Sudan University of Science and Technology

College of Graduate Studies

Faculty of Computer Science &Information Technology

Parallel Support Vector Machine for big data Classification

تصنيف البيانات الضخمة باستخدام نظام الدعم الآلي المتوازي

This Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of Doctor of Philosophy in Computer Science at Sudan University of Science & Technology

By

Iatimad Mohammed Sati Abdelkarim

Supervisor

Professor Dr. Johnson Agbinya

November 2020

DEDICATION

I would like to dedicate this work with special feeling of gratitude to my beloved parents, Mohamed Satti God bless his soul and Hussna Hassan. I would not reach this level without you support. I dedicate this work to my brothers and sisters who keep encouraging me and always be happy about my achievement more than me. I dedicate this work to my small family my husband Amjad Hashim, my son Raid, and my daughters Limar and NourAlhuda.

ACKNOWLEDGEMENT

I am much thankful to our supervisor and all professors who help us in writing our research and document at the technical level to search methodology, data collection, analysis of this research results, registration, finding of the data, and discussing the results and outcome. Thanks to everyone who was involved in this in the validation of this thesis.

I also would like to thank my supervisor, Professor Johnson Agbinya, for his patience, guidance, and constant encouragement. It has been lucky to work with him. Many thanks to Prof. Izzaldin Mohammed Osman, who be as our Spiritual Father.

I would like to thank my family for the support they provided me throughout my entire life and mainly through the process of pursuing a Ph.D. degree. Because of their unconditional love and prayers, I have the chance to complete this thesis. Special and many thanks to my father Mohamed Sati may God have mercy on him and forgive him. Special and many thanks to my husband, Amjad Hashim, for encouraging and supporting all these years. Many thanks go to my brother Mohamed for his always help. I would like to take this opportunity to say warm thanks to all my beloved friends, who have been so supportive along the time of doing my thesis.

ABSTRACT

With the rapid growth of data in various fields, big data analysis is considered a great challenge for traditional management systems and scientists. This research deals with big data analysis using parallel computing through some algorithms for machine learning methods. This research deals with big data analysis using parallel computing through some algorithms. A framework of Parallel SVMs based MapReduce is implemented on different datasets to perform supervised classification. Support Vector Machines are an excellent example of the commonly used methods for producing classification problems. It is a suitable classifier machine learning because of its generalization ability and expertise to classify big data accurately. However, the traditional SVM is not appropriate for huge datasets due to its high computational complexity.

This research studies the SVM algorithm and Parallel Support Vector Machine (PSVMs) and their applications in different big data fields. The implementation of PSVM is done in the Hadoop cluster running in the HPC center in Sudan. Three models are implemented in four datasets for classification. The PSVM is applied to real data. Then the k-means clustering is combined with the support vector machine. The real water quality dataset from the ministry of health and different water stations in Sudan (2006-2017) is used to classify whether the water is suitable for drinking or not. The Adult dataset is used to classify the income of a person. The diabetes data set is used to classify whether the patient has diabetes or not. The cover type dataset is used to classify seven wilderness areas located in the Roosevelt National Forest of northern Colorado. The numerical experiment applying the PSVM is compared with k- means clustering applied to SVM and SVM frameworks. The results showed that applying the parallel support vector machine gives the highest accuracy and positively reduces computation time. The performance is compared using time-consuming accuracy.

المستخلص

مع النمو السريع للبيانات في مختلف المجالات، يعتبر تحليل البيانات الضخمة تحديًا كبيرًا لأنظمة الإدارة التقليدية والعلماء. يتناول هذا البحث تحليل البيانات الضخمة باستخدام الحوسبة المتوازية من خلال بعض الخوارزميات لطرق التعلم الآلي. يتناول هذا البحث تحليل البيانات الضخمة باستخدام الحوسبة المتوازية من خلال بعض الخوارزميات. يتم تنفيذ إطار عمل المابريديوس المتوازي القائم على المابريديوس على مجموعات بيانات مختلفة لإجراء تصنيف خاضع للإشراف. تعد آلات المتجهات الداعمة مثالًا ممتازًا للطرق الشائعة الاستخدام لإنتاج مشاكل التصنيف. إنه مصنف مناسب للتعلم الآلي بسبب قدرته على التعميم وخبرته لتصنيف البيانات الضخمة بدقة. ومع ذلك، فإن ناقلات الدعم الالي التقليدية ليس مناسبًا لمجموعات البيانات الضخمة نظرًا لتعقيده الحسابي العالي .يدرس هذا البحث خوارزمية ناقلات الدعم الالي وخوارزمية الدعم الالي للمتجهات المتوازية (PSVM) وتطبيقاتهما في مجالات البيانات الضخمة المختلفة. يتم تنفيذ خوارزمية الدعم الالي للمتجهات المتوازية في مجموعة الهدوب (Hadoop) التي تعمل في مركز حاسوب عالي الأداء (HPC) في السودان. يتم تنفيذ ثلاثة نماذج في أربع مجموعات بيانات من أجل التصنيف. يتم تطبيق خوارزمية الدعم الالي للمتجهات المتوازية على البيانات الحقيقية. ثم يتم دمج مجموعة الوسائل k مع آلة ناقلات الدعم. تستخدم مجموعة بيانات جودة المياه الحقيقية من وزارة الصحة ومحطات المياه المختلفة في السودان (2006-2017) لتصنيف ما إذا كانت المياه صالحة للشرب أم لا. تُستخدم مجموعة بيانات الكبار لتصنيف دخل الشخص. تُستخدم مجموعة بيانات مرض السكري لتصنيف ما إذا كان المريض مصابًا بالسكري أم لا. تُستخدم مجموعة بيانات نوع الغلاف لتصنيف سبع مناطق برية تقع في غابة روزفلت الوطنية في شمال كولورادو.

تتم مقارنة التجربة العددية التي طبقت على الثلاث نماذج. أظهرت النتائج أن تطبيق آلة متجه الدعم المتوازي يعطي أعلى دقة ويقلل بشكل إيجابي من وقت الحساب. تتم مقارنة الأداء باستخدام دقة استغراق الوقت الطويل .

TABLE OF CONTENTS

Dedication…….….	I
Acknowledgements……	II
Abstract…….….	III
المستخلص……	IV
Table of contents…….	V
List of Figures…….….	VIII
List of Tables……...	XI
List of Abbreviation …….……...	XII
CHAPTER ONE **Introduction**	
1.1. Background…….….	1
1.2. Problem Statement……..…	5
1.3. Research Scope…….……	5
1.4. Research Objectives and Contributions…….……...	6
1.5. Research Contributions……..…	7
1.6. Research Organization…….….	7
CHAPTER TWO **Literature Review**	
2.1 Data Mining…….……..……...	9
2.2 Machine Learning…….……	10
2.3 Classification techniques …….…….....	11
2.3.1 Big data ……...….	11
2.3.1.1 Machine-generated data……...…….	12
2.3.1.2Web and social data (Transaction data) …….…….	13
2.3.1.3 Web and social data ……	13
2.3.1.4 Human-generated Data…….……...	13
2.3.1.5 Web and social data (Biometrics)…….……...	13

2.3.1.6 Machine-generated data (Transaction data)……	13
2.3.2 Support Vector Machine (SVM)	14
2.3.3 K-means clustering algorithm…….	18
2.3.4 Parallel Support Vector Machine PSVM)……	19
2.3.5 Hadoop Framework……..……..	20
2.3.6 Integration of Rapid-Miner and Hadoop……..	25
2.4 Related Work ……	26
2.4.1 MapReduce ……	26
2.4.2 parallel implementation…….	30
2.4.3 Hadoop and Radoop ……	37
2.5 Chapter summary…….	41
CHAPTER THREE **Research Methodology**	
3.1 Methodology Framework of proposed Implementation…….……	42
3.2 Dataset Description …….……...	43
3.3 Dataset Loading…….….	45
3.4 Introduction to RapidMiner Environment…….….	45
3.5 Data Preprocessing…….	46
3.6 K-mean Clustering Analysis…….….	48
3.7 Support Vector Machine……	53
3.8 k-means clustering applied to SVM……...	55
3.9 Parallel Support Vector Machine (PSVM)…….	60
3.10 Hadoop Implementation with PSVM…….	60
3.11 Parallel SVM Based Classification……	61
3.12 MapReduce…….	62

3.13 Description of PSVM Algorithm using MapReduce in Hadoop……	64
3.14 Summary…….	64
CHAPTER FOUR **Experimental Result & Discussions**	
4.1. The implementation of SVM……	65
4.2. The implementation of SVM with Gamma=1.5…….	68
4.3. Third Experiment The implementation of k-means clustering applied to SVM……	71
4.4. Result Comparison of result of SVM and k-mean applied to SVM……	74
4.5. Forth Experiment of PSVM on Hadoop cluster…….	77
4.6. The Efficiency of the Parallel SVM……...	79
4.7. First Experiment versus forth Experiment ……...	80
4.8. Result comparison for, First, Second, Forth Experiments……..	83
4.9. Chapter summary……	86
CHAPTER FIVE **Conclusion and future work**	
Conclusions……	88
Future work…….	90
References……	91
Appendix I:	100
Appendix II:	105

LIST OF FIGURES

Figure2.1: Big Data Architecture……..	12
Figure2.2: Support vector machine implementations…….	14
Figure2.3: SVM maximizes the margin between two classes. Left. Small margin. Right. Large margin [14].…….....	15
Figure2.4: Support vector machine classifier…….	16
Figure2.5: Multi-class classification…….....	17
Figure2.6: K-means clustering	19
Figure2.7: Training flow of Parallel SVM…….....	20
Figure 2.8: Architecture of Hadoop Cluster……..	21
Figure2.9: HDFS Architecture.…….	22
Figure 2.10: Overview of MapReduce system……..	23
Figure 2.11: Structure and Flow of PSVM Algorithm using MapReduce…….	24
Figure 2.12: The Architecture of the Rapid-Miner - Hadoop integration……..	25
Figure 3.1 (a) : Flow of classification task of single SVM ……	43
Figure 3.1(b): Flow of classification task of the k-means combine to SVM.	43
Figure 3.1(c): Flow of classification task of PSVM	43
Figure3.2: Left: two classes showing a small margin Right: two classes but large margin...	49
Figure 3.3: Linear separation of the datapoints into two classes……	52
Figure 3.4: parallel Hyperplane…….….	54
Figure 3.5: (a) Original data set (b) after one iteration of data removal…….	55
Figure 3.6: training flow of parallel Support Vector Machine…….	57
Figure 3.7. PSVM with Hadoop cluster using Radoop extinction……..	60
Figure 3.8: Parallel SVM Based classification algorithms……..	61
Figure 3.9: The MapReduce framework……..	62
Figure 3.10: MapReduce Design on Hadoop…….	63
Figure 4.1: Results accuracy of SVM…….	66

Figure 4.2: Results accuracy curve of SVM……..	66
Figure 4.3: Adult data before classification……	67
Figure 4.4: Adult data after classification……..	67
Figure 4.5: Correlation matrix of adult data……..	68
Figure 4.6: Results accuracy of SVM with Gamma=1.5……	69
Figure 4.7: Accuracy curve of SVM with Gamma=1.5…….	70
Figure4.8: scatter plot before the classification water dataset……	70
Figure 4.9: scatter plot after the classification water quality dataset…….	71
Figure 4.10: K-mean parameters……	72
Figure4.11: Results accuracy of SVM with the k-mean (k=2) …….....	72
Figure 4.12: (a)Cover type dataset before classification. (b) after preprocessing…….	73
Figure 4.13: Cover type dataset after classification……	73
Figure 4.14: Classification by regression. Confusion Matrix of cover type dataset……...	74
Figure 4.15: Results accuracy of SVM and SVM with k-mean kernel (k=2) ……	75
Figure 4.16: Curve of SVM and SVM with k-mean kernel (k=2) …….....	76
Figure 4.17: Statistical analysis of the two classes. After classification of adult dataset…...	76
Figure 4.18: Statistical analysis of the two classes. After classification of Diabetes dataset.	76
Figure 4.19: Comparison of execution Time for SVM algorithms and k-means applied to SVM……	77
Figure 4.20: (a).The configration of the Hadoop in RapidMine……	78
Figure 4.20:(b).The configration of the Hadoop in RapidMiner……	78
Figure 4.21**:** Result accuracy of PSVM…….	80
Figure 4.22**:** Results comparison accuracy of SVM and PSVM……..	81
Figure 4.23: Accuracy curve comparison of SVM and PSVM…….	81
Figure 4.24: Execution time of SVM and PSVM on varying sizes of dataset…….	82
Figure 4.25: Comparison of SVM, SVM with k-mean, PSVM…….	84
Figure 4.26: Curve of SVM, SVM with k-mean, PSVM……..	84
Figure 4.27: Execution time of SVM and PSVM on varying sizes of dataset……..	85

Figure 4.28: Comparison of execution time for K-means applied to SVM and PSVM with data sizes…….	86
Figure 4.29:(a, b). Comparison curve of execution time for K-means applied to SVM and PSVM…….	86

LIST OF TABLES

Table3.1: Dataset Description……	44
Table3.2: Examples of Well-Known Kernel Functions…….	51
Table 4.1: Dataset description with dimensions and classes……..….	65
Table 4.2: The accuracy of four datasets with deferent kernel types……..….	65
Table 4.3: The accuracy of four datasets with deferent kernel types……..….	69
Table 4.4: Results accuracy of SVM with k-mean (k=2) …….	72
Table4.5: Comparison of result of both models…….	74
Table4.6: Compare Execution Time for SVM algorithms and k-means applied to SVM….	77
Table4.7: Hadoop cluster configuration with resources…….	79
Table 4.8: Hadoop Cluster Results…….	79
Table 4.9: Accuracy comparison of SVM and PSVM……	81
Table 4.10: Comparison of Execution Time for SVM and PSVM algorithms……..….	82
Table 4.11: Comparison of SVM, SVM with k-mean, and PSVM……	83
Table 4.12: Comparison of Execution Time for k-means applied to SVM and PSVM algorithms……	85

LIST OF ABBREVIATION

CGPGA	Coarse-Grained Parallel Genetic Algorithm.
DGPM	Directorate General of Preventive Medicine
DiP-SVM	Distribution Preserving of kernel Support Vector Machine.
FTSVM	Fuzzy Twin Support Vector Machine.
GaBP	Gaussian Belief Propagation.
GPGPU	General-Purpose computing with Graphics Processing Unit.
GPU	Graphics Processing Unit.
HDFS	Hadoop Distributed File System.
HPC	High-Performance Computer.
ICA	Independent Component Analysis.
LibSVM	Library for Support Vector Machines.
MPI	Message Passing Interface.
PCA	Principal Component Analysis.
PNBA	Parallel Naive Bayes Algorithm.
PSMR	parallel Support Vector Machine based on MapReduce.
PSVM	Parallel Support Vector Machine.
QP	Quadratic Programming.
RBF	Radial Basis Function.
RDD	Resilient Distributed Datasets.
SFS	Successive Feature Selection.
SVM	Support Vector Machine.
TSVM	Twin Support Vector Machine.
UCI	University of California Irvine Repository.

CHAPTER ONE

INTRODUCTION

1.1. Background

Recently the technology is growing, and the size of data is also increasing accordingly. The classification technique is used to solve the above challenges, which classify the big data according to the format of that data which should be processed. The type of analysis to be used, the processing methods at work, and the data sources for the data in which the target system is required to acquire, load, process, analyze, and store [35]. Problems that involve classification are considered to be instances of a branch of machine learning called supervised learning [36]. The machine is giving a training set of correctly classified instances of data in the first stage, and then the algorithm devised from this learning is used for the next step of prediction.

The rapid growth of data in many fields is a great challenge for traditional data management techniques for handling and processing such a significant volume of data. The extensive growth of data is determined, and there arises a quest for identifying an effective storage mechanism which can handle vast dynamic data. The improvements in technology have paved the way for a solution using cloud storage. In the current scenario, Cardio Vascular Disease is the primary cause of human mortality across the world. This analysis is the hardcore need in today's medical research for prediction of Cardio Vascular Disease [49].

Big Data is unorganized data that override the processing complexity of traditional database schemes. If the data is too big, it needs to move too fast, so it doesn't match the rule restricting the management of our database architectures. This information comes from multiple, distinct, independent

sources with complex and evolving relationships in Big Data, which is kept on growing day by day [34].

The support vector machine is a supervised classification which, new data is classified based on the training set. This training set data called input data, which consists of multiple attributes or features. Each row tagged with a class label. The correct results known target. And are given in input to the model during the learning process. The construction of a proper training validation and test set is crucial. These methods are usually fast and accurate. Traditional Classification approaches perform weak results when working directly because of the large amount of data, but SVM can avoid the problems of representing this many data. SVM is the most promising technique and approach as compared to other classification approaches [37].

SVM balances proper and accurate for large amounts of data, and compromise between classifier complexity and error controlled explicitly. Another benefit of SVMs is that one can design and use an SVM kernels for a specific problem that could be applied directly to the data without the need for a feature extraction process. It is particularly essential problems, where the feature extraction process loses a massive amount of structured data. SVM is the classification technique used to process extensive training data. The Big and complex data can leave the SVM since the result of SVM will be significantly influenced when there is too much noise in the datasets. SVM provides an optimized algorithm to solve the problem of overfitting. SVM is a valid classification model useful in handling those complex data [37].

The main challenges in Big Data are 1- data accessing 2- arithmetic computing procedures. 3-semantics 4- domain knowledge for different Big Data applications. The difficulties raised by Big Data are volumes, distributed data distribution, and by complex and dynamic characteristics.

