012] sources.

Some interesting research has focused on the comparison of the MapReduce technology to parallel DBMS (Database Management Systems) [Floratou et al. 2012], [Stonebraker et al. 2010], [Chandra et al. 2019]. For example, the work in [Floratou et al. 2012] compares the performance of parallel Relational DBMS (RDBMS), specifically Microsoft SQL Server 2008 R2 Parallel Data Warehouse (PDW) to NoSQL database systems (such as Hive) in analytical workloads. The comparison is mainly with respect to heavy querying performed on big data. They conclude that although the NoSQL system provides for more flexibility and scales better with the increase in the data size, PDW employs "a robust and mature cost-based optimization and sophisticated query evaluation techniques that allow it to produce and run more efficient plans than the NoSQL system". Furthermore, they suggest that MapReduce-based systems adopt such techniques to improve query performance.

Work has also been done to develop new models for parallel and distributed data management and analysis systems. For example, research in [Bu et al. 2012] presents a programming model and architecture for iterative programs, i.e. HaLoop. The HaLoop service enhances MapReduce by offering support for programming in iterative applications, and moreover significantly augments efficiency via the task scheduler being *loop-aware* and through caching mechanisms. In other words, this is an extended and modified version of the Hadoop MapReduce framework which holds Hadoop's fault-tolerance properties, allows programmers to reuse existing mappers and reducers from conventional Hadoop

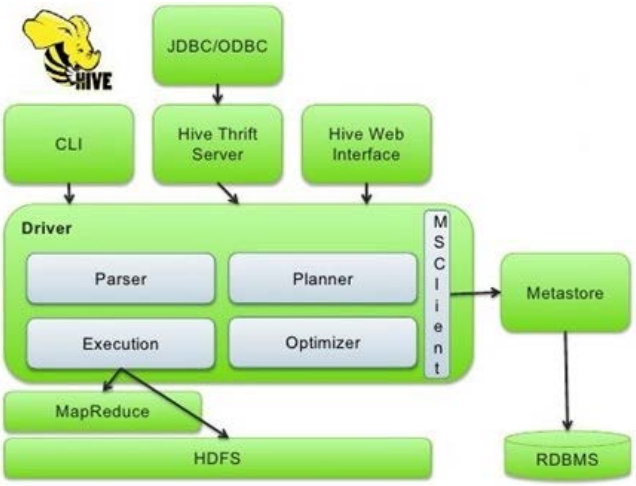


Fig. 2: Fundamentals of the architecture in Hive

applications. The parallel and distributed system supports large-scale iterative data analysis applications, including statistical learning and clustering algorithms. This makes it tremendously well-suited for big data storage, retrieval and analysis.

## 2.2 Apache Hive

Hadoop per se is not a database system. Hence, it is more advisable to use Hive - which was originally developed at Facebook, and then open-sourced [Thusoo et al. 2010]. Hive as an Apache open-source project, is intended to be a data warehouse system for Hadoop. It allows users to apply table definitions and structure on top of existing data files stored either directly in HDFS on in other data storage systems and further query the data in HiveQL, a SQL-like language. Hive queries are executed in MapReduce [Thusoo et al. 2010]. The architecture of Hive is presented in Fig. 2 as found in [Thusoo et al. 2010] and other sources in the literature.

In [Chandra et al. 2019], the authors present a thorough comparative study of Hive and MySQL data systems with respect to large-scale data management on the cloud. The study notes advantages and disadvantages of each system. They specifically note that Hive is designed for a high-latency, batch-oriented type of processing. This may not be an issue in analytical systems, since many companies may choose to pre-filter and pre-process their data before it is loaded into Hive warehouse for storage and analysis. Since Hive is Hadoop based it does not work well for ad-hoc queries and interactive responses. In this work, the authors discuss various approaches for data analytics, including: (A) processing the data on a local machine; (B) migrating the data from the local machine to the cloud; and (C) downloading the data directly to the cloud. The mechanism of downloading data directly to the cloud is depicted with an example in Fig.3 [Chandra et al. 2019]. Amazon Web Services (AWS) can be used for cloud computing, wherein the Elastic MapReduce (EMR) can be used for the direct download. The EMR instance can be launched in a convenient manner facilitating a direct download of the data. Since the EMR instances are Linux

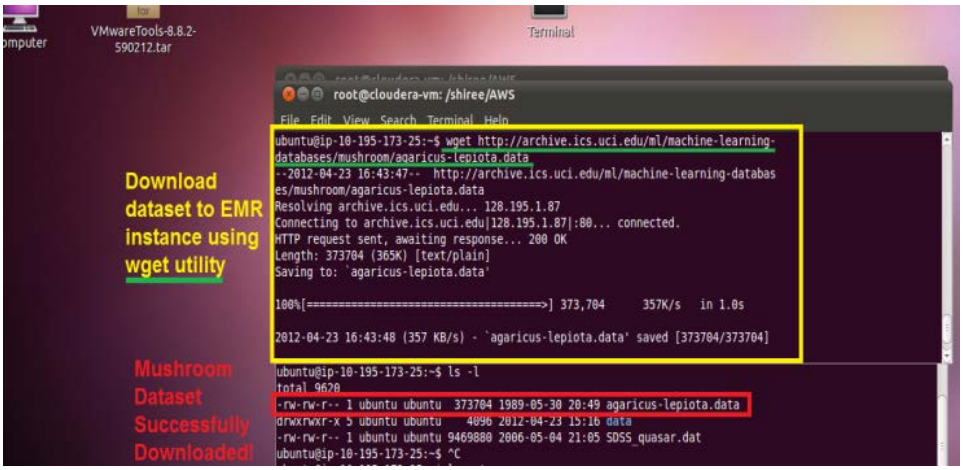


Fig. 3: Example of directly downloading datasets from the Web to the EMR (Elastic MapReduce) instance

based, a Linux utility called Wget can be used for a non-interactive download of data files to the EMR instance. In this manner, the local machine can be completely bypassed. Hence, this allows the processing of datasets that are much larger than the local machine can store, which is a huge advantage.