As mentioned before, big data require new tools to mine information. Performance in big data may lead to more confident decision making, and

better decisions can result in greater operational efficiency, cost reduction, and reduced risk [29]. These huge datasets cannot be classified by single SVM. The critical issue with traditional SVM is its unreasonable algorithmic complexity, the excessive memory requirement of the required quadratic programming in big data. The limitation of traditional SVM is its speed and size in both training and testing phases. An efficient parallel algorithm and its implementation are key to work with big data. The implementation of big data raises new issues and challenges because of its nature and complexity [11, 12]. Although many machine learning approaches have been proposed to analyze various data sets sizes, in a supervised or unsupervised way, just a few of them have been properly adapted to handle large data sets. In the beginning, different Decision Tree Learning was used to analyze the big data [15, 16]. Several parallel data mining algorithms have been developed using threads, MPI, MapReduce [7, 8]. And several parallel SVM also has been designed to be suitable with a large amount of data set.

Support Vector Machine (SVM) was organized by Vladimir N. Vapnik in 1995[1] [2]. SVM are supervised learning algorithms that can be used for classification and regression. The primary goal of SVMs is to find the unique hyperplane having the maximum margin that can linearly separate the two classes (see figure1). When the training data are not linearly separable in the input space, SVMs can use kernel functions to project the training data to a feature space of a higher dimension, in which the linear separation becomes easier. [3]. SVM has been widely studied by many scholars and applied in many kinds of practical fields. But their computational and storage requirements increase rapidly with the number of training vectors, putting many problems of possible interest out of their reach [4]. Support vector machine learning aims to classify data sets where the number of training data is small and where traditional use of statistics of large numbers cannot guarantee an optimal solution.

A parallel SVM is based on the cascade SVM model, where the training is realized through partial SVMs. Each subSVM is used as a filter, and this makes it drove partial solutions towards the global optimum. The alternative techniques may not be directly relevant for finding the global solution. The parallel SVM model divide the large-scale data problems into independent, smaller optimizations problems. The support vectors of the subSVM in the first level are used as the input of a later level. The subSVM can be grouped into one final SVM hierarchically. The support vectors of two SVMs are combined with being input to the next SVM. The process running until only one set of vectors is left. A single SVM never deals with the whole training set, but with partitioned sets. If the filters in the first levels are efficient in extracting the support vectors, then the optimization will be most significant [17].

Dealing with the large size of real-life data can cause problems regarding computationally expensive tasks as follows:

1- Memory: The whole data set may not apply to the memory; the inefficient memory access slows down the training and testing phases.
2- Speedup: The matrix operations might take too long time to be performed due to the computationally expensive tasks.
3- Scalability: Algorithms may not scale to a large number of processors or a large number of samples.
4- Accuracy: Approximation methods for reducing the size of the problem may lead to poor classification accuracy.

Some efficient parallel approaches have been used to speed up the optimization addressed by SVMs and to handle further the issues mentioned regarding memory, speedup, scalability, and accuracy.[54]

1.2. Problem Statement

Many of research was working on parallel SVM, and some of them combine the parallel SVM with other techniques to get high accuracy and performance. However, many problems deserve high complexity to implement.

- Classification is one of the data mining methods that classify unstructured data into the structured class and groups, and it helps the user for knowledge discovery and plans the future works.
- Dealing with big data the complexity and the relationship between data is becomes more complicated. Many proposed traditional algorithms have limitation and weakness such as:
- Low performance in large data set.
- Poor run-time performance (delay time).
- High computation complexity.
- It is difficult to implement single SVM with big data this include analysis, capture, search, sharing, storage, transfer, visualization, security, querying, updating, and information privacy, data accessing, and arithmetic computing procedures. So, in this research the parallel SVM is used for classification.

1.3. Research Scope

High-Performance Computer (HPC) and parallel computing are the paradigms that induce a remarkable change in the way in which hardware and software are designed, as well as big data, are managed. Big data is a risky sample that is filled with many complexities. These complexities can have a great impact on the operation and implementation of big data. Thus, this big data analysis needs proper combined assessment strategies to solve this problem. An appropriate environment and assessment model can help data mining algorithms to enhance the performance of the learning models. On the other hand, the use of Hadoop is present in the current research.

1.4. Research Objectives

The main objective of this research is to study and implement a parallel support vector machine for big data classification model. In achieving this goal, the following specific research objectives were established:

- To study a Parallel Support Vector Machines (PSVMs) based on MapReduce for parallel big data classification.
- To implement the traditional Support Vector Machine (SVM) model for big data classification.
- To implement K- means clustering algorithm applied to Support Vector Machines (PSVMs) model for big data classification.
- To implement Parallel Support Vector Machines (PSVMs) model for big data classification.
- To builds three models using four datasets and analyze the results from both statistical and functional perspectives.
- To compression study of the results of the three models depending on the accuracy and computation time.

This thesis presented the results of real work in the application of SVMs, a k-means clustering algorithm applied to SVM, and a parallel support vector machine to solve the big data problems and implemented in four data sets. Properties of PSVM learning based on MapReduce are used to classify big data in a parallel manner in both the linear and non-linear dimensions.

1.5. Research Contributions

In this section the contribution was briefly discussed.

The first contribution was building single SVM model for classification of four different data set. The single SVM is not suitable to classify big data, and it could not achieve a high-performance classification or not.

The Second contribution was implementing SVM combined with K-mean clustering model for classification of four different data sets to give a high accuracy in classification task.

The third contribution was designing PSVM with Hadoop clustering to classify four different data sets and it is found that the PSVM with Hadoop (HPC center in Sudan) gives higher accuracy than single SVM and SVM combined with the K-mean models.

The fourth contribution is the use real data set of water quality (from Mistry of health Sudan) used for the implementation of the models.

1.6. Research Organization

The rest of the chapters are organized as follows:

Chapter 2 begins by defining the methods and techniques used in the thesis. Then it provides a review of the existing literature concerning models based on the support vector machine algorithm (SVM) and parallel support vector machine for big data classification in this thesis. Further, it reviews the existing classification models used for big data.

Chapter 3 describes the methodology of the research. This methodology is divided into data collection, data Pre-processing, data mining techniques for classification, and their real implementation on the four data sets.

Chapter 4 represents experimental results using classification techniques with different data sets, and it is analyzed.

Chapter 5 presented the conclusion of this research.

CHAPTER TWO

LITERATURE REVIEW

Data mining is a significant research area for large-scale data. We are currently in the world of big data, in which big data technology is being rapidly applied to many and deferent fields, many techniques are emerging in the field of Big Data. The Hadoop file system is one of them. In most of the

references, the most effective classifier considered is the support vector machine. The SVM classification model depends on the number of support vectors generated by the classifier. The number of support vectors is directly proportional to the required memory to store the support vectors. The most commonly used sequential SVM is challenging to work with large scale data set. So, if the data is growing, we need to use a particular environment to work with it like Hadoop or spark.

The primary definition of big data is a term for data sets that are large and complex that conventional data processing applications are inappropriate. Challenges include analysis, capture, search, sharing, storage, transfer, visualization, security, querying, updating, and information privacy [9, 14]. Researchers have developed many algorithms and methods to deal with big data.

It is challenging to measure the size of structured and unstructured data. The analysis of big data is required machine-based systems and technologies. Effective implementation techniques are the key to meeting the scalability and performance requirements entailed in such scientific data analysis [31]. This chapter is divided to three sections, section one presents the definition of basic concepts, section two describes the related work, at last in section three chapter is concluded.

Support Vector Machine (SVM) was organized by Vladimir N. Vapnik in 1995[1] [2]. SVM are supervised learning algorithms that can be used for classification and regression. The primary goal of SVMs is to find the unique hyperplane having the maximum margin that can linearly separate the two classes (see figure1). When the training data are not linearly separable in the input space, SVMs can use kernel functions to project the training data to a feature space of a higher dimension, in which the linear separation becomes easier. [3]. SVM has been widely studied by many scholars and applied in many kinds of practical fields. But their computational and storage

requirements increase rapidly with the number of training vectors, putting many problems of possible interest out of their reach [4]. Support vector machine learning aims to classify data sets where the number of training data is small and where traditional use of statistics of large numbers cannot guarantee an optimal solution.

2.1. Data Mining

Data mining is one of the most important scientific topics all over the world, which is useful in most scientific fields. It is a valuable technique for extracting knowledge from a mass of stored raw data. By using various models in data mining, human errors are significantly reduced [56]. Data mining, or knowledge discovery in databases, has been popularly recognized as an important research issue with broad applications specially in big data analysis. Data mining aims to discover hidden knowledge, unknown patterns, and new rules from large databases that are potentially useful and ultimately understandable for making crucial decisions. It applies data analysis and knowledge discovery techniques under acceptable computational efficiency limitations and produces a particular enumeration of patterns over the data [56].

Data mining algorithms falls under 4 classes

Association rule learning: This category of algorithms search for relation between variables. This is used for application like knowing the frequently visited items. The popular algorithms are (a priori partition, FP- Growth ECLAT).[77]

Clustering: This category of algorithms discovers groups and structures in the data such that objects within the same group i.e. cluster are more like each other than to those in other groups. The popular algorithms of clustering are (K-Means, Expectation, Maximization, DBSCAN, and Fuzzy C Means)

Classification: This category of algorithms deals with associating an unknown structure to a well-known structure. The popular algorithms for the classification are (Decision Tree – C4.5, KNN, Naïve Bayes, and Support Vector Machines)

Regression: This category of algorithms attempts to find a function to model the data with least error. The popular algorithms for the regression are (Multivariate linear regression). [77]

2.2. Machine Learning

The Machine Learning field evolved from the field of Artificial Intelligence, which aims to stimulate the intellectual abilities of humans by machines [57]. Machine learning is a scientific method concerned with the design and development of algorithms that are taken as an input of empirical data, such as from sensors or databases. The main two parts of machine learning are supervised learning and unsupervised learning. Vapnik and *et al*. (2013) [5] had shown that machine-learning focuses on the design of algorithms that recognize intricate patterns and make predictions and intelligent decisions that depend on input data. An essential task in Machine Learning is a classification [31]. Machine learning is requisite to meet the challenges posed by big data and uncover hidden patterns, knowledge, and insights from big data to turn its potential into real value for business decision making and scientific exploration [58].

2.3. Classification techniques

The classification method analyzes the big data according to it is organization. There are many challenges of big data like the load and store, the form of analysis, the processing techniques [8]. With supervised learning, there is a problem that involves classification, and they are regarded to be an instance of machine learning [59]. In the first step, machine learning is given a training set of rightly classified examples of datasets. Then, the algorithm is designed

using this learning for prediction. Universal Classification techniques are weak when working immediately with a huge volume of data, but PSVM can express this big data. SVM is the standard technique used for classification methods. The use of SVM kernels for a particular problem could be applied directly, and this is the most advantage of SVM, so no need for using the feature extraction process. Because of the loss of data by the feature extraction process in huge data, the use of kernel is a critical problem. The SVM results will be significantly affected when there is too much noise in the datasets. SVM is an efficient and reliable classification routine used to manage complex data [60].

2.3.1.Big data

Big data is unstructured data that are complex to be processed in the original database systems. Because of fast-growing data, big data doesn't deal with the rule restricting the behavior of the database architectures. This data comes from multiple different sources with complications. The developing relationship in big data is growing every day. The main challenges to big data are data locating, computing functions, and the environment where algorithms are applied, and the recourses are to be used. Big data creates various challenges for traditional Machine Learning algorithms in terms of scalability, adaptability, and usability, and presents new opportunities for inspiring transformative and novel ML solutions to address many associated technical challenges and create real-world impacts [61].

Big Data is the datasets that cannot manage by the traditional data mining techniques and software tools available. Big Data seems like a huge size dataset that covers any information in its massive volume, which cannot be explored without using new algorithms or proper data mining techniques [61]. The big data architecture is divided into three tiers, as presented in Figure 2.1.

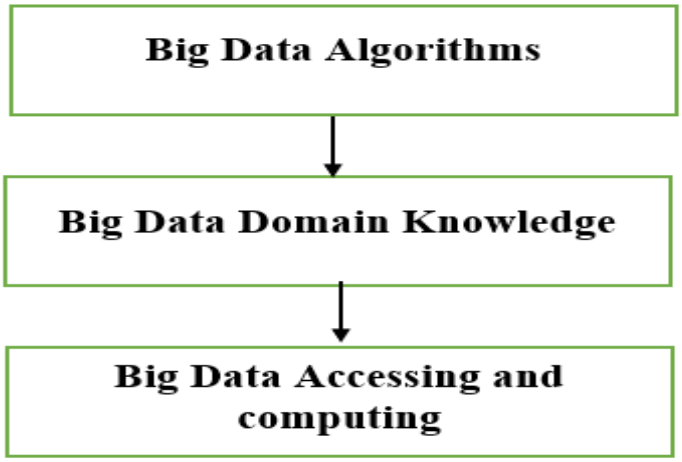


Figure2.1: Big Data Architecture.

There deferent common types and description of big data are described as follows:

2.3.1.1. Machine-generated data

Service companies have rolled out some meters to measure the consumption of water, gas, and electricity at orderly intervals of one hour or less. These intelligent meters generate large volumes of interval data that needs to be analyzed. Utilities also run big, expensive, and complicated systems to generate power. Each grid is included sophisticated sensors that monitor voltage, current, frequency, and other critical operating characteristics [61].

2.3.1.2. Web and social data (Transaction data)

Telecommunications operators need to develop customer churn models that include social media and transaction data to follow up with the competition. The churns value depends on the customer attributes such as customer master data such as date of birth, gender, location, and income, and the social behavior of customers. Telecommunications providers who implement a predictive analytics strategy can manage and predict churn by analyzing the calling patterns of subscribers [61].

2.3.1.3. Web and social data

When a new product is released is launched, the marketing departments use Twitter serves to manage analysis to discover what users are saying about the latest products or services. Customer sentiment must be combined with customer profile data to obtain meaningful results [61].

2.3.1.4. Human-generated Data

IT departments are analyzing application logs by turning to big data solutions to analyze application logs to get insight that can increase system performance. Log files from various application vendors are in different formats; they must be standardized before IT departments can use them [61].

2.3.1.5. Web and social data (Biometrics)

Facial recognition technology is used to combine the photo from social media to make personalized offers to customers based on buying behavior and location. Could this capability have a significant impact on retailers? Retailers would need to make proper privacy disclosures before implementing these applications [61].

2.3.1.6. Machine-generated data (Transaction data)

Retailers can target customers with specific promotions and coupons-based location data. Solutions are typically designed to detect a user's location upon entry to a store or through GPS. Location data combined with customer preference data from social networks enable retailers to target online and in-store marketing campaigns based on buying history. Notifications are delivered through mobile applications, SMS, and email [61].

2.3.2.Support Vector Machine (SVM)

Support Vector Machines (SVMs) are powerful classification and regression tools. They have been widely studied by many scholars and applied in many kinds of practical fields. But their compute and storage requirements increase rapidly with the number of training vectors, putting many problems of

practical interest out of their reach. The SVM implementations are shown in Figure2.2 [4].

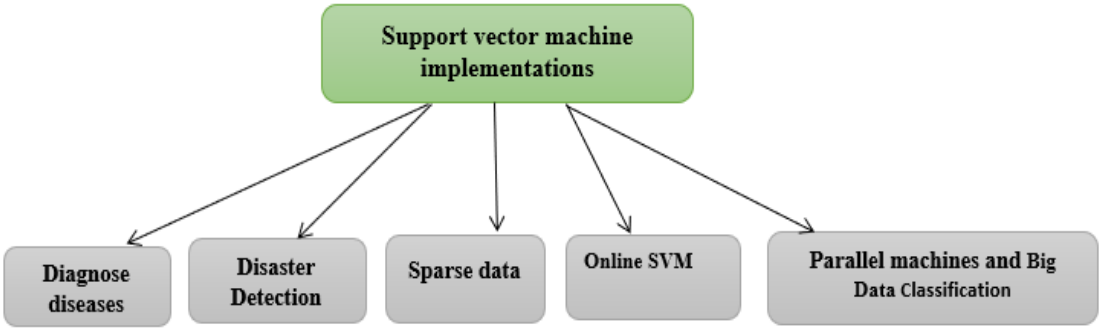


Figure2.2: Support vector machine implementations.

Support Vector Machine (SVM) was proposed by Vladimir N. Vapnik in 1995[34, 35]. SVM are supervised learning algorithms that can be used for classification and regression. The primary goal of SVMs is to find the unique hyperplane having the maximum margin that can linearly separate data classes, this is shown in Figure2.3. When the training data are not linearly separable in the input space, SVMs can use kernel functions to project the training data to a feature space of a higher dimension, in which the linear separation becomes easier [3]. SVM has been widely studied by many scholars and applied in many kinds of practical fields. Their computational and storage requirements increase rapidly with the number of training vectors, posing many problems of practical interest out of their reach [22]. Support vector machine learning aims to classify data sets where the number of training data is small and where traditional use of statistics of large numbers cannot guarantee an optimal solution. Figure2.3 considered the two decision boundaries on the same data which is small margin and large margin. The SVM classifier maximizes the margin between two classes [5]. The two decision boundaries on the same data set [36].

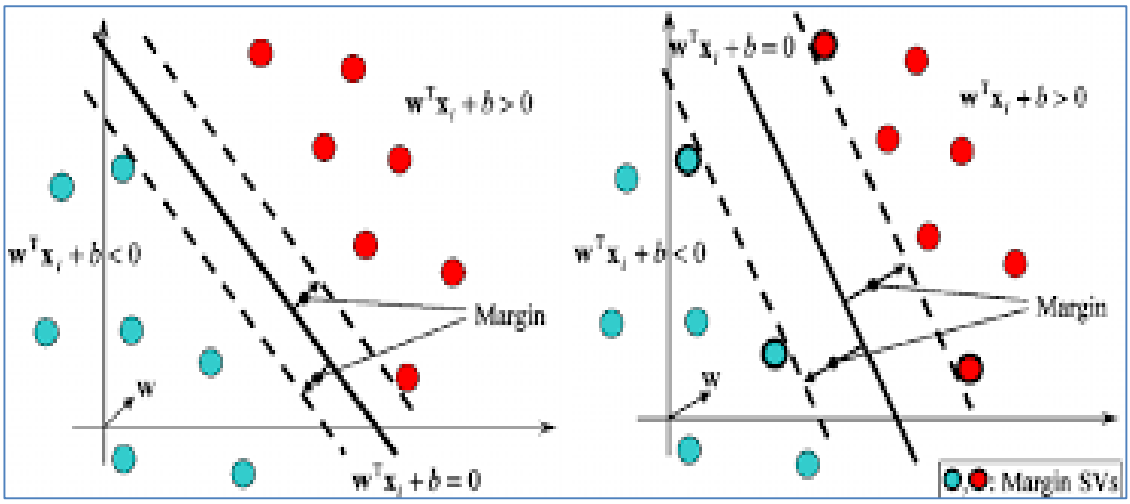


Figure2.3: SVM maximizes the margin between two classes. Left. Small margin. Right. Large margin [14].

Support Vector Machines are well known for their strong theoretical foundations, generalization performance, and ability to handle high-dimensional data. In binary classification, if $((x_i, y_i) \ldots (x_n, y_n)$ are the training data set, x_i are the vectors constitute the instances, and $y_i \in \{-1, +1\}$ are the labels of those instances. An optimum hyperplane was built by SVM, which linearly discriminates in a higher dimensional feature space that chooses the largest margin separation between the two classes. The SVM classifier is shown in Figure 2.4. The SVM classifier is also used for Multi-class classification which is shown in Figure2.5. The solution of SVM obtained by minimizing the primal objective function, and this is shown in equation (1) [37].

$$\min_{w,b} \; j(w,b) = \frac{1}{2} \square w \square^2 + c \sum_{i=1}^{n} \xi_i$$

$$with \forall_i \left\{ \begin{array}{l} y_i (w.\Phi(x_i) = b \geq 1 - \xi_i) \\ \xi_i \geq 0, \end{array} \right\} \qquad (1)$$

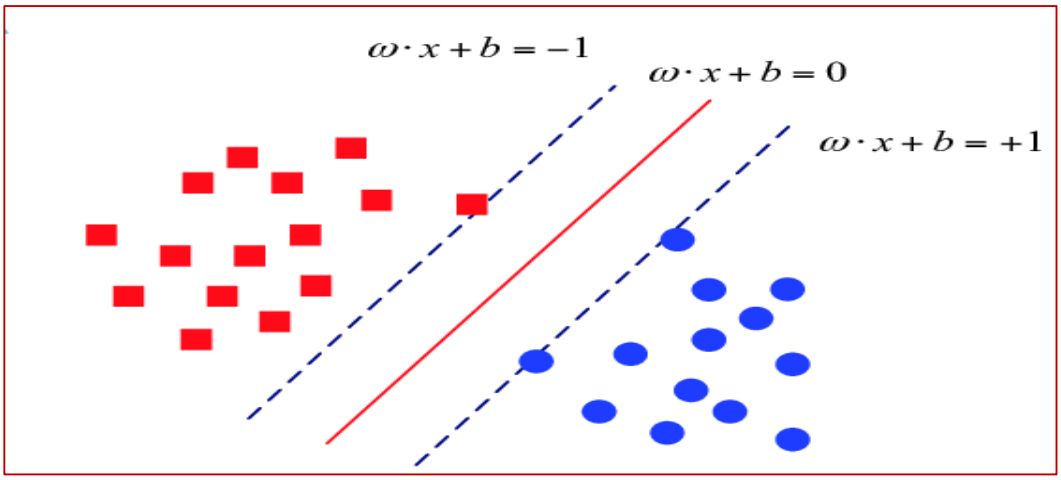


Figure2.4: Support vector machine classifier [4]

In equation (1), w is the coefficient vector of the hyperplane, b is the offset, y_i are the labels. Φ (x_i) is the mapping from input space to feature space, and ξi are the slack variables that permit the non-separable case by allowing misclassification of training instances. The convex quadratic programming (QP) problem in equation (2) is solved by optimizing the dual cost function:

$$\max_{\alpha} G(\alpha) = \sum_{i=1}^{N} \alpha_i y_i - \frac{1}{2} \sum_{i.j} \alpha_i \alpha_j K(x_i . x_j).$$

$$subjectto \begin{cases} \sum_i \alpha_i \\ A_i \le \alpha_i \le B_i \\ A_i = \min(0, cy_i) \\ B_i = \max(0, cy_i) \end{cases} \qquad (2)$$

Where K (x_i; x_j) = (∅ (x_i) ∅ (x_j)) is the kernel matrix representing the dot products ∅ (x_i). ∅ (x_j) in feature space.

The description of general SVM can be as follows. Let l training samples be T= {(x_1, y_1), ⋯, (x_l, y_l)}, where $x_i \in R_n$, $y_i \in$ {1, −1} (classification) or $y_i \in R$ (regression), i=1, ⋯, l. The Nonlinear mapping function is ∅ (x_i) entailing a kernel K (x_i, x_j) = ∅ (x_i) ∅ (x_j). The implementation of SVM classification is solved by equation (3) [4].

$$\min_{w\xi_i b} \left\{ \frac{1}{2} \square w \square^2 + c \sum_i \xi_i \right. \qquad (3)$$

$$s.t.y^{i}(\Phi(x_i)w+) \geq 1-\xi_i \forall_i = 1,\ldots,n \qquad (4)$$

The classification precision of the SVM model can be calculated as

$$Accuracy = \frac{correctly\ predicted\ data}{Total\ testing\ data} * 100\% \qquad (5)$$

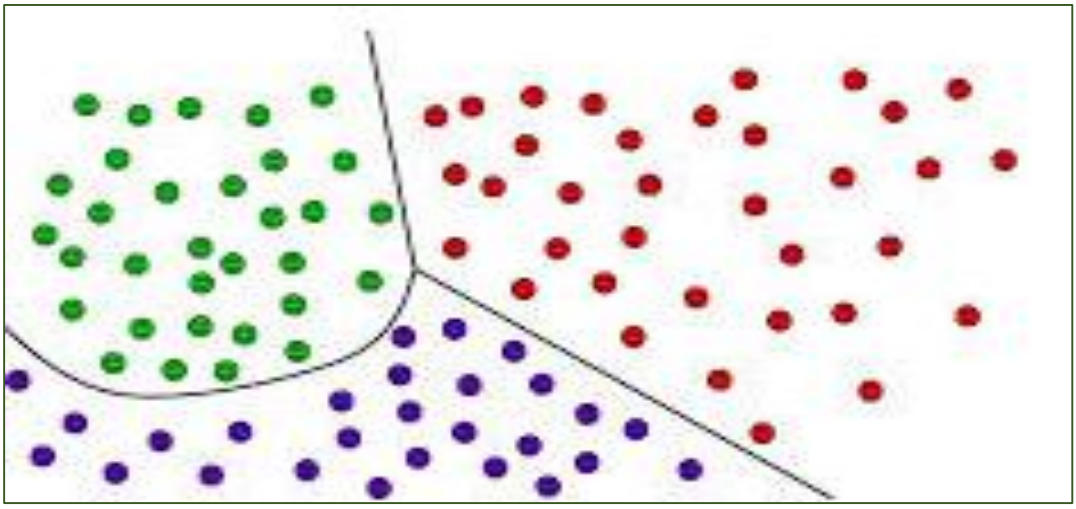


Figure2.5: Multi-class classification [4].

2.3.3.K-means clustering algorithm

Clustering is divided the data into groups. Each group is established by similar data; it means that the similarity between dates in the same group is smaller than others [54, 55]. K-means is a centroid-based algorithm which takes the input parameter, usually named k. Then partition s a set of n objects into k clusters leading to high intra-cluster similarity and low inter-cluster similarity. The mean values of the objects in a cluster is the way to determine cluster similarity. It can be viewed as the cluster`s centroid. The k-means algorithm initially selects k objects, each of which primarily shows a cluster mean or Centre. The remaining objects are assigned to cluster with most similarity depending on the distance between the object and cluster mean. Then it computes the new mean iterating until the centroid function converges.

The k-mean is used for clustering the data set, the k-mean clustering algorithm is used to cluster the original training data points into k clusters this is shown in Figure2.6. K-means clustering is an unsupervised algorithm working based on the similarity. It is an iterative algorithm frequently used in the field of data mining, and quite efficient in partitioning the data points. Note that the result of SVM is substantially dependent on the value of k. the k-means uses kernels to assessment distances between examples and clusters. The k-mean algorithm is quadratic in number of Examples and does not return a Centroid Cluster Model. Thus, it is necessary to sum all examples of a cluster to calculate one distance. After clustering, some clusters may contain the data of two class labels (called duo-cluster) [54, 55].

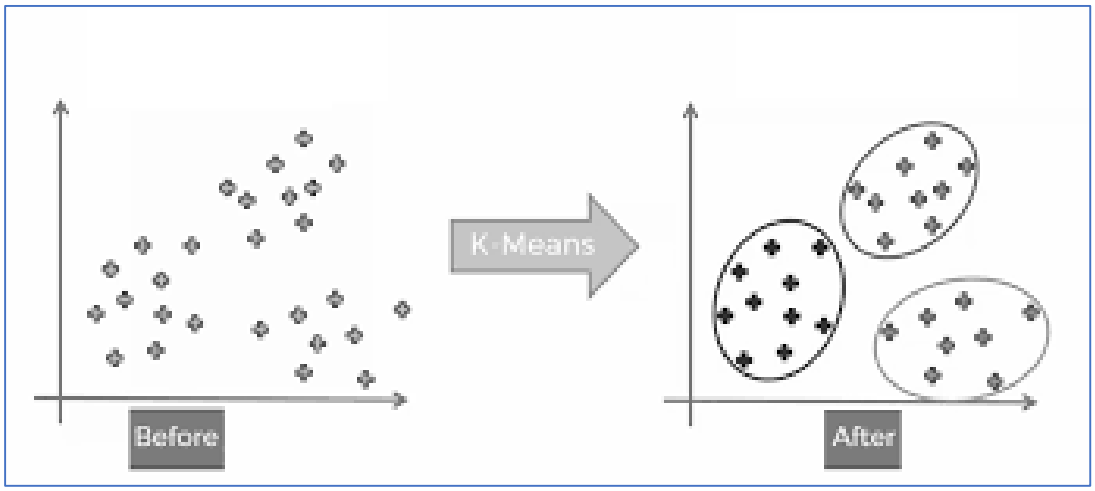


Figure2.6: K-means clustering. [54, 55].

2.3.4.Parallel Support Vector Machine (PSVM)

The parallel computing of SVMs is becoming a necessity for improving the performance of SVMs for big data and already has demonstrated promising results for improving large-scale problems [38].

A parallel SVM is based on the cascade SVM model, where the training is realized through partial SVMs. Each subSVM is used as a filter, and this

makes it straightforward to drive partial solutions towards the global optimum. Support vectors form subSVM are used as the input of later subSVMs. Large scale data optimization problems can be divided into independent, smaller optimizations problems by using the PSVM model. The output support vectors from the first subSVM are used as input to later subSVMs. All the subSVM can be combined into one final SVM hierarchically [4]. The parallel SVM training process can be described as in Figure 2.7.

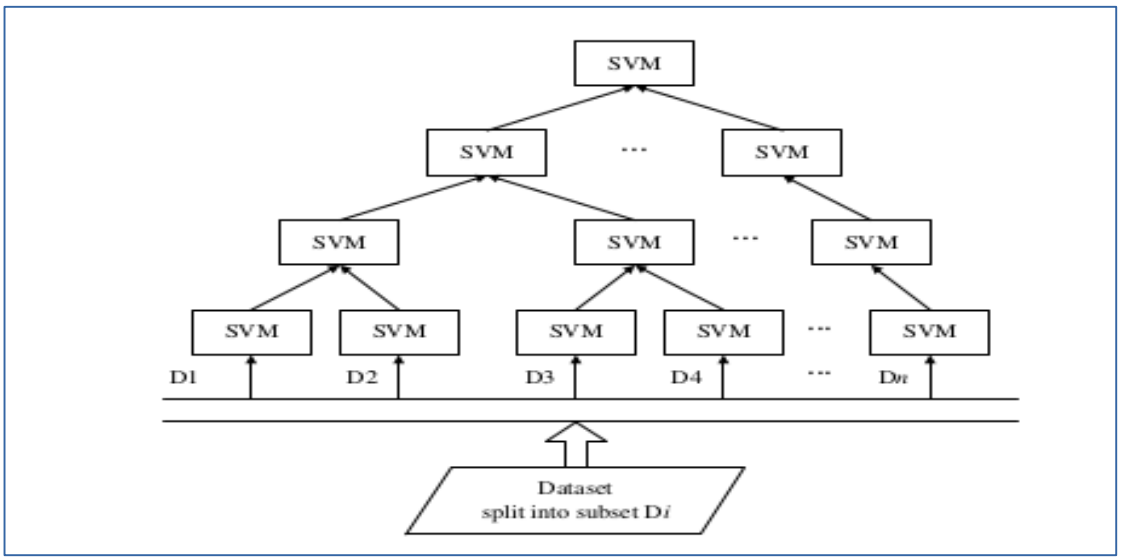


Figure2.7: Training flow of Parallel SVM [5].

This Cascade SVM algorithm considers the possibility of multiple runs through the cascade for each data set. After finishing runs through the cascade, the subsets for the first step of the next term created by combining the remaining SVs of the final model with each subset from the first level of the first run. From the architecture of parallel SVM, we can find that it is a hierarchical structure. The low-level SVM training has to perform when all the upper-level sub SVM is trained. In the last level of the architecture, all the support vectors should be included in the training samples. The sample size must be more significant than the number of support vectors. When the ratio

between the support vectors and training sample is bigger, the speedup will be less. It is the limitation of the cascade SVM model [5].

2.3.5.Hadoop Framework

Hadoop framework is open-source software that encourages distributed application. It allows user applications to communicate and work with several independent computer nodes and terabytes or even petabytes of data. Essential characteristics of the Hadoop framework are partitioning the data into thousands of machines and executed in a parallel manner. The Hadoop cluster can be set up by simply using commodity hardware. These commodity servers can process large data efficiently [3]. The Hadoop framework works with two main components. These two main components are Hadoop Distributed File System (HDFS) and MapReduce distributed programming model [3]. The architecture of Hadoop framework is shown in Figure 2.8.

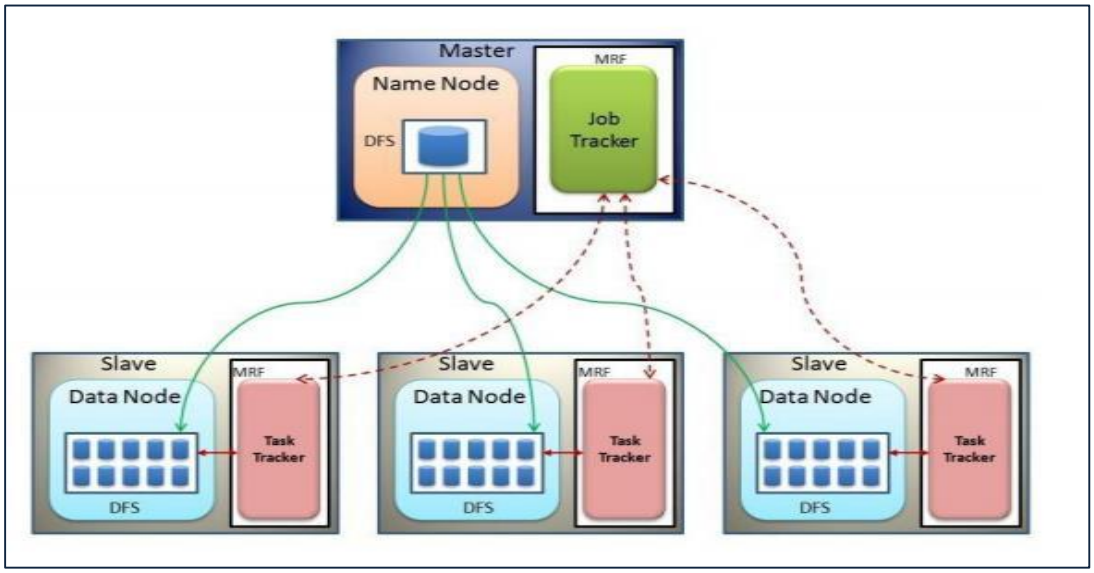


Figure 2.8: Architecture of Hadoop Cluster [31].

HDFS stores big data on a dedicated server called the NameNode. Application data are stored on other servers named DataNodes. By default, HDFS stores

three separate copies of each data block to ensure reliability, availability, and performance [31].