The work in [Abadi 2009] points out that "The infrequent writes in analytical database workloads, along with the fact that it is usually sufficient to perform the analysis on a recent snapshot of the data (rather than on up-to-the-second most recent data) makes the 'A', 'C', and 'I' (atomicity, consistency, and isolation) of ACID easy to obtain. Hence the consistency trade-offs that need to be made as a result of the distributed replication of data in transactional databases are not problematic for analytical databases". The research in [Abadi 2009] also calls for a hybrid solution for cloud based systems. The paper presents an analysis of deploying transactional and analytical database systems in the cloud. They conclude that the characteristics of the data and workloads of typical analytical data management applications are well-suited for cloud deployment. More specifically, (A) shared-nothing architecture scales well with the increasing data sizes; (B) since analysis can typically be performed on a snapshot of the data as opposed to in real-time, writes in analytical databases are infrequent and performed in a batch as such making consistency non-problematic for analytic databases; and (C) data security is usually a big concern for cloud based systems, in analytic databases, however, data can be pre-processed ahead of time leaving out particularly sensitive data or by applying an anonymization function. They further look into MapReduce systems and shared-nothing parallel databases as two platforms for data management on the cloud. They conclude that a hybrid solution is needed to satisfy all of the desired qualities of a cloud-based database system. They note the advantage of MapReduce to immediately read data off the file system and answer queries without any kind of loading stage.

## 3. MACHINE LEARNING AND DATA MINING WITH HAAS

MapReduce has been adapted by a large number of organizations as the technology for large-scale data storage and processing. Today enterprises not only store big data, but use it to discover knowledge. A number of articles have been published on the implementation of parallel data mining and machine learning algorithms on MapReduce framework [Ghoting et al. 2011], [Riondato et al. 2012]. Typical characteristics of data mining algorithms include the following: (A) they have iterative behavior, meaning they require performing multiple passes over the data; (B) the input data set contains numeric and discrete categorical attributes; (C) models are computed and represented with vectors, matrices and histograms, and other data structures.

In HaaS, one of the available tools for data mining and machine learning on the cloud includes Apache Mahout, an open source machine learning library from Apache focused on multiple areas of machine learning, such as recommender engines, clustering, and classification [Owen et al. 2011]. Machine learning algorithms are written in Java and some portions are built upon the Apache Hadoop distributed computation project. Apache Mahout offers a Java library to be used and adapted by developers. The framework of Apache Mahout is illustrated in Fig. 4, adapted from [Owen et al. 2011]. It encompasses multiple libraries for collaborative filtering, clustering, classification etc. as observed here.

Other work published on the subject includes HaLoop, an extended and modified version of the Hadoop MapReduce framework which holds Hadoop's fault-tolerance properties, and allows programmers to reuse existing mappers and reducers from conventional Hadoop applications. Its parallel and distributed system supports large-scale iterative data analysis applications, including statistical learning and clustering algorithms [Bu et al. 2012]. The research in [Ghoting et al. 2011] presents a portable infrastructure designed with the intention of parallelizing machine learning - data mining (ML-DM) computations. It provides built-in support to process data stored in a variety of formats. The system in this work called NIMBLE is being used by programmers at IBM Corporation. The article discusses deficiencies of using MapReduce for ML-DM computations: (1) there is a need for custom code to manage large computations (iterative and recursive); (2) when multiple computations can be performed inside a single MapReduce job, users are responsible for co-scheduling and pipelining these computations.

The paper in [Riondato et al. 2012] implements as a Java library a parallel algorithm, PARMA, which uses sets of small random samples for approximate frequent itemsets or association rule mining in Hadoop MapReduce. As noted in the article, since the algorithm is programmed in Java, there is feasibility of integration with Mahout. The work in [Pitchaimalai et al. 2010] implements and compares performance of Naïve Bayes algorithm in standard SQL queries, UDF (User Defined Functions), and MapReduce over different data set sizes. Here they find that SQL and UDF outperform MapReduce, although they emphasize the usefulness of MapReduce in large clusters of inexpensive hardware and its fault tolerances. They also call for future research into hybrid solutions combining SQL and MapReduce.

Twitter presents a case study [Lin and Kolcz 2012] of how machine learning tools are integrated into their analytics platform using Hadoop based Apache Pig, a high-level language for large-scale data analysis. It is interesting to note, that while Twitter had the goal of us-

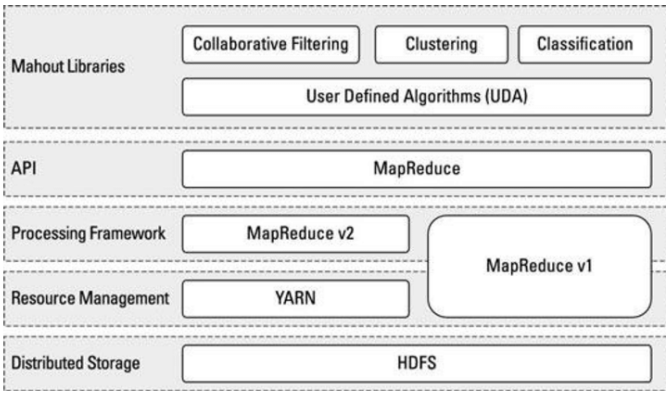


Fig. 4: An insight into the Mahout framework

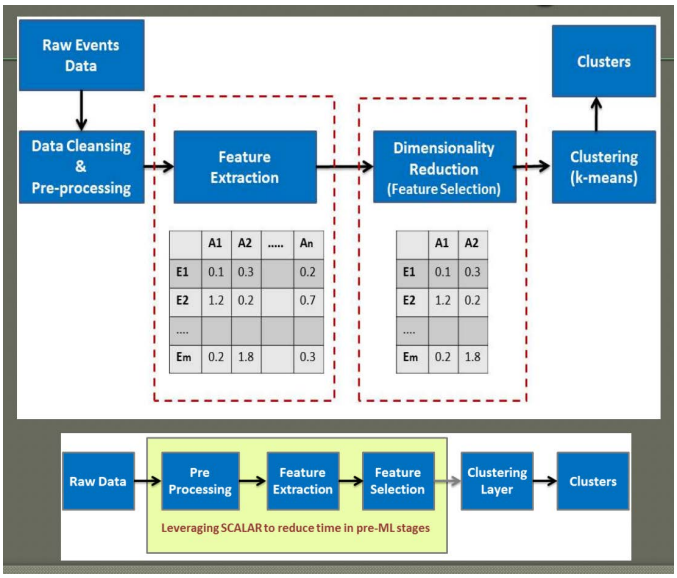


Fig. 5: Clustering process example for technologies in HaaS

ing off-the shelf solutions for their machine learning and analytics processes, Mahout did not work for them. This is because Twitter had already established production workflows for analytics and integrating Mahout would have required significant reworking of production code. They did however make their code open-source, thus providing the ability to encode data in Pig using the hashed encoding capabilities of Mahout (Sequence-Files of Vector Writables), and allowing training of logistic regression models.

Big data analytics including information retrieval, data mining and machine learning, encompassing cloud services and other technologies is described in a tutorial on "Scalable Learning Technologies for Big Data Mining" [de Melo and Varde 2015]. This covers the fundamentals of MapReduce, Hadoop and Hive, along with machine learning algorithms using packages such as Apache Mahout and Apache Spark, and mining of data

streams using Apache's Storm and Flink. It addresses comparisons of data storage capacity, processing speeds, retrieval efficiency and other aspects across various technologies. For example, a typical clustering process as seen in these technologies is depicted in Fig. 5. This is with particular reference to Apache Spark's MLlib (Machine Learning library), which constitutes a scalable framework with a variety of machine learning algorithms encompassing classification with Support Vector Machines (SVM), Naïve Bayes, Decision Trees; regression including linear regression and regression trees; collaborative filtering with alternate least squares (ALS); clustering with k-means; optimization with stochastic gradient descent (SGD) and limited memory BFGS (Broyden–Fletcher–Goldfarb–Shanno) algorithm; and dimensionality reduction with Singular Value Decomposition (SVD) and Principal Components Analysis (PCA). Spark has speeds much higher than Mahout. More advanced packages such as Storm and Flink are suitable for large-scale data streams.

Considering this background, a few studies with discussions are presented next on the utilization of such technologies in the realm of environmental computing. Hence, this article illustrates how HaaS can provide useful solutions in multiple applications here.

## 4. AUTOMATED TEXT CLASSIFICATION FOR ENERGY MANAGEMENT

In practice, text classification techniques have a wide range of applications spanning news, emails, homepages etc. Extremely large datasets that are becoming increasingly widespread require an increased amount of training data for improved accuracy. Mahout's algorithms are designed to be highly scalable. Thus, with the increase of the number of records required to train a model. the time and memory required for training a Mahout algorithm may not increase linearly, making scalable algorithms in Mahout widely useful [Owen et al. 2011]. Accordingly, this section focuses on textual data in the area of energy management within environmental computing. Mahout's Naïve Bayes algorithm is explored to describe an application that can be used for text classification on energy-related data.

Text can be collected from published literature such as research articles and white papers as well as other open sources on energy-related themes. This can be preprocessed so as to retain the plain text in the sources without images, infographics etc. It can then be subjected to training by deploying the Naïve Bayes algorithm to classify textual inputs. This can be classified into text categories pertaining to energy sources, including "Fossil Fuel", "Solar Panel", "Wind Turbine" and "other Source", in order to map to the traditional energy sources such as fossil fuels and more modern ones such as solar panels, wind turbines, while also maintaining a category of all other sources. The saved text files can be stored in HDFS on Amazon's Elastic Cloud Computing Cluster (EC3) [AWS 2021] running Hadoop 1.0.3 with Apache Mahout 0.7 installed.

Figs. 6 and 7, modified from [de Melo and Varde 2015], provide example snapshots illustrating the model training and the text classification output respectively. The model can be built upon Mahout libraries to do the following: (A) take a directory of text files as an input; (B) convert it to a sequence file format; and (C) generate TF-IDF weighted sparse vectors that are then classified. Maven can be used to manage dependencies, and to run the program on the cloud. For each text file in the input directory, the classification can contain the index of the category label and the associated score. The output is the category label which corresponds to the best score of the classification model.