The HDFS name node runs the NameNode daemon. The job submission node runs the JobTracker, which is the single point of contact for a client wishing to execute a MapReduce job. The JobTracker monitors the progress of running MapReduce jobs and is responsible for coordinating the execution of the mappers and reducers. Typically, these services run on two separate machines, although in smaller clusters, they are often co-located. The bulk of a Hadoop cluster consists of slave nodes (only three of which are shown in the Figure 2.9) that run both a TaskTracker, which is responsible for actually running user code, and a DataNode daemon, for serving HDFS data. HDFS is written in java language and is a portable filesystem of Hadoop. HDFS stores all its metadata to its devoted server known as NameNode, also called master node NameNode is the first node through which the user communicates to perform any input and output to the Hadoop cluster [3]. The HDFS Architecture is shown in Figure2.9.

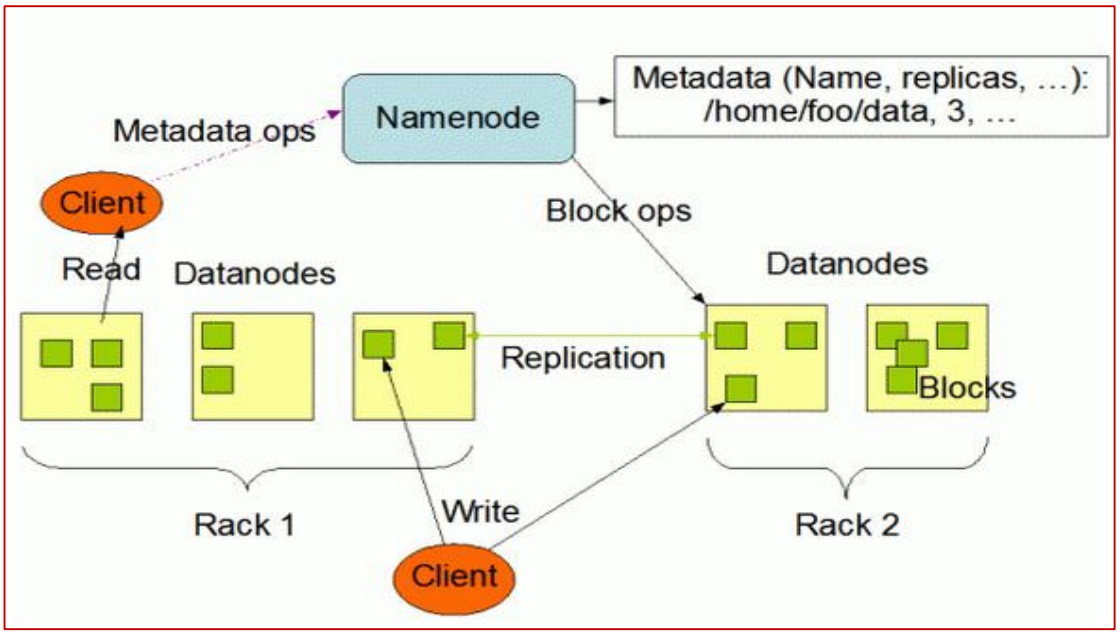


Figure2.9: HDFS Architecture [3].

MapReduce is a programming model derived from the Map and Reduces function that is combined from functional programming. MapReduce is used widely to run parallel applications for large scale datasets processing. It used key/value pair data type in the Map and Reduce functions. The overview of the MapReduce system is shown in Figure 2.10.

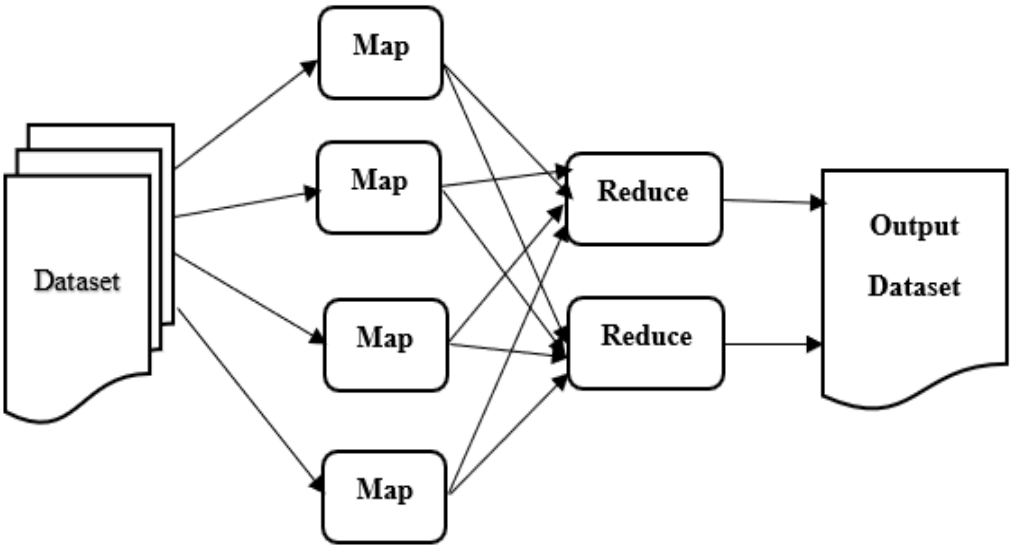


Figure 2.10: Overview of MapReduce system [3].

In the PSVM Algorithm. Training Dataset is having Instances, Attributes, and the user provides Class-Labels. In Map step, map tasks process an associated data chunk in its space. The output of each map process is the localized SVM weight vector (w j). In reduce step, reduce is computed the global weight vector (W global) by summing the individual maps' weight vectors. The output will be the results with a Model having Global W and SV (support vectors).

A map-reduce job usually splits the input dataset into independent chunks. The map tasks process the spilled jobs in a parallel manner. The framework

sorts the outputs of the maps, which are then inputted to the reduce tasks. Structure and Flow of PSVM Algorithm using MapReduce in Figure 2.11.

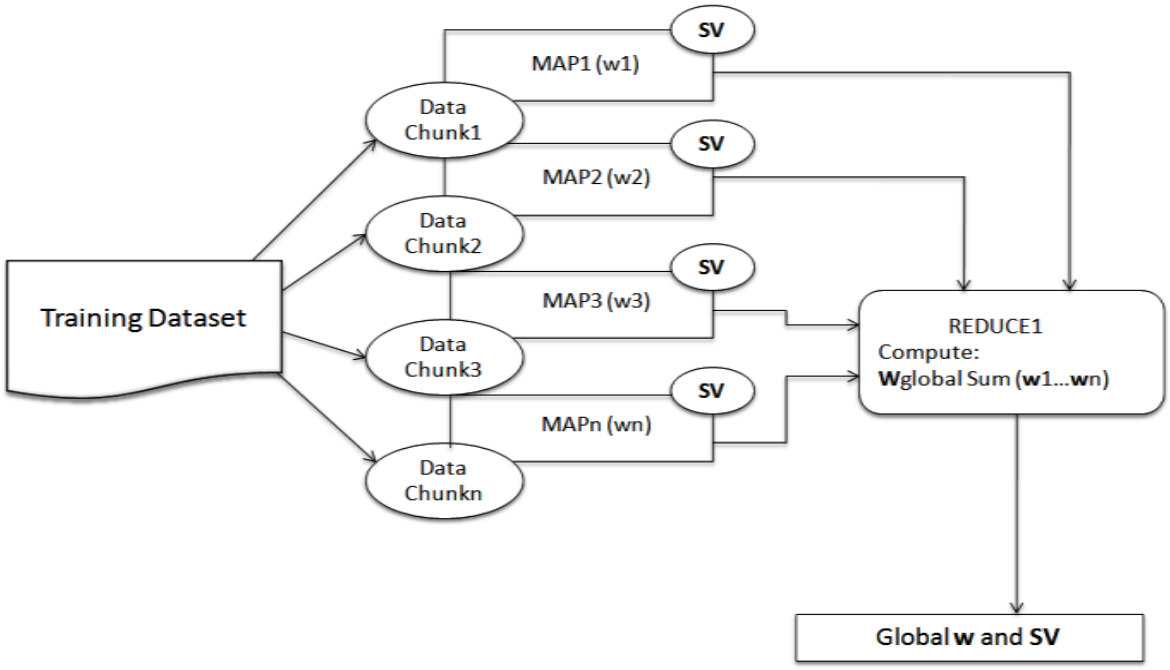


Figure 2.11: Structure and Flow of PSVM Algorithm using MapReduce [3].

A MapReduce based SVM for large scale data, which implemented on the Hadoop framework, is proposed in [3], where the impact of penalty and kernel parameters on the performance of parallel SVM is analyzed. In [17], the parallel SVM based on iterative MapReduce model Twister is analyzed, training samples are divided into subsample, and each subsample is trained with an SVM model. LibSVM is used to train each subSVM. The support vectors of each subSVM are taken as the input of the next layer subSVM. The global SVM model will be obtained through iteration. The advantage of this method is reducing the computation time and being efficient in data-intensive problems; the partition number can be estimated according to the concrete problems. Reference [10, 11and 13] deal with the big data and its content,

scope, samples, methods, advantages, and challenges, and discusses privacy. Challenges associated with network intrusion prediction are dealt with in [12].

2.3.6.Integration of Rapid-Miner and Hadoop

RapidMiner is a data science platform. It provides a combined environment for data preprocessing, machine learning analysis, deep learning, text mining, and predictive measurement. It is used for many applications such as business, commercial applications, research, education, training, rapid prototyping, and application development. RapidMiner supports the steps of the machine learning analysis, including data preparation, results in visualization, model validation, and optimization. [20].

The Hadoop integration in RapidMiner is done by an extension named Radoop. This extension communicates with the Hadoop cluster to run the jobs and provides additional operators for RapidMiner. Data analytics functions of Hive and Mahout are decided to be used because they are highly optimized. The overall architecture can be seen in Figure 2.12.

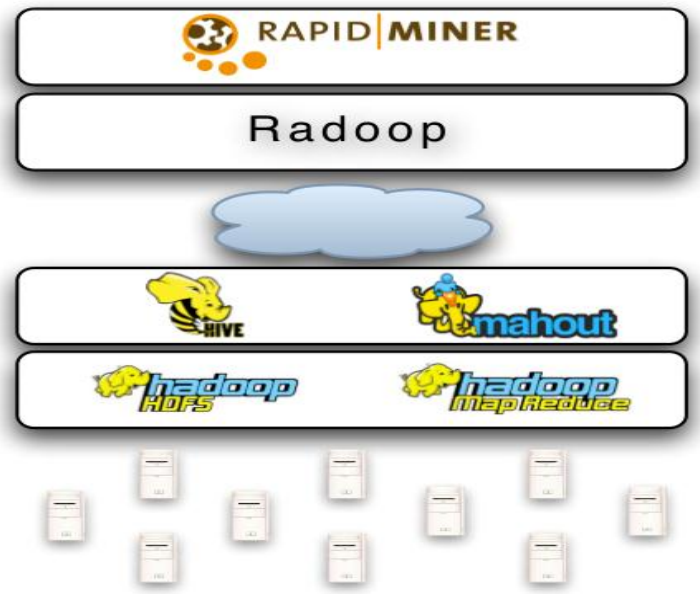


Figure 2.12: The Architecture of the Rapid-Miner - Hadoop integration [20].

The integration of Hadoop, HDFS, Hive, and Mahout Functionalities in the RapidMiner environment are complicated distributed processes. It cannot be used with RapidMiner's simple operator flow interface. The nature of a Radoop process begins with adding the RadoopNest operator. It contains all the general settings for the cluster, like the IP address of the Hadoop master node, and all other Radoop operators can only be used inside this operator [20].

Radoop powered by RapidMiner is client software that connects to a Hadoop cluster and executes processes created with an intuitive graphical user interface. Radoop Nest is an important building block that contains at least one Radoop Nest operator. It specifies the connection to the Hadoop cluster. The subprocess you put inside the Radoop Nest describes the process that runs on this Hadoop cluster. All other parts of the process outside the Nest process the data in the memory. In Radoop, the data store in tables in Hive, the Hadoop ExampleSet object used to describe it. It is essential to note that the Hadoop ExampleSet only stores several pointers and settings. Still, all data is stored in Hive on the distributed file system, so there is no significant memory consumption during Radoop processes.

2.4. Related Work

2.4.1.MapReduce

He, Qing, Zhuang et al. [23] had proposed implementation methods of several classifications' algorithms based on MapReduce; these methods can process large data sets, where the large task is partitioned into small pieces which can be executed simultaneously by the CPUs in the cluster. Bickson and Dolev [19] proposed a parallel implementation of an SVM solver using Message Passing Interface (MPI). They introduce a distributed SVM solver based on

the Gaussian Belief Propagation (GaBP) algorithm. GaBP is a message-passing algorithm for performing inference on graphical models (trees), representing a particular status of continuous BP where the fundamental distribution is Gaussian. They improve the original version by reducing the communication load, represented by the number of messages sent in each optimization iteration, from O (n^2) to O (n) aggregated messages, where n is the number of data points. Previously, it was shown that the GaBP algorithm is very efficient for sparse matrices. The algorithm exhibits excellent performance for dense matrices as well and can be used with kernels.

Shim, Kyuseok (2017) [21] had introduced a MapReduce framework based on Hadoop and discusses how to design practical MapReduce algorithms and present the state-of-the-art in MapReduce algorithms for data mining, machine learning and similarity joins. MapReduce is a programming model that allows the development of scalable parallel applications to process big data on large clusters of commodity machines [21]. The MapReduce framework executes the primary function on a single master machine where we may preprocess the input data before map functions are called or post-process the output of reduce functions. Depending on the applications, a pair of maps and reduce tasks may be executed once or multiple times [21].

In [25] A MapReduce based distributed parallel SVM training algorithm for binary classification problems is presented. This work shows how to distribute an optimization problem over cloud computing systems with the MapReduce technique. At each node, a subset of the training dataset is used for training to find out a binary classifier function. The algorithm collects support vectors (SVs) from every node in the cloud computing system and then merges all SVs to save as global SVs [25].

A parallel SVM based on MapReduce (PSMR) algorithm for email classification is proposed [7]. The performance of the algorithm proposed is better than Naive Bayes (NB) classifier and one-by-one SVM classifier. First,

SVM was used to classify the folder of each email based on a particular field of data from the email. Then, SVM was used on each email treated as a single bag-of-words. Naive Bayes is a simple algorithm for constructing classifiers which attribute the instances the class with the highest posterior probability. A limitation of Naive Bayes is the assumption of independent predictors, a difficult condition to be filled up in real life [30].

Kiran, M. et al [31]. Had analyzed Sequential Support Vector Machine in WEKA and various MapReduce Programs, including Parallel Support Vector Machine on the Hadoop cluster. In this way, algorithms are verified and validated on a Hadoop Cluster using the concept of MapReduce. The performance of the above applications has been shown for execution time, training time, and the number of nodes. Experimental results show that as the number of nodes increases, the execution time decreases [31].

Kiran, M. et al [31]. Had analyzed Sequential Support Vector Machine in WEKA and various MapReduce Programs, including Parallel Support Vector Machine on the Hadoop cluster. In this way, algorithms are verified and validated on a Hadoop Cluster using the concept of MapReduce. The performance of the above applications has been shown for execution time, training time, and the number of nodes. Experimental results show that as the number of nodes increases, the execution time decreases [31].

Cloud computing, which is emerging as a new computational paradigm shift, is Proposed. Hadoop-MapReduce has become a powerful computation model for processing large data on distributed commodity hardware clusters, such as clouds. In all Hadoop implementations, the default FIFO scheduler is available where jobs are scheduled in FIFO order with support for other priority-based schedulers also. Various scheduler improvements possible with Hadoop and also provided some guidelines on how to improve the scheduling in Hadoop in Cloud Environment [32]

Konstantin and et al [33]. Had studied a Hadoop MapReduce has been observed in all Standalone, Pseudo-distributed, and Fully Pseudo-distributed mode. This Hadoop cluster contains four nodes, one Master (Name- Node) and three Slaves (Data- Node). Scaling up the Hadoop Cluster- having Client and Secondary Name- Node will be studied [33].

Phu, Ngoc et al., 2017[44] had proposed a new model using an SVM algorithm with Hadoop MapReduce for English document level emotional classification in the Cloudera parallel network environment. The new model is tested on the English testing data set, and it achieves 63.7% accuracy of sentiment classification on this English testing data set. This model can be applied to many other languages, although these data sets are small. However, the new model can be applied to the big data set with millions of English documents in the shortest time. The study of the model shows that the average time of the semantic classification of the SVM algorithm in the sequential environment is higher than the average time of the emotion classification of the SVM in the Cloudera parallel network environment [44].

The significant challenges are observed in [23] are that complex machine learning algorithms, such as Neural Networks, are complicated to implement in the MapReduce paradigm. The mahout had a proposal to implement the Neural Network with backpropagation learning on Hadoop but had never achieved so far [29]. In theory, in many iterative machine learning algorithms, each iteration needs to pass over the whole set of data at least once. If the large-scale data is stored on hard drives or distributed in the cluster, each pass is costly due to the cost of communication between memory and secondary storage. Also, generally, iterations cannot be executed in parallel because, by design, the steps in an iterative algorithm are in the serial format [30]. MapReduce works well when the parallel problem has no dependency or communication between parallel tasks.