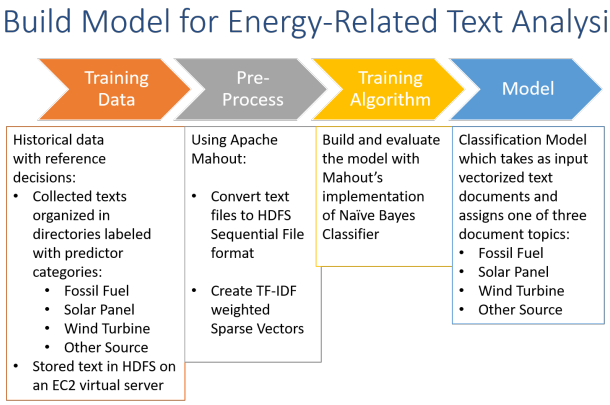


Fig. 6: Example of model development for text analysis on energy management

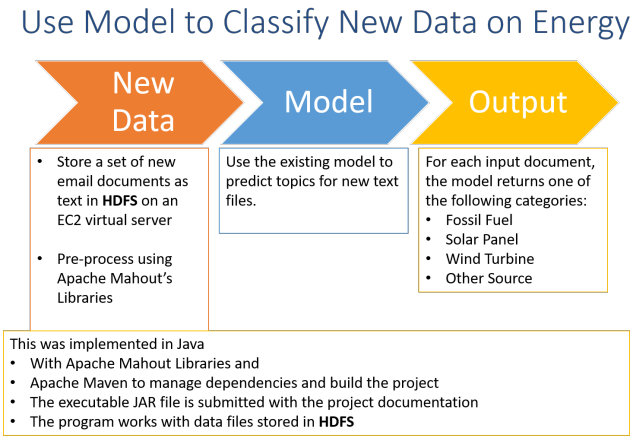


Fig. 7: Example of results from text classification on energy management

Using the interface, Mahout commands can be executed to train and test the classifier model. The process involves the following steps.

(1) Convert directories containing the text files into sequential file format.
(2) Create document vectors with Term Frequency – Inverse Document Frequency (TF-IDF) weighting (implemented by Mahout), to give the topic words more importance in the resulting document vectors.
(3) Use the Mahout command to split the training input into training and test sets with a specified split percentage.
(4) Train the algorithm using the training dataset to create a Naive Bayes classifier model.
(5) Test the performance with the test dataset by running Mahout's *testnb* command.

The predicted category can be used to automatically classify energy-related textual data into various categories. It can be used in conjunction with or supplementary to other tools

and techniques [Kraska et al. 2013], [Kiros et al. 2014], [Le and Mikolov 2014] in specific contexts. The text classification application is built upon Mahout Java libraries and runs on the cloud. It is useful to provide at-a-glance analysis and visualization of energy-related text as per its relevance to various energy sources. This can help stakeholders in energy management gain an understanding of energy trends in order to delve deeper into further details for making decisions, planning ahead and so forth. Hence, it offers automated analysis to facilitate decision-making and to gain more knowledge on energy management.

Relevant work here includes email classification using Mahout [Hammond and Varde 2013] which demonstrates the utilization of machine learning with appropriate classifiers for textual data in email. Another pertinent area is that of sentiment analysis, which can be applicable to numerous avenues of environmental computing, including energy management [Du et al. 2019], [Koupaei et al. 2020]. Likewise, the Web per se plays important roles in energy management as a whole [Shrestha and Varde 2023], [Ayala et al. 2019], considering the analysis of several data types including textual data. A myriad of research, such as the works mentioned here, can depict the importance of HaaS (without explicitly mentioning the term "HaaS"). For instance, some interesting work entails cloud technology for the greening of data centers [Pawlish et al. 2010] to make them more environment-friendly, and the adaptation of hybrid approaches with server-based and cloud-based technologies [Pawlish et al. 2014] for energy efficiency. While these works do not specifically mention HaaS, they are pertinent in cloud data management, especially with its relevance to energy savings and hence environmental computing.

## 5. RECOMMENDER DEVELOPMENT FOR ECOFRIENDLY PRODUCTS

Recommender engines are used in many e-commerce, retail, financial services, insurance and marketing systems. Accordingly, recommenders can be geared towards ecofriendly products to encourage more users to buy them, thus serving users' needs while also helping the environment. They can be implemented by using the item similarity recommender algorithm available in Mahout, hereinafter referred to as Mahout's Recommender. It is important to collect data on users' buying history, such that it is relevant to environmentally sustainable items, e.g. those built with recycled materials [Varde and Liang 2023], those that are renewable [Xu 2024], those that serve as sustainable office supplies [Ye et al. 2014] and so forth. Such data can be stored on the cloud, specifically Amazon S3 [AWS 2021]. It can be stored as a text file with transactional data on fields such as user id, item-id, product description, quantity purchased etc. Amazon Elastic MapReduce (EMR) service [AWS 2021] is beneficial to start a cluster of Hadoop nodes running Hive and *jdbc.HiveDriver* to connect to the EMR Master Node. HiveQL queries can then run in the command line interface or as a program to create a new table and insert formatted data from a select statement. This data serves as the input to Mahout's Recommender.

Once the data is pre-processed, it becomes the input to run Mahout's Recommender job, implemented with Apache Hadoop Distributed File System and MapReduce. It can initiate a series of more MapReduce jobs, as described in [Owen et al. 2011]. For example, input data derived with HiveQL as user-id, item-id, and preference, can be used to generate user vectors to compute a co-occurrence matrix, from which the algorithm derives recommendation vectors and eventually user recommendations. This can be stored in a compressed text file of the following format: *user-id [item1:pref1, item2:pref2, item3:pref3]* etc.

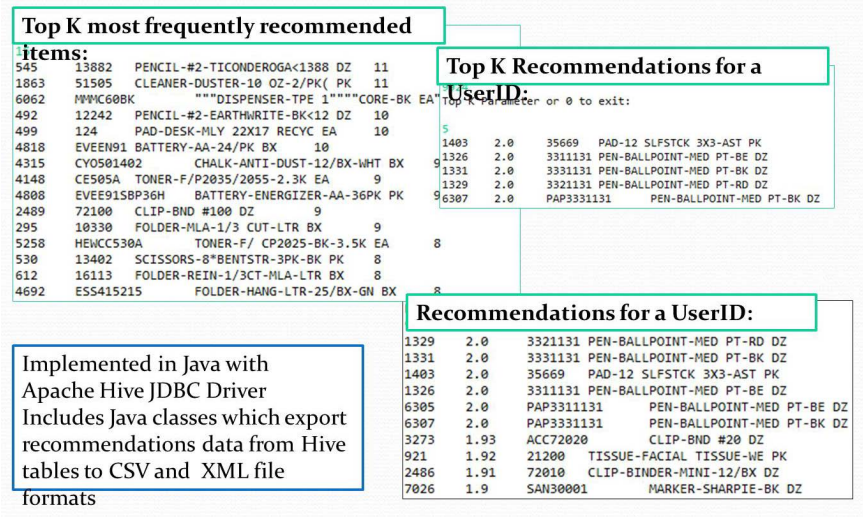


Fig. 8: Mahout's Recommender for designing a prototype recommender engine in ecofriendly products

Note that HiveQL is useful to parse the output into a Hive table partitioned by user-id to speed up querying the output and provide HiveQL queries for the following.

—Top-k recommended items in general

—Recommended items for a given user

—Top-k recommended items for a given user

Fig. 8, referenced from [Hammond and Varde 2013], shows a sample of a recommender engine prototype for ecofriendly products. This indicates recommendations in all the categories mentioned herewith, i.e. the top-k recommended items based on most frequently occurring ones in general, the recommended items for a given user based on the user-ID, and the top-k recommended items for a given user. In each of these categories, the first column represents the recommended item-id, followed by the calculated user preference for that item, product code and description.

As mentioned previously, Hive is a data warehouse system and is not intended for online transaction processing where fast response time matters. Hence, it is interesting to experiment with additional modules e.g. to convert the recommender engine output into a CSV (comma separated variable) file so it can be used in another application with an underlying RDBMS, e.g. MySQL, SQL Server or Oracle. User recommendations can be produced in XML file format that can conveniently be used in web applications. A recommender prototype thus built can potentially complement other pertinent applications, e.g. [Afsar et al. 2021], [Gandhe et al. 2018], [Lourenco and Varde 2020], [Zhang et al. 2021], especially where big data is involved via large scale systems. This is highly relevant to environmental computing and illustrates interesting use cases for HaaS in this avenue.

## 6. PROPERTY PREDICTION FOR DECISION-MAKING IN URBAN PLANNING

The real estate market provides substantial pathways for big data mining in urban planning, which is an important facet of environmental computing. Data in this the urban planning domain is ever-growing with more real estate properties being developed every year. It is thus important to retrieve significant information about property prices in the context of real estate in given neighborhoods and hence discover knowledge to make predictions about future prices, in order to assist decision-making urban planning in general. The Mahout library includes sequential and MapReduce (parallel) implementations of the Random Forests (RF) classifier which can accept data with numeric and categorical attributes as inputs. Hence, this classifier can be useful with respect to property value prediction. Based on analyzing real estate data in urban planning, Mahout's RF classifier can help to estimate property value ranges, given the building area, lot area, zoning, land use, street name, zip code etc. Since real estate data is often in a fixed file format, it can first be stored on the cloud, and thereafter be used to execute Hive select queries for training and testing.

Mahout's commands can be used to generate the dataset that describes the data and stores the labels to be predicted. The BuildForest command can help to build a new RF model, and the *TestForest* command is beneficial to evaluate the model. The execution of these commands is explained in detail in [Owen et al. 2011]. The *TestForest* class can produce an output file that entails the following. For each line of the data input file, it lists the index of the prediction label. The *TestForest* class can be used with the RF dataset description file and the model file to classify new data as mentioned here.

(1) For sequential classification, a modified version of the *TestForest* class is created; thus for each input line, the output file has the the prediction label (instead of the index value) followed by the data input line.

(2) A code snippet can merge the input and output files line by line, writing the predicted category label followed by the input data to a new file; the merged file looks similar to the output and the program can be used to interpret the output of MapReduce or a sequential implementation of RF.

(3) Another program can classify one input instance at a time so that it encompasses a string of attributes in the same format used to train the model, and the output of this program is the predicted category label, e.g. property value range.

Fig. 9, adapted from [Hammond and Varde 2013] shows sample classification for data records. As seen here, the predicted value range in a highlighted record is "1M+", i.e. "more than one million" which can refer to properties often classified as mansions; while another predicted range for a different record shows "500-749K", i.e. "more than 500,000 and less than 750,000 which reflects very high income neighborhoods (but typically not classified as mansions). As these property prices can be estimated using big data mining with cloud services, they pave the way to develop tailor-made tools for such applications, which can be useful for decision support in urban planning. As is well-known, the real estate market is huge. People usually invest in residential and commercial real estate for their housing and workplace respectively, in addition to rental purposes. Hence, prior estimation of property prices is beneficial to assist decision-making for buyers, sellers, brokers, governmental bodies, city planning agencies etc.