2.4.2.parallel implementation

Zanghirati and Zanni [20] had introduced a parallel implementation of SVM solver using MPI based on a decomposition technique that splits the problem into smaller quadratic programming subproblems. These subproblems are solved by a variable projection method, which has proven to be quite valuable in the solution of nonlinear least-squares problems in which a substantial number of the parameters are linear. Its benefits are efficiency and, more importantly, a better likelihood of finding a global minimizer rather than a local one [31]. This is well suited to a parallel implementation and is very useful in the case of Gaussian support vector machines. The (VPM) method with a special updating rule for its projection parameter has been appropriately studied for the QP problems. The outcomes of each subproblem are combined. The parallel solution can be used in peer-to-peer and grid environments, where there is no central authority that allocates the work. However, an implementation using the asynchronous communication model is used in that paper due to a lack of support for asynchronous communication, which could affect the speed of training time.

SVM training is a computationally intensive process. Several SVM formulations, solvers, and architectures for improving SVM performance have been proposed, including distributed and parallel computing techniques [21]. A parallel SVM training algorithm using Graphics Processing Unit (GPU) was introduced in [22], in which training multiple SVMs were performed using subsets of the training data. Next, the classifiers are combined into a final single classifier. The training data is then reallocated to the classifiers based on their performance, where the performance gain of multi-threading highly depends upon the hardware specification, and the process is iterated until convergence is reached. The aim is letting the multiple training tasks be aware of each other and share the kernel matrix cached in memory. The novelty of this method enables every job to synchronize together at each iteration of the training phase, and if some of these tasks share

the same support vectors, there is no need to do duplicated kernel computations, but fetch the kernel results from memory.

Collobert et al., [26] had proposed a Parallel Mixture of SVMs for very large-scale problems. They use a mixture of several SVMs, each of them trained only on the part of the dataset. The method is much faster than training only one SVM. This mixture can be simply parallelized, which could improve again significantly the training time, and each expert can be trained separately, and the algorithm is able to combine. For a more realistic problem, they carried out a series of experiments on the part of the UCI Forest Cover Type dataset. They modified the seven classes classification problem into a binary classification problem where the goal was to separate class two from the other six classes. A series of experiments were done in order to see the influence of the hidden number units of a simple linear function named gater. There is a definite performance improvement when the number of hidden units is increased, while the enhancement with additional experts occurs but is not so strong. However, the training time increases also rapidly with the number of hidden units while it slightly decreases with the number of experts if one uses one computer per expert.

Rebentrost, Patrick et al. (2014) [22] had implemented a support vector machine, an optimized binary classifier on a quantum computer, with a complexity logarithmic in the size of the vectors and the number of training examples. A quantum support vector machine with O (log NM) run time in both training and classification stages can be implemented [22].

There are many parallel support vector machines implemented, but there is no clear suggestion for every application situation. Many factors, including optimization algorithm, problem size and dimension, kernel function, parallel programming stack, and hardware architecture, impact the efficiency of implementations. It is up to the user to balance trade-offs, particularly between computation time and classification accuracy [38].

Parallel computing of SVMs is becoming a necessity for improving the performance of SVMs for big data and has already demonstrated promising results for enhancing large-scale problems [38]. The challenge due the big data is the improvement regarding computation time, accuracy, scalability, and memory issue, sowing to the immense an increasing size of real-life data requiring a reasonable choice for end-users [38].

The efficient hardware implementation of cascade support vector machines is optimized to efficiently handle problems where the data belongs to one of the two classes, such as image object classification, and hence can give speedups over single SVM classifiers. However, SVM classification is a computationally challenging task, and existing hardware architectures for SVMs consider only unified classifiers. This model is used to design low-cost parallel SVM coprocessors and intelligent embedded systems for on-line real-time classification applications to allowing SVM architectures to tackle larger-scale problems [40].

The quickening cascade SVMs through a hybrid processing hardware architecture is optimized for the cascade SVM classification flow. Accompanied by a method to reduce the required hardware resources for its implementation and algorithm to improve the classification speed by utilizing cascade information to discard data samples [40].

A parallel algorithm of a local support vector machine, called kSVM, is proposed for the effectively non-linear classification of large datasets. It uses k means algorithm to partition the data into k clusters followed by anon-linear SVM in each cluster to classify the data in a parallel way on multi-core computers. The kSVM algorithm is faster than the standard SVM in the non-linear classification of large datasets while maintaining the classification correctness [41].

A coarse-grained parallel genetic algorithm (CGPGA) is used to optimize the feature subset and parameters for SVM simultaneously. The distributed

topology and migration policy of CGPGA can help to find optimal feature subset and parameters for SVM in a significantly shorter time, to increase the quality of the solution found. The new fitness function that joins the classification accuracy is obtained from the bootstrap approach, several features, and several support vectors are proposed to lead the search of CGPGA to the direction of optimal generalization error. However, two problems must be efficiently addressed for SVM, feature selection, and parameter optimization [42].

The spread topology and migration system of CGPGA enable to search for the solution space with different search strategies in a parallel way, thereby providing strong search ability and high efficiency. The approach is not only optimized SVMs' model parameters but also efficiently obtained the discriminating feature subset. The proportion of support vectors in the model produced by the method was maintained at a low level. So, the classification is faster on the unseen new pattern's applications were extended to more broad fields where classification has to be done at high speed [42].

Singh, Dinesh and et al [43]. Had presented a distribution preserving of kernel support vector machine (DiP-SVM) model. The first and second-order statistics of the entire dataset are retained in each of the partitions. The DiP-SVM is achieved a minimal loss in classification accuracy among other distributed support vector machine techniques on several benchmark datasets [43].

The function of a learning support vector machine for large datasets has been performed by splitting the dataset into manageable sized and training a sequential support vector machine on each of these partitions separately to obtain local support vectors. While distributed SVMs have proven to be much faster than sequential SVMs on large datasets. However, this process regularly leads to the loss of classification accuracy as global SVs have not been chosen as local SVs in their respective partitions [43].

Kshirsagar and et al. (2018) [45] had used Parallel computing framework is used to accelerate the SVM-based classification. The graphics processing unit (GPU) has the characteristics of multi-threads and powerful parallel processing capability. The general-purpose computing with GPU (GPGPU) is developed. It is a new area due to the highly parallel nature of GPU. With this GPU, parallel computing can be achieved with low cost and low power consumption [45].

The given GPU has achieved maximum speed up with high accuracy. This speedup can be further increased for a given number of training samples by using GPUs having more compute capability. This approach can be extended for multi-classification by using parallelism related to both CPUs and GPUs. This approach can be more beneficial with more complex datasets [45].

Singh et al. (2018) [46] had proposed Projection-SVM, a distributed implementation of kernel support vector machine for large datasets using subspace partitioning. A decision tree is constructed on the projection of data along the direction of maximum variance to obtain smaller partitions of the dataset. On each partition, a kernel SVM is trained independently over a cluster, thereby reducing the overall training time and reducing the prediction time significantly [46]. The distributed SVM is trained in the model faster and requires less time in prediction for new data points. The dominant eigenvector and decision tree for the partitioning of the dataset are less expensive computation costs in comparison to the kernel k-means approach with complexity as proposed in [47] [48]. So, the proposed approach also achieves excellent classification performance with small accuracy changes.

Vivekanandan, Swathi and et al. (2018) [49] had adapted A Parallel Support Vector Machine for big data analysis due to its limitation in handling big data. The analysis is taken from the heart Disease dataset, and the performance of classification algorithms is compared to other frameworks, it found that the Parallel Support Vector Machine outperforms different algorithms. In the case

of Big Data, SVM is identified to suffer from slow processing time. Hence, Parallel SVM based classification is preferred to classify the large-scale dataset. It has enormously reduced the execution time and also classifies the data accurately [49].

A new combined solution based on parallel and approx SVM for Big Data classification using the extended versions of the Support Vector Machines (SVMs) is proposed. This combination was given the name Parallel Support Vector Machines (PSVM). The main disadvantage of a PSVM model is that the feature can be removed over time, so the accuracy is decreased. To solve this problem, they used an approach that approximates any SVM model based on the Radial Basis Function (RBF) kernel, which has been called the Approx SVM. This new approach helped to overcome two main problems, which are the inability to handle large-scale datasets and the change of attributes' numbers over time. The parallel SVM has the advantage of decreasing the execution time when building the classification model. So, the researchers in this paper had obtained exciting results in terms of accuracy compared to the standard SVM. Besides, the parallel approx SVM considerably decreased the time needed to build the new model when there is new data over time [50].

Sadasivam, G. Sudha, et al. (2018)[51] had proposed a new parallel approach and classification, which consists of the preprocessing of data, data selection, or feature extraction. The feature selection methods have been analyzed for the extraction of datasets, these are a support vector machine with recursive feature elimination (SVM-RFE), minimum redundancy maximum relevance (mRMR), principal component analysis (PCA), successive feature selection (SFS) and independent component analysis (ICA) [51]. A powerful method to determine kinship relations between a given pair of facial images using feature descriptors to learn the SVM classifier is proposed. The feature descriptors are used to extract the salient facial features. These extracted facial features are then concatenated to create a high-dimensional feature vector.

Support Vector Machine (SVM) learns these high-dimensional feature vectors to classify facial images based on feature similarities. (KinFaceW-I) Dataset is used to validate the Kinship Verification accuracy. A positive kinship pair corresponds to a real or own parent-child pair. While negative kinship pair corresponds to a pair of one's parent with another's child **[52].**

Rezvani, Wang, and et al. (2019) had proposed a new Fuzzy twin support vector machine (FTSVM) model for solving binary classification problems that combines the idea of intuitionistic fuzzy number with twin support vector machine (TSVM). An adequate fuzzy membership is employed to reduce the noise created by the pollutant inputs, which is a useful machine learning technique that can overcome the negative impact of noise and outliers in tackling data classification problems. Linear and nonlinear functions are used to formulate two nonparallel hyperplanes. An IFTSVM not only reduces the influence of noises, but it also distinguishes the noises from the support vectors. Further, this modification can minimize a newly formulated structural risk and improve the classification accuracy. The outcome shows that an IFTSVM is able to produce promising results as compared with those from the original support vector machine, fuzzy support vector machine. However, it is sensitive to C, in which, if it is not appropriately chosen, the IFTSVM produces inferior results. Our future work is focused on enhancing the structure of the IFTSVM to solve the imbalance classification problems. [53]

2.4.3.Hadoop and Radoop

Reference [27] had developed a Fast Parallel SVM Algorithm for Massive Classification Tasks. It extends a recent finite Newton classifier for building a parallel incremental algorithm. Newton's method minimizes a quadratic approximation to the function we are interested in. If there are millions of data points, the Newton SVM algorithm can classify them in minutes on a PC. Although the Newton SVM algorithm is fast and efficient to classify large

datasets, it needs loading the whole dataset in memory. The new algorithm uses graphics processors to gain high performance at low cost, and only subsets of the data are considered and loaded in memory at any time. In contrast, the solution is updated in the growing training set. In synthesis, the authors have extended Newton SVM in two ways. 1) Developed an incremental algorithm for classifying massive datasets (billions of data points) of dimensionality up to 103. 2) Using a GPU (massively parallel computing architecture), 3) developed a parallel version of the incremental Newton SVM algorithm to gain high performance at a low cost. In Priyadars et al. [3], an algorithm for MapReduce based SVM is implemented, which runs on several size files, and training time have been calculated on the Hadoop cluster.

Prekopcsak, Zoltan, et al [20]. Had presented an extension for the RapidMiner data mining tool called Radoop, which provides the use of operators for running distributed processes on Hadoop. They described integration and development details and provide runtime measurements for several data transformation tasks. Radoop is an efficient extension for big data analytics and scales well with increasing data set size and the number of nodes in the cluster [20].

Distributed computing is an excellent promise for handling large data, but it is not familiar with a single machine. It needs new programming models and tools that can be used for data analysis. Many projects aim to solve efficient data access and provide different data analytics functions in a distributed environment. Still, they usually need complex command-line mechanisms or even programming to make them work. The RapidMiner, a data mining suite, hides all the complexity of distributed data analysis and provides big data processing techniques in the familiar analytics environment [20].

ABDAR and Moloud (2015) [23] had used RapidMiner and IBM SPSS Models data mining tools together. The two above tools examined the accuracy of different data mining algorithms such as C5.0, C4.5, Decision

tree, and Neural Network. It was predicting the prevalence of these diseases or early diagnosis of them using these algorithms. According to the results, the C4.5 and C5.0 algorithms by using IBM SPSS Modeler and Rapid Miner tools had 72.37% and 87.91% of accuracy, respectively [23].

Radoop [26, 27] is a product that resulted from the integration effort between the open-source data analytics tool RapidMiner [28] and Apache Hadoop. RapidMiner is a machine learning software environment that possesses data mining, text mining, predictive analytics, and business analytics capabilities. RapidMiner has excellent graphical and visualization capabilities that combine algorithms. It is an efficient tool compare with other open-source analytics tools. Such as Weka and R. Radoop claims the capabilities of multiple machine learning algorithms, including regression, classification, and clustering, scoring these models on Big Data in Hadoop and in-memory analytics with RapidMiner operators either on Big Data subsamples or iteratively on all splits of the data [28]. However, there is not a clear list of supported machine learning training algorithms and their performance benchmarks [23].

Zheng, Jiang et al. (2014) [24] had discussed the situation and limitations of current approaches, analytic models, and tools utilized to conduct predictive machine learning analytics for huge volumes of data where the data processing causes the processor to run out of memory [24].

Bello-Orgaz, Gema et al. (2016)[25] had presented a new methodology that is produced to allow for accurate data mining and information fusion from social media and of the latest applications and frameworks that are currently appearing under the social networks, social media and big data paradigms. Different big data frameworks, like Apache Hadoop, Spark, has allowed for the efficient utilization of data mining machine and learning algorithms in different areas. Social big data comes from joining the efforts of the two previous domains: social media and big data. MapReduce is presented as one

of the most efficient big data solutions. This programming paradigm and its related algorithms were developed to provide significant improvements in large-scale data-intensive applications in clusters. MapReduce delivers an excellent technique to work with large datasets. The algorithm can split the large data into small pieces and process it in a parallel manner [25].

SVM classifier depends on the number of support vectors required. In SVM classification, the memory needed to store the support vectors is directly proportional to the number of support vectors. Observations and Result analysis show that in Sequential SVM as the number of instances increases, training time also increases. Also, in the Hadoop Cluster, it has been verified and validated that as the number of nodes increases, for the large size of Input data, execution time decreases. From this, it is shown that Parallel SVM using MapReduce Model performs efficiently. An advantage of using HDFS & MapReduce is the data awareness between the NameNode & DataNode and also between JobTracker & TaskTracker [31].

An algorithm of parallel naive Bayes is proposed and implemented to solve the problem of the Chinese text data, because this data is increasing on the internet, making it challenging to classify data by using spark platform for big data [39]. The authors used parallel computing in the entire training and prediction of naive Bayes classifier using resilient distributed datasets (RDD). PNBA was then implemented in Hadoop. The result was compared to the Spark. It was found that the Spark PNBA gives more accuracy than the Hadoop PNBA, especially in terms of speed and scalability [39].

The mixture of general-purpose graphics processing unit (GPGPU) computing and MapReduce method on an Apache Hadoop framework is proposed to deal with computational complexity and the large volume of data. The experimental results show improved time efficiency in feature extraction and classification. The parallel version of the proposed methods using CPU+GPU clusters improves the time efficiency for feature extraction. By

using Hadoop clusters, the training time of an SVM can be reduced. A limitation in SVM-RFE is that parallelization is achieved only on SVM training. As the process of feature elimination is recursive, it cannot be parallelized. MapReduce programming requires data to be independent, as parallelism is achieved by splitting data into blocks. [51].

Many parallel support vector machines (PSVMs) are implemented, but there is no clear suggestion for every application situation. Many factors, including optimization algorithm, problem size and dimension, kernel function, parallel programming stack, and hardware architecture, impact the efficiency of implementations. It is up to the user to balance trade-offs, particularly between computation time and classification accuracy [38]. Parallel computing of SVMs is becoming a necessity for improving the performance of SVMs for big data and has already demonstrated promising results for enhancing large-scale problems. The challenge due the big data is the improvement regarding computation time, accuracy, scalability, and memory issue, sowing to the immense an increasing size of real-life data requiring a reasonable choice for end-users [38].

2.5. Chapter summary

This chapter considered the basic concept of the Support vector machine, and it is parallel implementation, definition, and the basic idea of the k-means clustering algorithm. Then the architecture of the PSVM and its flows is presented. At last, the chapter illustrates the existing methods of many implementations of parallel computing using different techniques.

CHAPTER THREE

RESEAECH METHODOLOGY

3.1 Methodology Framework of proposed Implementation

Big Data management has gained the importance with the development of two hasten trends in the field of Machine Learning Technology. The effective use of Big Data is a key basis of competition and delivering a new wave of production growth [62]. Classification is one of the data mining mechanisms used for classifying the unstructured data into the structured class, and it helps the user for knowledge and plan. The framework in Figure 3.1 (a,b,c)shows the flow of the steps of the classification task. The organization of these steps as follows:

- Data collection
- Data preprocessing
- Apply the four data sets on single SVM with different kernels and parameters.
- Next, apply the k-means clustering to SVM as one model with different kernels and parameters and then compare their results.

- Then, apply PSVM with Hadoop cluster using Radoop extinction for Big Data classification.
- The results of the SVM model, result of the k-mean applied to SVM model, and the results of Parallel Support Vector Machine are compared.

Figure 3.1(a): Flow of classification task of single SVM.

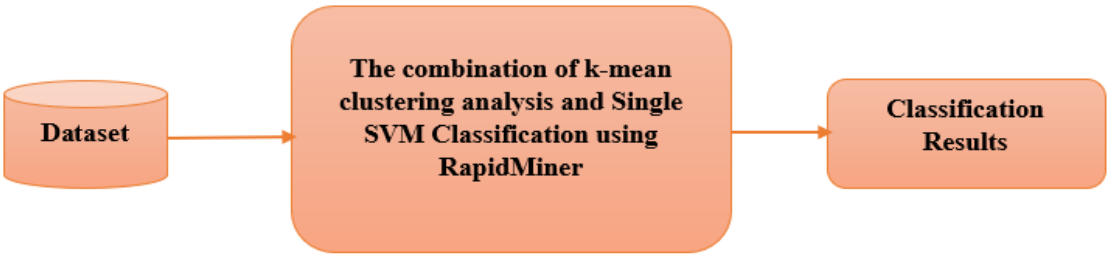


Figure 3.1(b): Flow of classification task of the k-means combine to SVM.

Figure 3.1(c): Flow of classification task of PSVM

3.2. Dataset Description

Four datasets are used in this thesis. The first dataset is the Adult dataset; this data extracted from the UCI repository [76]. In the adult database, 42 attributes classifying into two classes. Each attribute denoted by binary variable (0 or 1). Labels are indicated by (+1 or -1) [76]. The second dataset

is Diabetes data set, this data has been prepared to analyze factors related to readmission as well as other outcomes about patients with diabetes. The dataset represents ten years (1999-2008) of clinical care at 130 US hospitals and integrated delivery networks. It includes over 50 features representing patient and hospital outcomes information extracted from the database for encounters that satisfied the necessary criteria. The third dataset is the River Nile water quality dataset. This dataset was collected and prepared by the ministry of health and different water stations in Directorate General of Preventive Medicine (DGPM) in Sudan from (2006-2017). This data contains 20 full chemical and physical parameters are used to predict if this water is suitable for drinking or not. The fourth dataset is the Forest Cover type datasets [76]. From the UCI repository. Independent variables were derived from data obtained from the US Geological Survey (USGS) and USFS data. Data is in raw form and contains binary (0 or 1) columns of data for qualitative independent variables [76]. The description of the four datasets is shown in Table3.1.

Table3.1: Dataset Description

Dataset	**Dataset characteristics**	**Attribute Characteristics**	**Associated Tasks**	**Instances**	**Attributes**	**Missing Values**	**Area**
Adult	Multivariate	Categorical, Integer	Classification	48842	14	Yes	Social
Diabetes	Multivariate	Integer	Classification	100000	55	Yes	Life
Water quality	Multivariate	Categorical, Integer	Classification	888	20	yes	Life real

Cover type dataset	Multivariate	Categorical, Integer	Classification	581012	54	no	life

3.3. Dataset Loading

RapidMiner tool is used to import the datasets from a computer as Exel or CSV files to implement the SVM and k-means combine to SVM to do the classification process. The HDFS in Hadoop is used to store the dataset in the hive table. Then the dataset is retrieved from the Hive table using one of the Radoop extension operators from the Hadoop cluster to do the classification process. In the Hadoop cluster platform, the job of MapReduce is done to partition the data into equal groups.

3.4. Introduction to RapidMiner Environment

RapidMiner is an excellent tool for conducting data mining workflows for various tasks, ranging from different areas of data mining applications to different parameter optimization schemes [63]. It provides an integrated environment for data preparation, machine learning, deep learning, text mining, and predictive analytics. One of the main benefits of RapidMiner is its advanced ability to do the process of program execution of complex workflows, all this is done within a visual user interface, without the need for traditional programming skills.

RapidMiner is used for different business and commercial applications, as well as for research, education, training, rapid prototyping, and application development. RapidMiner works very well with all steps of the machine learning process, including data preparation, results in visualization, model validation, and optimization. Also, RapidMiner provides many extensions.

Radoop extension that removes the complexity of data prep and machine learning on Hadoop and Sparks that deal with big data analysis [63].

3.5. Data Preprocessing

Working with big data is very difficult and needs to work in an appropriate environment. The four datasets shown in table1 varied in sample size and dimension. Some data cleaning performed to deferent datasets. This data also had to be transformed into a format suitable for SVMs, to improve the performance. Hence, preprocessing was required. There are missing values in most of the data set, which decrease the accuracy level and performance of classification. So, data cleaning performed to removing the missing value. The RapidMiner tool is used to do the preprocessing, replace missing value operator is used to remove the missing values from the attributes. Missing values can be replacing by the minimum, maximum, zero, or average cost of that Attribute. The nominal to the numerical operator is used to changing the type of non-numeric attributes to a numeric type because the SVM did not deal with non-numeric data. This operator does not only change the type of selected attributes, but it also maps all values of these attributes to numeric values.

3.6. K-mean Clustering Analysis

The primary task of clustering [64]. Is to group the objects into clusters; the objects in the similar cluster are more alike than those in various clusters. The clustering can find the relationships amongst data objects in an unsupervised way. Many clustering algorithms have introduced and developed; using this clustering algorithm enhances the efficiency, and accuracy performance. According to cluster mode, clustering algorithms can be categorized into centroid-based clustering, hierarchical clustering, distribution-based clustering and density-based clustering [64].

K-means clustering is an unsupervised algorithm that is working on the similarity. It is an iterative algorithm considerably used in the data mining field; K-means is an efficient algorithm in partitioning the data points [54, 55]. K-means is simple and effective, so it used only one parameter that indicates the number of clusters, the user determines this parameter. The k-mean uses kernels to estimate distances between examples and clusters. The calculation of one distance needs to sum over all Examples of a cluster [54, 55]. So, this algorithm is quadratic in the number of examples and does not return a Centroid Cluster Model. After clustering, some clusters contain the data of two class labels called duo-cluster [65].

The centroid is the position of the center in the n-dimensional space of the n Attributes of the Example Set. It was determined the specific cluster in the k-means algorithm. The k-means algorithm starts points (k) are randomly drawn examples of the input Example Set. All example sets are assigned to their nearest cluster, and they are used to recalculate the centroids of the clusters. These steps will be repeated for the new centroids until the max optimization steps are reached. The procedure is repeated max runs times (max run =10, k=2) with different sets of start points [65].

K-mean is a heuristic algorithm, the result may depend on the initial clusters, so there is no assurance that it will converge to the global optimum. As the result of the K-means algorithm is uncertain, we usually run it multiple times, and cluster result is determined through a voting mechanism [54, 55]. The steps of this algorithm are described as follows.

- First, we randomly choose K points in the database as the initial cluster center.
- Repeat the first step.

- Each object is assigned to the most similar cluster based on the mean value of the objects in a cluster.
- Update the mean value of a cluster
- Until the mean values of clusters do not change.

The k-means algorithm initially selects k objects. The remaining objects are assigned to their cluster with the most similarity according to the length between the object and the cluster mean. Next, it computes the new measures of the mean for each cluster iterating until the centroid function converges [66]. Generally, the square-error criterion used, which is defined as follows:

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} |p - m_i|^2 \qquad (1)$$

Where E is the sum of the squared error for all in the dataset; P is the mean of cluster instances when both P and m are multidimensional. That is using for every object in every cluster. The length from the object to its cluster center is squared, and summed up.

Given a set of d-dimensional vectors $D = \{x_i \mid i = 1,..., N\}$, and the k-means algorithm is initialized by selecting k (centroids) randomly [35], the algorithm proceeds by shifting between two steps till convergence:

1. Data Assignment. In iteration *t*, each data point is assigned to its closest centroid

$$S_i^{(t)} = \left\{ x_i : \left\| x_j - m_i^{(t)} \right\| \le \left\| x_j - m_{i^*}^{(t)} \right\| \; for all i^* = 1,..., k \right\} \qquad (2)$$

2. Relocation of means. Calculate the new centroid of the data points in the cluster.

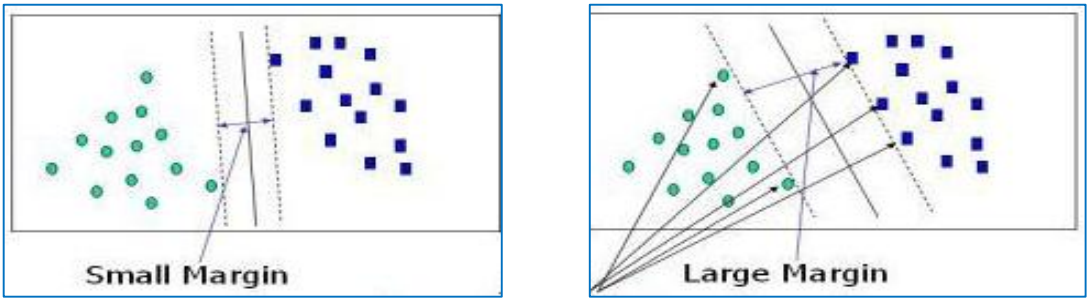
$$m_i^{(t+1)} = \frac{1}{\left| S_i^{(t)} \right|} \sum_{x_j \in S_i^{(t)}} x_j \qquad (3)$$

3.7. Support Vector Machine

Supervised learning uses the collected information as training data and produces a model, which is a function that, if given input to then generates the required output. A Classification algorithm is a method for selecting a hypothesis from a set of options that best fits a set of observations. A Support Vector Machine is a discriminative classifier formally defined by a separating hyperplane. The algorithm outputs new examples by an optimal hyperplane [54].

SVM is a very powerful classifier for handling large datasets in high dimensional space with a robust mathematical property that is quadratic optimization problem. However, it has a high computational cost. Thus, this results in more training time for large datasets [54].

Support Vector Machine (SVM) is introduced in [34]. SVM is a supervised learning algorithm that is used for classification and regression [35]. The essential goal of SVM is to find the unique hyperplane with the maximum margin that can linearly separate the classes, as shown in (Figure1). When the training data is not linearly separable data in the input space, it can be projected to a feature space of higher dimension by using the SVM kernel functions, in which the linear separation becomes easier. Figure3.3 shows SVM classification when it is linear separable or nonlinearly separable [3]. Many researchers had studied and applied SVM in many practical fields. Their computational and storage requirements increase rapidly with the number of training vectors, and this is demonstrated in different problems of practical interest out of their reach [67]. Support vector machine learning aims



to classify data sets where the number of training data is small and where regular use of statistics of large numbers cannot assure an optimal solution. Two decision boundaries on the same data are shown in Figure 3.2[36].

Figure3.2: Left: two classes showing a small margin Right: two class large margin [36].

Support Vector Machines are prevalent for their strong theoretical foundations, performance, generalization, and capability to handle high-dimensional data. In binary classification, if (xi, yi) … (x_n, y_n) are the training data set where x_i are the vectors constitute the instances and $y_i \epsilon$ {-1, +1} are the labels of those instances. An optimum hyperplane was built by SVM, which linearly discriminates in a higher dimensional feature space that chooses the largest margin separation between the two classes. The SVM classifier is shown in Figure 3.2. The solution of SVM obtained by minimizing the primal objective function, and this is shown in equation (4) [69].

$$\min_{w,b} \; j(w,b) = \frac{1}{2} \square w \square^2 + c \sum_{i=1}^{n} \xi_i$$
$$with \forall_i \begin{Bmatrix} y_i (w.\Phi(x_i) = b \geq 1 - \xi_i) \\ \xi_i \geq 0, \end{Bmatrix} \qquad (4)$$

In equation (4) w is the coefficient vector of the hyperplane, b is the offset, yi is the labels. Φ (.) is the mapping from input space to feature space, and ξ_i are the slack variables that permit the non-separable case by allowing misclassification of training instances. The convex quadratic programming (QP) problem in equation (5) solved by optimizing the dual cost function:

$$\max_{\alpha} G(\alpha) = \sum_{i=1}^{N} \alpha_i y_i - \frac{1}{2} \sum_{i.j} \alpha_i \alpha_j K(x_i . x_j).$$

$$subject to \begin{cases} \sum_i \alpha_i \\ A_i \le \alpha_i \le B_i \\ A_i = \min(0, cy_i) \\ B_i = \max(0, cy_i) \end{cases} \quad (5)$$

In equation (5) K (x_i; x_j) = ($\emptyset(x_i)\emptyset(x_j)$) is the kernel matrix representing the dot products $\emptyset(x_i)$. $\emptyset(x_j)$ in feature space. The general SVM can describe as follows. Let l training samples be T= {(x_1, y_1), ⋯, (x_l, y_l)}, where $x_i \in Rn$, $y_i \in$ {1, −1} (classification) or $yi \in R$ (regression), i=1, ⋯, l. The nonlinear mapping function is $\emptyset(x_i)$ entailing a kernel K (x_i, x_j) = $\emptyset(x_i)$ $\emptyset(x_j)$. Classification SVM implemented through solving the following equations [68].

$$\min_{w\xi_i b} \left\{ \frac{1}{2} \square w \square^2 + c \sum_i \xi_i \right. \quad (6)$$

$$s.t. y^i (\Phi(x_i) w +) \ge 1 - \xi_i \forall_i = 1, ..., n \quad (7)$$

The classification regulation of the SVM model can be calculated as

$$Accuracy = \frac{\#correctly\ predicted\ data}{Total\ testing\ data} * 100\% \quad (8)$$

It is essential to choose the appropriate kernel function of SVM. The kernel function must satisfy the Mercer condition. Many kernel functions models have developed. Commonly used kernel functions are included in Table3.2 [69, 38].

Table3.2: Examples of Well-Known Kernel Functions

Kernel Function	**Inner Product**	**Kernel Type**
Linear kernel	$K(x_i . x_j) = x_i T x_j$	Linear

Gaussian/Radial-Basis Function (RBF)	$K(x_i.x_j) = \exp(-\gamma \left\|x_i - x_j\right\|^2)^d, \gamma > 0$	Non-linear
Polynomial	$K(x_i.x_j) = (\gamma x_i T x_j + r)^d, \gamma > 0$	Non-linear
Sigmoid or Laplacian	$K(x_i.x_j) = \exp(-\gamma \left\|x_i - x_j\right\|^2)^d, \gamma >$ 0 Here, γ, r, and d are kernel parame	Non-linear

Support Vector Machine Algorithm steps:

- Define an optimal hyperplane and find the maxim margin.
- Extend the above definition for non-linearly separable problems.
- Finally, the data is mapping to high dimensional space where it is easier to make the classification process.

There are several essential extensions on the above basic formulation of SVM. The soft margin idea was included to extend the SVM algorithm so that the hyperplane allows a few of such noisy data to exist. To solve the problems that require more than two classes. A linear binary classification task is shown in figure 3.3 [41]. We can repeatedly use one of the classes as a positive class and the rest as the negative classes to train several SVM models, SVM can be easily extended to perform regression analysis [16].

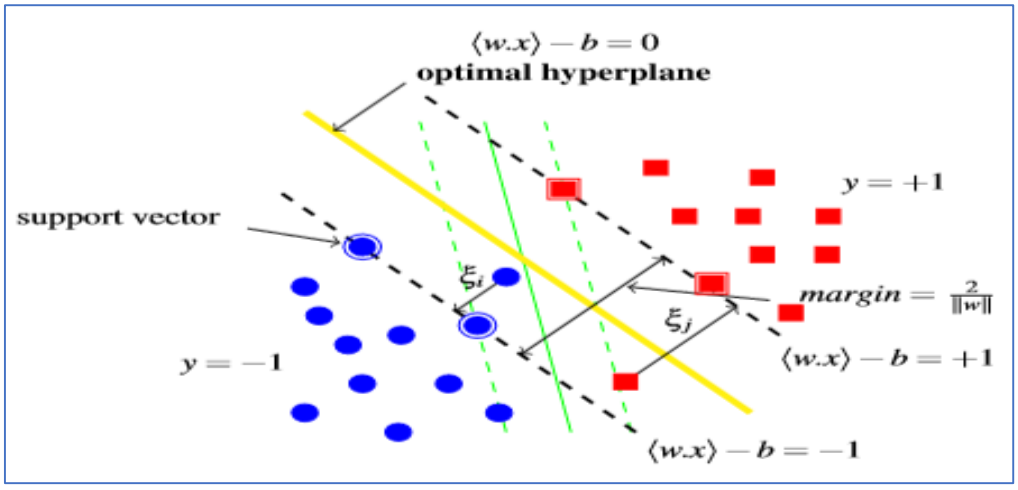


Figure 3.3: Linear separation of the data points into two classes.

The RapidMiner tool is used to train the support vector machine, and four data sets shown in table1 are applied on it. The process of SVM is trained with different kernel types likes (dot, polynomial, radial), and the parameters gamma, C, and epsilon are used as fixed values. Dealing with big data with the use of traditional support vector machine, it notes that when the training data increases, the training time increases, leading to weak performance. So, k-mean clustering is used to reduce the number of training examples producing better accuracy and time consuming.

3.8. k-means clustering applied to SVM

Parallel Hyperplane: A hyperplane H is the set of points ($x_1, x_2, x_3, \ldots x_n$) that is satisfy a linear equation is:

$a_1x_1 + a_2x_2 + a_3x_3 + \ldots\ldots\ldots + a_nx_n = b_1$ (original hyperplane equation).

Normal Vector = ($a_1, a_2, a_3, \ldots\ldots, a_n$)

$b_1y_1 + b_2y_2 + b_3y_3 + \ldots\ldots\ldots + b_ny_n = b_2$ (second hyperplane equation)

$c_1z_1 + c_2z_2 + c_3z_3 + \ldots\ldots\ldots + c_nz_n = b_3$ (third hyperplane equation)

The distance from the original hyperplane is:

$$d = \frac{b}{\sqrt{a_1^2 + a_2^2 + a_3^2 + \ldots. + a_n^2}} \qquad (9)$$

The expressions below explain the multiple parallel hyperplanes equations at different distances: $a_1x + b_1y + c_1z + d_1 = 0$, $a_2x + b_2y + c_2z + d_2 = 0$, $a_3x + b_3y + c_3z + d_3 = 0 \ldots\ldots$ $a_nx_n + b_ny_n + c_nz_n + d_n = 0$.