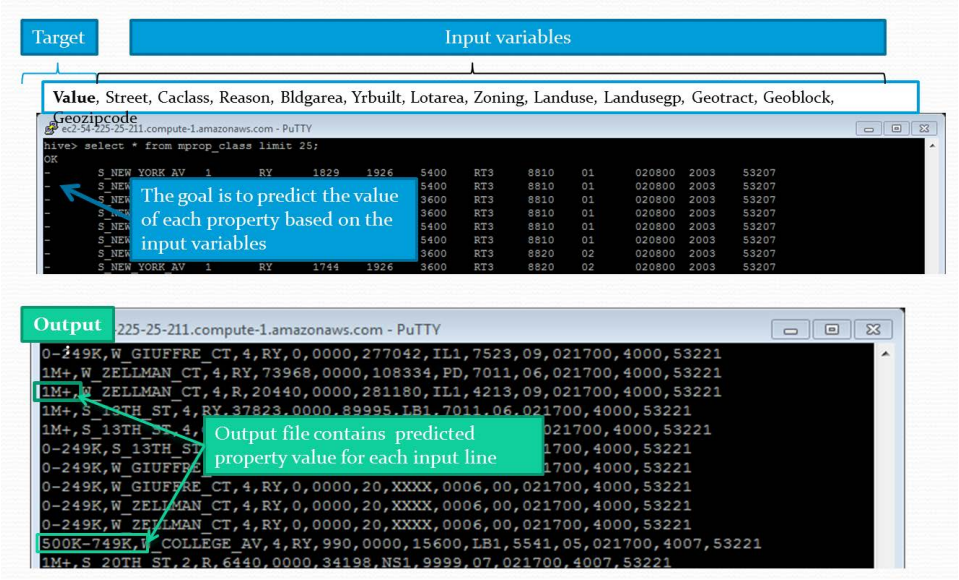


Fig. 9: Example for Estimation of Real Estate Property Values in Urban Planning

Note that work addressed here can be potentially useful in other suitable applications, hence being orthogonal to the literature [Yu et al. 2021], [Soltani et al. 2021]. Some applications can leverage smart applications in urban planning [Rathore et al. 2016], [Du et al. 2017], [Anthopoulos and Vakali 2012] with facets of smart governance [Puri et al. 2018], [Lopes 2017], [Tomor et al. 2019] that is an important characteristic of smart cities, hence contributing to a smart planet. All of these aspects are vital to environmental computing, where HaaS can be considerably useful, as depicted here.

## 7. CONCLUSIONS

This article addresses HaaS technologies available over the cloud for big data mining in environmental computing. More specifically, it explores Apache Hive and Mahout with its multiple facets such as HiveQL, Mahout Recommender etc. A few highlights in environmental computing are as follows.

—A Naïve Bayes classifier in HaaS can be used to build a text classification application for energy management to automatically classify energy-related textual data from different sources into pertinent categories, such as fossil fuels, solar panels etc.

—An item similarity recommendation algorithm in HaaS can rank and recommend potential ecofriendly products that a user may be interested in purchasing based on mining relevant data on various sustainable product categories.

—A Random Forests classifier in HaaS can be useful to build a model to estimate property value ranges in real estate data useful for decision support in urban planning.

—Applications can be designed using HaaS on the cloud without incurring huge capital investments of initial server acquisition, recurrent maintenance, in-house licensed software purchases, periodic updates, and other major expenses.

Thus, HaaS being in line with cloud technologies can offer a pay-as-you-go model where developers and stakeholders can only pay for what they need while also taking advantage of some open access freeware provided over the cloud. Accordingly, many interesting domain-specific applications can be developed analogous to those in environmental computing mentioned here.

Some limitations of HaaS could relate to the potential lack of consistent availability, privacy and confidentiality issues, cybersecurity concerns and other risks that could seem to make developers and stakeholders more cautious. These and other limitations can potentially offer the scope for future work on enhanced research in HaaS technologies. In addition, further research could possibly encompass proposing customized software for targeted applications based on the results of multiple studies. Advanced studies can be conducted that encompass a cluster of distributed machines on the cloud along with detailed comparisons of server versus cloud implementations in domain-specific settings.

A next generation paradigm is *edge computing* [Yeung 2022], a networking technology that facilitates remote devices to conduct data processing at the "edge" of the network, by the device / local server. Edge computing thrives on cloud computing and gets closer to the edge of the respective machine, hence being considered an evolution of cloud computing. The edge computing paradigm can be further explored with respect to *edge AI* [IBM 2024]. As the name implies, *edge AI* entails deploying artificial intelligence within edge computing; it adapts neural networks and deep learning for model-training for recognizing, describing, and classifying the concerned data. Hence, edge computing and in particular edge AI can be studied on a deeper level with respect to their applications in numerous domains, including environmental computing. Multiple avenues such as these offer much scope for future work.

In sum, this article discusses joint work in the areas of cloud technologies (more specifically HaaS) and environmental computing. It targets the computer science and engineering communities with specific attention to AI and data science in particular due to its emphasis on data mining and machine learning. It can also benefit professionals from areas such as business management and other application domains in addition to environmental computing, since many such users would seek to optimize solutions in the concerned applications for efficiency and hence cost-effectiveness. It opens doors to further exploration, including the use of more advanced technologies such as edge computing with edge AI. On the whole, this article highlights Hadoop-as-a-Service with its substantial importance.

## ACKNOWLEDGMENTS

The author Dr. Aparna Varde thanks her recent grants from the NSF (National Science Foundation) of the United States, grant numbers 2104742 and 2018575, as well as another grant from the NOAA (National Oceanic and Atmospheric Administration) through their New Jersey Sea Grants Consortium. The relevant cloud computing work of her former MS students from Montclair State University, especially that of Klavdiya Hammond and Shireesha Chandra, is also being gratefully acknowledged here.

## REFERENCES


ABADI, D. J. 2009. Data Management in the Cloud: Limitations and Opportunities. *IEEE Data Engineering Bulletin*.

AFSAR, M. M., CRUMP, T., AND FAR, B. H. 2021. Reinforcement learning based recommender systems: A survey. *CoRR abs/2101.06286*.

ANTHOPOULOS, L. G. AND VAKALI, A. 2012. Urban planning and smart cities: Interrelations and reciprocities. In *The Future Internet: Future Internet Assembly 2012: From Promises to Reality 9*. Springer Berlin Heidelberg, 178–189.

ARAKI, K., OHASHI, K., YAMAZAKI, S., HIROSE, Y., YAMASHITA, Y., YAMAMOTO, R., MINAGAWA, K., SAKAMOTO, N., AND YOSHIHARA, H. 2000. Medical markup language (mml) for xml-based hospital information interchange. *Journal of medical systems 24*, 195–211.

AWS. 2021. Amazon web services. https://aws.amazon.com/.

AYALA, I., AMOR, M., AND FUENTES, L. 2019. An energy efficiency study of web-based communication in android phones. *Scientific Programming 2019*, 1, 8235458.