Two planes $a_1x + b_1y + c_1z + d_1 = 0$ and $a_2x + b_2y + c_2z + d_2 = 0$ are parallel if $a_1 = k\, a_2$, $b_1 = k\, b_2$ and $c_1 = k\, c_2$. The distance between $a_1x + b_1y + c_1z + d_1 = 0$ and $a_2x + b_2y + c_2z + d_2 = 0$ is equal to the distance from a point (x_1, y_1, z_1) on the first plane to the second plane the parallel hyperplane is shown in Figure 3.4 [54].

$$\frac{|ax_1 + by_1 + cz_1 + d|}{\sqrt{a^2 + b^2 + c^2}} = \frac{|d_2 - d_1|}{\sqrt{a^2 + b^2 + c^2}} \qquad (10)$$

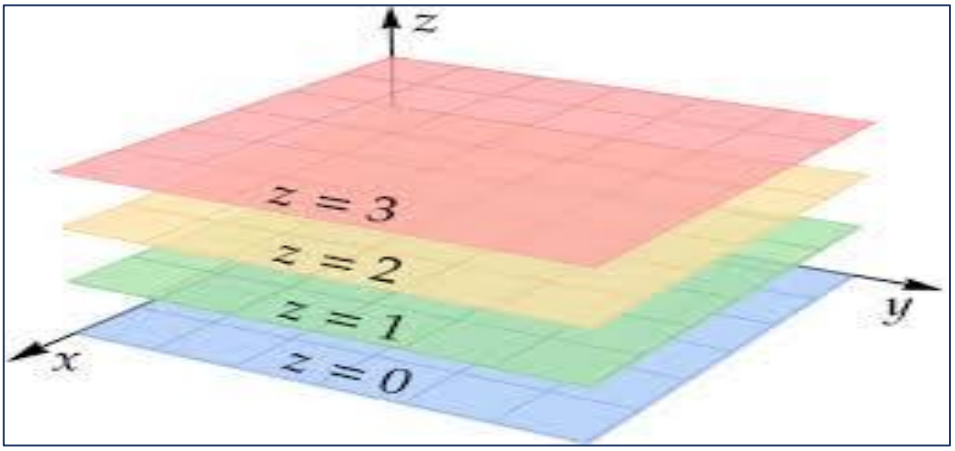


Figure 3.4: parallel Hyperplane.

By using the RapidMiner platform, the k-means clustering algorithm is used for clustering the data set. It is used to group the original training data points into k clusters. Note that when we apply the K-mean cluster with SVM, the final result of SVM is dependent on the amount of k. k-means use kernels to estimate distances between Examples and clusters, this is shown in Figure 3.5. The k-means work to select the most informative samples. SVM classifier built through training on those selected samples, experiments show that this model reduces the scale of the training set, thus effectively saves the training and predicting the time of SVM, and guarantees the generalization problem and performance. The experiments will be discussing in more detail in chapter4.

The classification process based on a combination of k-mean clustering and SVM can be summarized as follows [70].

- Preprocess the collected data sets. (Replace missing values).
- Some samples are selected to process with k-mean.
- Partition the samples into *k* parts.
- The output samples from the k-mean, which are two partitions, are taken as the input of the SVM.
- Classification result.

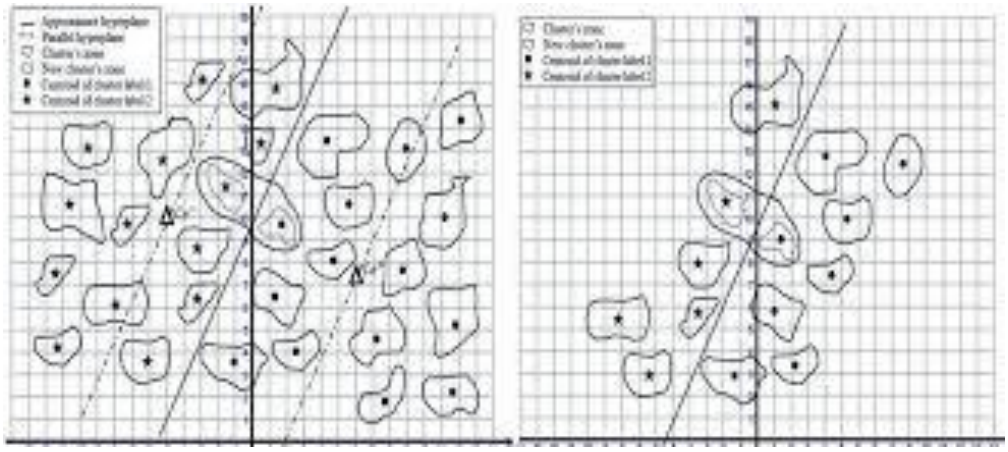


Figure 3.5: (a) Original data set (b) after one iteration of data removal

K-means clustering parameter set as k =2. Then, the SVM. Then the k-means clustering is applied to SVM with three kernels (dot, polynomial, and Radial). The parameters are set as a fixed value as (gamma =1.0, C=0.0, convergence epsilon = 0.01). The accuracy result shows that applying k-means to SVM gives better performance than using SVM only. The experiments will be discussing in more detail in chapter4.

3.9. Parallel Support Vector Machine (PSVM)

There are many parallel implementations for support vector machines (SVMs), but there is no clear suggestion for every application situation. Many factors, including optimization algorithm, problem size and dimension, kernel function, parallel programming stack, and hardware architecture, impact the efficiency of implementations. It is up to the user to balance trade-offs, particularly between computation time and classification accuracy [38].

Parallel computing of SVMs is becoming a necessity for improving the performance of SVMs for big data and already has demonstrated promising results for improving large-scale problems. The challenge due the big data is the improvement regarding computation time, accuracy, scalability, and memory issue, sowing to the immense an increasing size of real-life data requiring a reasonable choice for end-users [38].

The acceleration of cascade SVMs through a hybrid processing hardware architecture is optimized for the cascade SVM classification flow. Accompanied by a method to reduce the required hardware resources for its implementation, and a method to improve the classification speed by utilizing cascade information to discard data samples [40].

The Cascade SVM is a series of distinct stages procedure that combines the results of multiple structured support vector machines to create one model, this model is shown in Figure 3.6. The Cascade SVM presents several benefits over a single SVM because it can reduce computation time and storage-requirements [71]. The main idea is to repeat and reduce a data set to its crucial data points before the last step is reached. Locating potential support vectors and removing all other samples from the data are done by the following steps: [72].

1. Preprocess the collected sample datasets
2. Partition the data into *n* subsets of equal size.
3. Independently train an SVM on each of the data subsets.
4. Combine the SVs of the pairs of SVMs to create new subsets.
5. Repeat steps 2 and 3 for some time.
6. Train an SVM on all SVs that finally obtained in step 4.

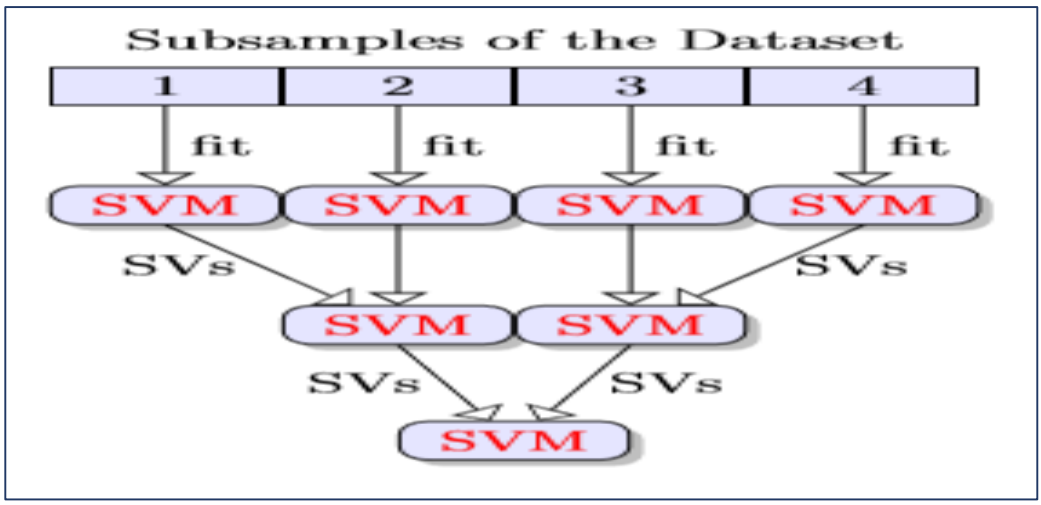


Figure 3.6: training flow of parallel Support Vector Machine.

This Cascade SVM algorithm is considered the possibility of multiple runs through the cascade for each data set. After that, subsets of the first step for the next term created by combining the remaining SVs of the final model with each subset from the first level of the first run [72].

The architecture of parallel SVM is a hierarchal structure. The low-level SVM training has to perform when all the upper-level sub SVM trained. In the last level of the architecture, all the support vectors should include in the training samples. The sample size must be more significant than the number of support vectors. When the ratio between the support vector and training sample is bigger, the speedup will be less. It is the shortcoming of the cascade SVM mode [72].

The SVM has a problem with quadratic programming. Improving computation speed through parallelization is difficult due to dependencies between the computation steps.[20] A mixture of several SVMs is used, each of them has a weight and trained only on the part of the data set [68, 73]. The training method is given as follows:

1. Partition the dataset into *P* blocks D_1, . . ., D_P, each processor handles roughly *N/P*
2. Processor Pr reads the part of dataset Dr based on its responsibility, and builds local SVM Sr.
3. Processor P_0 trains the weight matrix $w \in R^{P\times N}$ by minimizing cost function

$$C = \sum_{i=1}^{N}\left[\tanh\left(\sum_{r=1}^{P}\omega_{ri}S_r(\vec{x}_i)\right)-y_i\right]^2 \qquad (11)$$

Where Sr ($\vec{x}_i$) is the output of Sr given input $\vec{x}_i$

The method to divide the problem into smaller tasks is proposed in reference [74]. In each task, specific parts of α are chosen to be optimized, while the rest of α remains in constant value. The selected part is called the working set.

Then the program repeats the select optimize process until global optimality conditions are satisfied [74]. Let B denote the working set with n variables, and N denotes the non-working set with (l − n) variables. Then, α, y and Q can be correspondingly written as:

$$\alpha = \begin{vmatrix} \alpha_B \\ \alpha_N \end{vmatrix}, y = \begin{vmatrix} y_B \\ y_N \end{vmatrix}, Q = \begin{vmatrix} Q_{BB} & Q_{BN} \\ Q_{NB} & Q_{NN} \end{vmatrix} \quad (12)$$

Thus, the small task can be written as:

$$\min \frac{1}{2}\alpha_{\mathbf{B}}^T Q_{BB\alpha B} - \alpha_{\mathbf{B}}^T (1 - Q_{BN\alpha N} + \frac{1}{2}\alpha_{\mathbf{B}}^T Q_{NN\alpha N} - \alpha_{\mathbf{B}}^T \quad (13)$$

Subject to

$$\alpha_{ByB}^T + \alpha_{NyN}^T = \mathbf{O} \quad (14)$$

$$\mathbf{O} \le \alpha_B \le C \quad (15)$$

The disadvantage with regular SVM when the data is growing is 1) Its unreasonable algorithmic complexity, the extreme memory requirement of the required quadratic programming in large scale datasets.2) Its speed and size in both the training and testing phase.3) The weakness of the performance measures. An efficient parallel support vector machine algorithm and its implementation are essential to work with large scale data [3]. The general SVM training algorithm can be summarized as follows:

1. Choose a kernel function k ($\vec{x}_i$, $\vec{x}_j$)

2. Maximize the function below, subject to $\alpha_i \ge 0 and \sum_{i=1}^{N} \alpha_i y_i = 0$

$$W(\vec{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N} \alpha_i \alpha_j y_i y_j k(\vec{x}_i, \vec{x}_j) \quad (16)$$

In equation (16) α_i are non-negative Lagrange multipliers, it indicates the support level of instance x_i to the hyperplane. In case of $\alpha_i = 0$means that removing x_i from training set does not interfere the position of hyperplane [71].

3. The bias *b* is found as follows:

$$b = \frac{1}{2}\left[\min\left(\sum_{i|y_i=1} \alpha_i y_i k(\vec{x}_i, \vec{x}_j) \right) + \max\left(\sum_{i|y_i=1} \alpha_i y_i k(\vec{x}_i, \vec{x}_j) \right) \right] \quad (17)$$

4. Given a new object, $\vec{z}$ the optimal α_i go into the decision function:

$$D(\vec{z}) = sign\left(\sum_{i=1}^{N} \alpha_i y_i k(\vec{x}_i, \vec{z}) + b \right) \quad (18)$$

When a mixture of several SVMs is used, it is improved the computation speed through parallelization, each of them has a weight and trained only on the part of the data set [71].

3.10. Hadoop Implementation with PSVM

The essential characteristics of the Hadoop framework are partitioning the data into thousands of machines and execute it in a parallel manner. The framework in Figure 3.7 shows the flow steps of the classification task of the PSVM implementation on the RapidMiner tool using Radoop extension on Hadoop cluster. The organization of these steps as follows:

- Upload dataset on HDFS in Hadoop cluster.
- Retrieve the data from HDFS.
- Preprocess the datasets
- Train the PSVM

- Classification result.

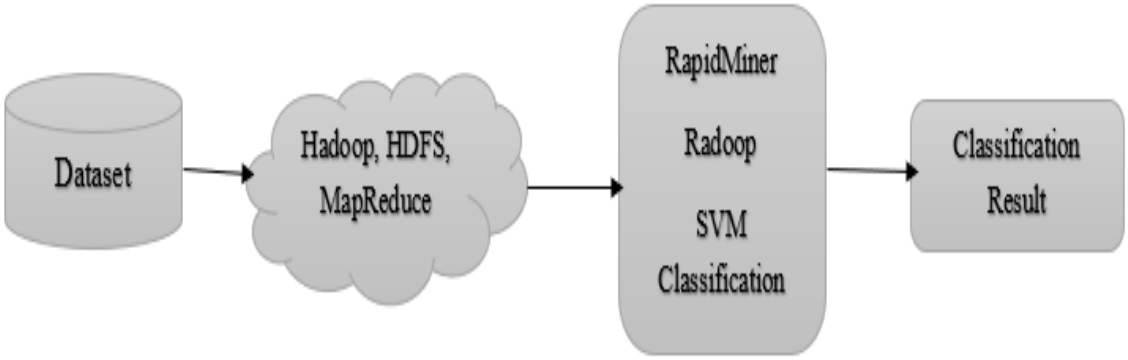


Figure 3.7. PSVM with Hadoop cluster using Radoop extinction.

3.11. Parallel SVM Based Classification

Big data environment has been simulated by importing the four data sets into RapidMiner [65]. RapidMiner is a data science software platform developed that supports all steps of the machine learning process, including data preparation, results in visualization model validation, and optimization. The dataset is distributed horizontally by making 4 CPU core nodes in the HPC cluster, and parallel execution is performed using PSVM as given in Figure 3.8. All the simulations are carried out using Intel Core i7, 2.90 GHz system, 8GB memory. The generated results are combined and represented.

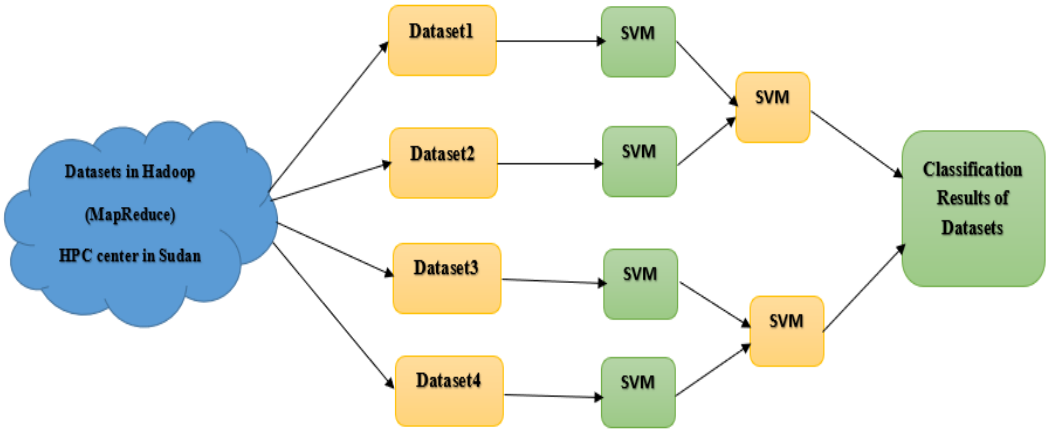


Figure 3.8: Parallel SVM Based classification algorithms.

3.12. MapReduce

MapReduce is a generalization model used to parallelize the processing of large-scale datasets. It uses two functions, namely a map and a reduce operation. The data is treated as a Key (k), Value (v) pair. A map operation takes a $\{k_1, v_1\}$ pairs and transmits an intermediate list of $\{k_2, v_2\}$ pairs. A reduce operation takes all values represented by the same key in the intermediate list and processes them accordingly, emitting a final new list [74]. The MapReduce framework is shown in Figure 3.9.