BU, Y., HOWE, B., BALAZINSKA, M., AND ERNST, M. D. 2012. The HaLoop approach to large-scale iterative data analysis. *VLDB Journal 21*, 2, 169–190.

CHANDRA, S., VARDE, A. S., AND WANG, J. 2019. A Hive and SQL Case Study in Cloud Data Analytics. In *IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference, UEMCON*. IEEE, 112–118.

DE MELO, G. AND VARDE, A. S. 2015. Scalable Learning Technologies for Big Data Mining. In *Database Systems for Advanced Applications DASFAA*. Lecture Notes in Computer Science, vol. 9050. Springer, xvii–xviii.

DEAN, J. AND GHEMAWAT, S. 2004. MapReduce: Simplified data processing on large clusters. In *Proceedings of the 6th Conference on Symposium on Opearting Systems Design & Implementation - Volume 6*. OSDI'04. 10–10.

DEAN, J. AND GHEMAWAT, S. 2010. MapReduce: a flexible data processing tool. *Communications of the ACM 53*, 1, 72–77.

DU, X., KOWALSKI, M., VARDE, A. S., DE MELO, G., AND TAYLOR, R. W. 2019. Public opinion matters: mining social media text for environmental management. *ACM SIGWEB 2019*, Autumn, 5:1–5:15.

DU, X., LIPORACE, D., AND VARDE, A. S. 2017. Urban legislation assessment by data analytics with smart city characteristics. In *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*. IEEE, 20–25.

FLORATOU, A., TELETIA, N., DEWITT, D. J., PATEL, J. M., AND ZHANG, D. 2012. Can the Elephants Handle the NoSQL Onslaught? *Proc. VLDB Endow. 5*, 12, 1712–1723.

GANDHE, K., VARDE, A. S., AND DU, X. 2018. Sentiment Analysis of Twitter Data with Hybrid Learning for Recommender Applications. In *IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference, UEMCON*. IEEE, 57–63.

GEVAERT, C. M. 2022. Explainable ai for earth observation: A review including societal and regulatory perspectives. *International Journal of Applied Earth Observation and Geoinformation 112*, 102869.

GHOTING, A., KAMBADUR, P., PEDNAULT, E., AND KANNAN, R. 2011. NIMBLE: A Toolkit for the Implementation of Parallel Data Mining and Machine Learning Algorithms on Mapreduce. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 334–342.

GRUENHEID, A., OMIECINSKI, E., AND MARK, L. 2011. Query optimization using column statistics in Hive. In *International Database Engineering and Applications Symposium (IDEAS)*. ACM, 97–105.

HADOOP. 2021. 10 best cloud service providers. https://www.hdfstutorial.com/blog/hadoop-cloud-service-providers-2/.

HAMMOND, K. AND VARDE, A. S. 2013. Cloud Based Predictive Analytics: Text Classification, Recommender Systems and Decision Support. In *Proceedings of the 2013 IEEE 13th International Conference on Data Mining Workshops*. ICDMW '13. IEEE Computer Society, Washington, DC, USA, 607–612.

IBM. 2024. What is edge ai? https://www.ibm.com/topics/edge-ai.

KARTHIKEYAN, D., VARDE, A. S., AND WANG, W. 2020. Transfer learning for decision support in covid-19 detection from a few images in big data. In *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 4873–4881.

KIROS, R., SALAKHUTDINOV, R., AND ZEMEL, R. S. 2014. Multimodal neural language models. In *International Conference on Machine Learning (ICML)*. 595–603.

KO, H., LU, Y., YANG, Z., NDIAYE, N. Y., AND WITHERELL, P. 2023. A framework driven by physics-guided machine learning for process-structure-property causal analytics in additive manufacturing. *Journal of Manufacturing Systems 67*, 213–228.

KOUPAEI, D. M., SONG, T., CETIN, K. S., AND IM, J. 2020. An assessment of opinions and perceptions of smart thermostats using aspect-based sentiment analysis of online reviews. *Building and Environment 170*, 106603.

KRASKA, T., TALWALKAR, A., DUCHI, J. C., GRIFFITH, R., FRANKLIN, M. J., AND JORDAN, M. I. 2013. MLbase: A Distributed Machine-learning System. In *Conference on Innovative Data Systems Research (CIDR)*.

LE, Q. V. AND MIKOLOV, T. 2014. Distributed representations of sentences and documents. In *International Conference on Machine Learning (ICML)*. 1188–1196.

LIN, J. J. AND KOLCZ, A. 2012. Large-scale machine learning at Twitter. In *ACM SIGMOD International Conference on Management of Data*. ACM, 793–804.

LIU, Y., WEI, X., XIAO, J., LIU, Z., XU, Y., AND TIAN, Y. 2020. Energy consumption and emission mitigation prediction based on data center traffic and pue for global data centers. *Global Energy Interconnection 3,* 3, 272–282.

LOPES, N. V. 2017. Smart governance: A key factor for smart cities implementation. In *IEEE ICSGSC*. IEEE, 277–282.

LOURENCO, J. AND VARDE, A. S. 2020. Item-Based Collaborative Filtering and Association Rules for a Baseline Recommender in E-Commerce. In *IEEE International Conference on Big Data*. IEEE, 4636–4645.

MCLVOR, K. 2016. The DevOps Pipeline. Tech. rep., QA Limited, Berkshire, England. Feb.

OWEN, S., ANIL, R., DUNNING, T., AND FRIEDMAN, E. 2011. *Mahout in Action*. Manning Publications.

PAWLISH, M., VARDE, A. S., AND ROBILA, S. A. 2010. A decision support system for green data centers. In *ACM CIKM Workshop for PhD students in information and knowledge management*. 47–56.

PAWLISH, M., VARDE, A. S., ROBILA, S. A., AND RANGANATHAN, A. 2014. A call for energy efficiency in data centers. *ACM SIGMOD Record 43,* 1, 45–51.

PAWLISH, M. J. AND VARDE, A. S. 2018. The DevOps Paradigm with Cloud Data Analytics for Green Business Applications. *ACM SIGKDD Explorations 20,* 1, 51–59.

PITCHAIMALAI, S. K., ORDONEZ, C., AND GARCIA-ALVARADO, C. 2010. Comparing SQL and MapReduce to compute Naive Bayes in a single table scan. In *International ACM CIKM Workshop on Cloud Data Management, CloudDB*. ACM, 9–16.

PURI, M., VARDE, A., DU, X., AND DE MELO, G. 2018. Smart governance through opinion mining of public reactions on ordinances. In *IEEE ICTAI*. 838–845.

RATHORE, M. M., AHMAD, A., PAUL, A., AND RHO, S. 2016. Urban planning and building smart cities based on the internet of things using big data analytics. *Computer networks 101*, 63–80.

RIONDATO, M., DEBRABANT, J. A., FONSECA, R., AND UPFAL, E. 2012. PARMA: A Parallel Randomized Algorithm for Approximate Association Rules Mining in MapReduce. In *ACM International Conference on Information and Knowledge Management (CIKM)*. 85–94.

SHRESTHA, S. AND VARDE, A. S. 2023. Roles of the web in commercial energy efficiency: Iot, cloud computing, and opinion mining. *ACM SIGWEB 2023,* Autumn, 1–16.

SOLTANI, A., PETTIT, C., HEYDARI, M., AND AGHAEI, F. 2021. Housing price variations using spatio-temporal data mining techniques. *Journal of Housing and the Built Environment*.

STONEBRAKER, M., ABADI, D. J., DEWITT, D. J., MADDEN, S., PAULSON, E., PAVLO, A., AND RASIN, A. 2010. MapReduce and parallel DBMSs: friends or foes? *Communications of the ACM 53,* 1, 64–71.

TANCER, J. AND VARDE, A. S. 2011. The Deployment of MML for Data Analytics over the Cloud. In *IEEE International Conference on Data Mining (ICDM) - Workshops*. 188–195.

TETKO, I. V., ENGKVIST, O., KOCH, U., REYMOND, J.-L., AND CHEN, H. 2016. Bigchem: challenges and opportunities for big data analysis in chemistry. *Molecular informatics 35,* 11-12, 615–621.

THUSOO, A., SARMA, J. S., JAIN, N., SHAO, Z., CHAKKA, P., ZHANG, N., ANTHONY, S., LIU, H., AND MURTHY, R. 2010. Hive - A petabyte scale data warehouse using Hadoop. In *International Conference on Data Engineering (ICDE)*. 996–1005.

TOMOR, Z., MEIJER, A., MICHELS, A., AND GEERTMAN, S. 2019. Smart governance for sustainable cities: Findings from a systematic literature review. *Journal of urban technology 26,* 4, 3–27.

VARDE, A., RUNDENSTEINER, E., JAVIDI, G., SHEYBANI, E., AND LIANG, J. 2007. Learning the relative importance of features in image data. In *2007 IEEE 23rd international conference on data engineering workshop*. IEEE, 237–244.

VARDE, A. S. AND LIANG, J. 2023. Machine learning approaches in agile manufacturing with recycled materials for sustainability. *AAAI 2023 conference, Bridge Program, arXiv:2303.08291*.

WHITE, T. 2012. *Hadoop: The Definitive Guide*. O'Reilly Media, Inc.

XU, L. 2024. Renewable resource of aggregation-induced emission materials: From photophysical mechanisms to biomedical applications. *Coordination Chemistry Reviews 506*, 215701.

YE, H., CHAN, J., BHATTI, R., ET AL. 2014. Eco-friendly office supplies, pens and markers. *Open Collections, University of British Columbia*.

YEUNG, T. 2022. What's the difference between edge computing and cloud computing?, nvidia. https://blogs.nvidia.com/blog/difference-between-cloud-and-edge-computing/.

YU, Y., LU, J., SHEN, D., AND CHEN, B. 2021. Research on real estate pricing methods based on data mining and machine learning. *Neural Computing and Applications 33*, 3925–3937.

ZHANG, Q., LU, J., AND JIN, Y. 2021. Artificial intelligence in recommender systems. *Complex Intelligent Systems 7*, 439–457.

---

**Dr. Aparna Varde** is a tenured Associate Professor in the School of Computing (SoC) at Montclair State University (MSU), NJ, USA. She is an Associate Director of the Clean Energy and Sustainability Analytics Research Center (CESAC) at MSU. She has served as the Inaugural Associate Director for Graduate Studies and Research in SoC at MSU for a year. Dr. Varde has been a visiting researcher at Max Planck Institute for Informatics, Saarbrücken, Germany multiple times. Her work spans AI, Machine Learning, Data Mining, Databases, Environmental Computing, and Computational Linguistics. Her honors include 9 best paper awards at IEEE conferences and other venues, and 3 more notable mentions. She is Doctoral Faculty in the PhD Program in Environmental Science and Management at Montclair. Dr. Varde has over 150 publications (journals, conferences, book chapters, edited volumes) by IEEE, ACM, AAAI, Springer etc. She has been a dissertation advisor, committee member, and mentor for around 10 PhD students at Montclair State University (including a Fulbright Scholar) and an external examiner / referee for 4 PhD students worldwide (including Queensland University of Technology, Australia). Dr. Varde has been a panelist for NSF, PC member at conferences, and reviewer / editorial board member for journals, by IEEE, ACM, Elsevier etc. Her research is funded by grants from organizations such as PSEG, NSF and NOAA. She has generated more than 2 million dollars of external funding in various capacities. Dr. Varde is classified as an outstanding researcher by the Citizenship and Immigration Services, USA.