The input of map function is a set of key-value pairs, designated as k_1 and v_1, provided directly from the user-defined input files. Within the map function, the user specifies what to do with these keys and values. The map function outputs another set of keys and values, designated as k_2 and v_2. The reduce function sorts the key-value pairs by k_2. All of the associated values v_2 are reduced and emitted as value v_3.[74]. The map and reduce functions are as follows:

$$\text{Map } (k_1, v_1) \rightarrow [(k_2, v_2)] \quad (15)$$

$$\text{Reduce } (k_2, [v_2]) \rightarrow [\, v_3] \quad (16)$$

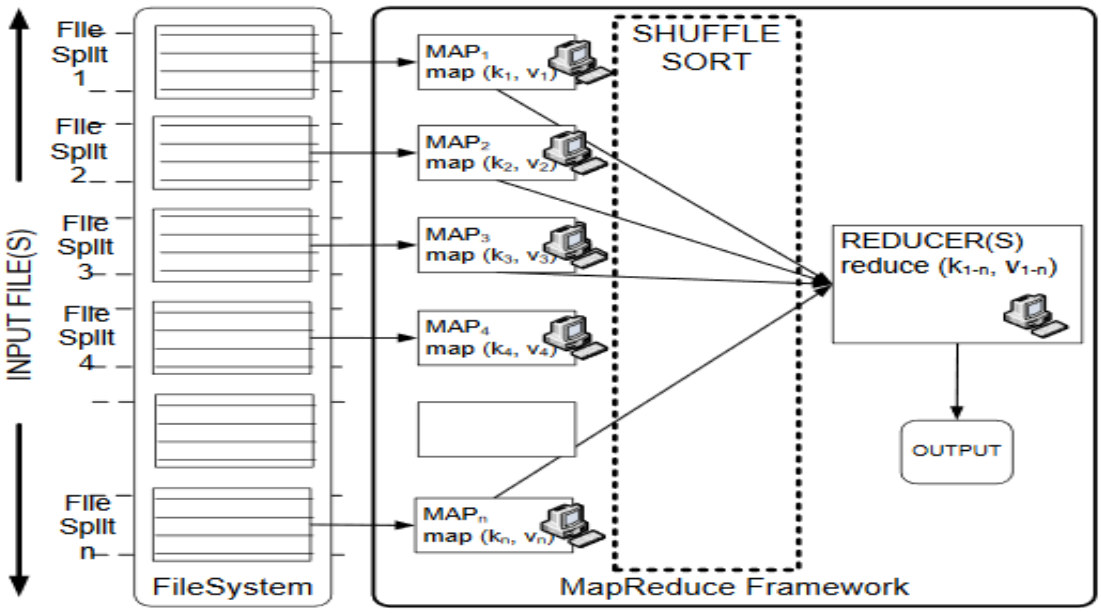


Figure 3.9: The MapReduce framework [74].

3.13. Implementation of PSVM with MapReduce in Hadoop

The architecture of Hadoop MapReduce programming model is shown in Figure 3.10. It shows how the input is divided into logical chunks and partitioned into various separate sets. These sets are then sorted and passed to the reducer. MapReduce model performs Mapper and Reducer interfaces to implement the map and reduce function [71].

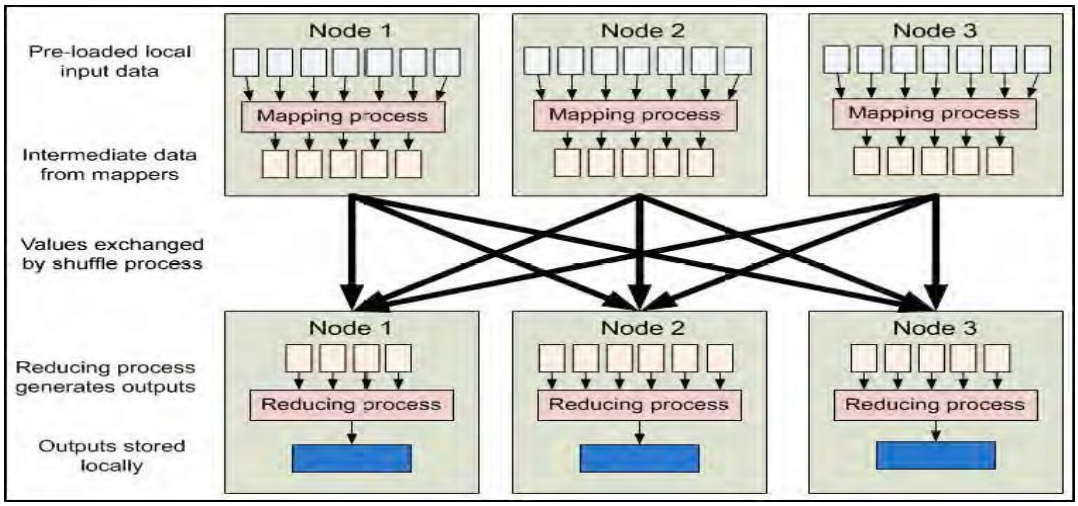


Figure 3.10: MapReduce Design on Hadoop [71].

Radoop extension in the RapidMiner tool is used. It is scaled with the size of the data set and the number of processing nodes. Four nodes have used in the cluster of HPC (High-Performance Computer in Sudan). The Hadoop and Hive table installed on it. All Radoop operators can access from the usual operator panel of RapidMiner, under the Radoop category. The process design under the RadoopNest operator, and some similar operators used on Hadoop, the data set is loaded into a hive table, then it can be retrieved from the distributed file system in the Hadoop [65].

During execution, the process usually starts MapReduce jobs that perform the desired operations on the data. The data residue on the cluster and Radoop only load references, metadata, and statistics about the table. It takes the same quantity of time to retrieve large and small tables. The result is returned to Radoop after the process of MapReduce is done, then the PSVM is applied,

the output data also are written to the distributed file system. The performance result shows that the using of PSVM on Hadoop cluster is enhanced the accuracy and reduce the computation time. In the comparison between SVM, K-means applied to SVM, and PSVM on Hadoop. We note that the Radoop is much faster, even with four processing nodes. The memory-based solutions might perform better on small data sets, but Hadoop has excellent scalability, and it suitable for a more complicated process [65].

3.14. Summary

This chapter has introduced the methodology design of the thesis. It explains the detailed description of data sets. The preprocessing methods to produce the final datasets is explained. The chapter then introduced the description and implementation of the machine learning algorithms.

First, the SVM is used to implement the four datasets, second, the k-means clustering algorithm applied to Support Vector Machines (SVM) is used as one model to implement the four datasets. Then, it is considered to find the most accurate classifiers among the combined and paralyze algorithms. Third, to simplify and improve the support vector machine accuracy, the PSVM on Hadoop cluster approach is used. The comparison of the results of three models had shown that the PSVM give the best performance.

CHAPTER FOUR
EXPERMENTAL RESULTS AND DISSCUSSION

4.1. First Experiment the Implementation of SVM

The four datasets that shown on table1 are implemented using SVM algorithm. This is done by using three types of kernels, the Radial kernel has the higher accuracy for all datasets. The parameters C, gamma, and epsilon Figure 4.1 and have fixed value which are, 0.0, 1.0, 0.01 respectively as shown in Table 4.2, Figure 4.2. The four-dataset shown in Table 4.1 are imported to RapidMiner platform as excel or csv files. Figure 4.3 shows the scatter before binary classification of adult dataset, while Figure 4.4 shows the scatter after binary classification of adult dataset. Figure 4.5 shows the correlation matrix of adult data set.

Table 4.1: Dataset description with dimensions and classes.

dataset	Associated Tasks	Instances	Dimensions	Classes
Water quality	Classification	888	20	2
adult	Classification	48842	14	2
Cover type	Classification	581012	54	5
Diabetes	Classification	100000	55	2

Table 4.2: The accuracy of four datasets with deferent kernel types.

Kernel type	(polynomial)	(dot)	(radial)	Computation Time
Water quality	69.33%	69.11%	69.79%	56m 30s
Diabetes	67.23%	57.01%	96.40%	4h,54m,60s
Adult	78.01%	76.66%	93.40%	1 h and 40m

cover type	69.90%	75.08%	74.52%	1 days and 22 h

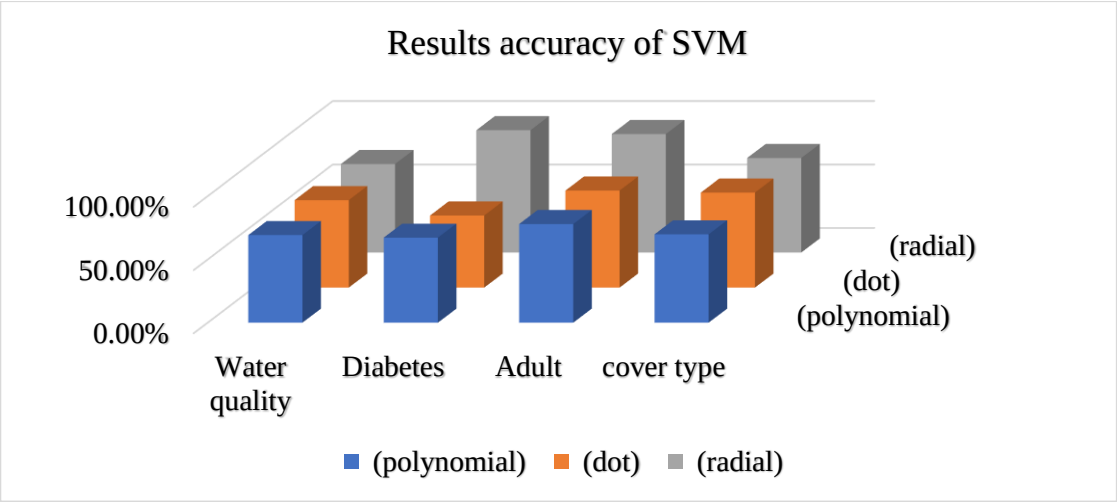


Figure 4.1: Results accuracy of SVM

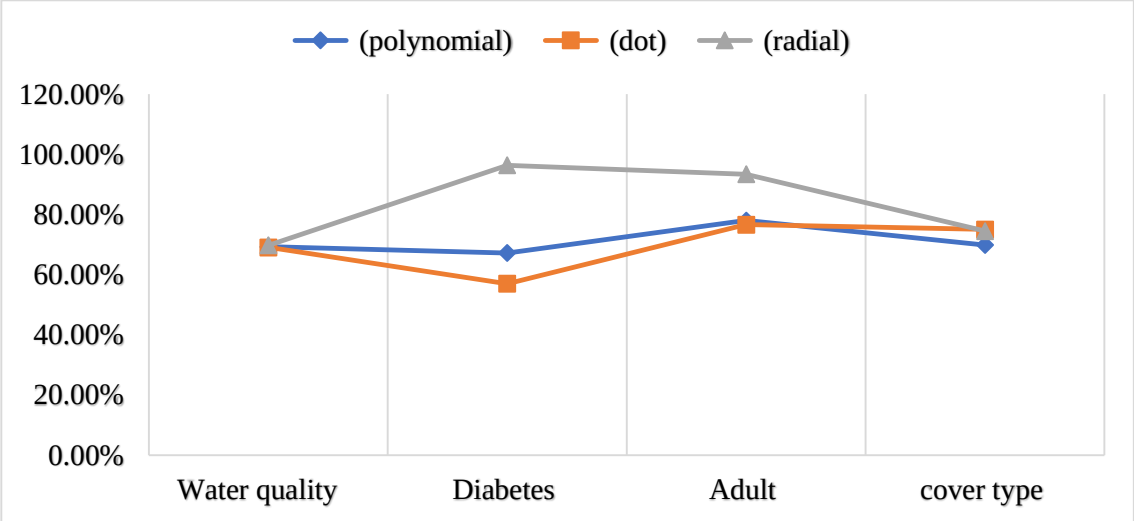


Figure 4.2: Results accuracy curve of SVM

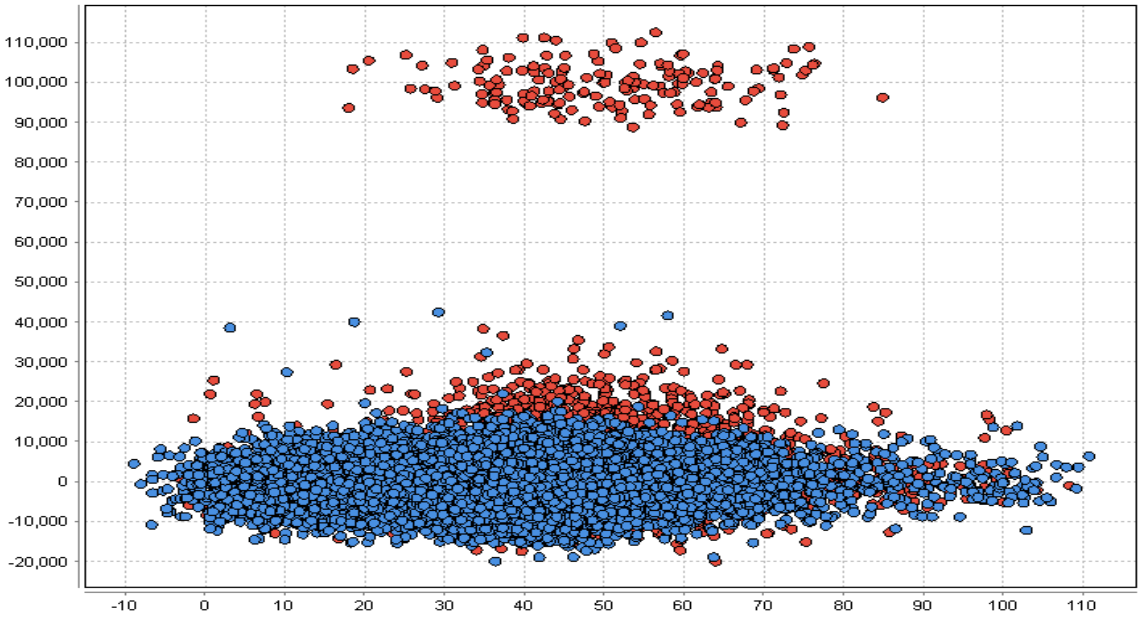


Figure 4.3: Adult data before classification.

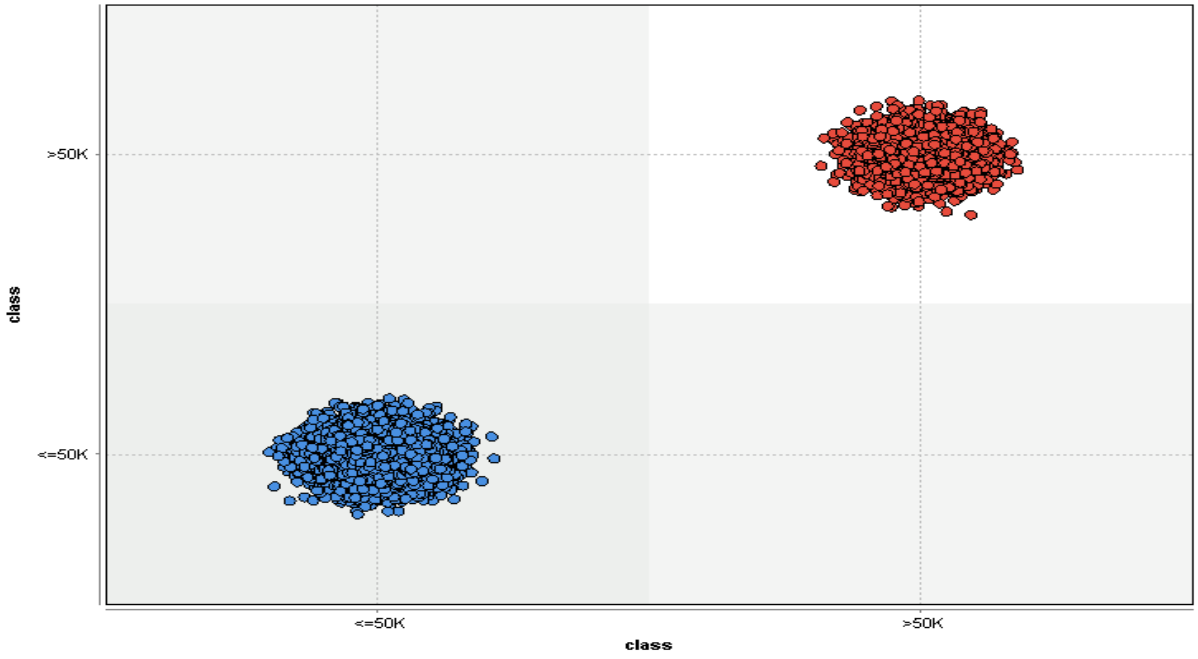


Figure 4.4: Adult data after classification

Attribut...	age	workcla...	educati...	educati...	marital-...	occupat...	relation...	race	sex	native-c...	class
age	1	0.194	0.038	-0.046	0.349	0.015	-0.200	0.135	0.063	-0.150	0.268
workclass	0.194	1	-0.021	0.042	-0.004	0.047	-0.003	-0.179	0.011	0.028	0.088
education	0.038	-0.021	1	-0.351	0.026	0.038	0.026	-0.091	-0.074	0.086	-0.045
educatio...	-0.046	0.042	-0.351	1	-0.089	-0.192	-0.091	0.019	0.078	-0.194	0.440
marital-s...	0.349	-0.004	0.026	-0.089	1	-0.000	0.002	0.103	0.284	-0.021	0.031
occupati...	0.015	0.047	0.038	-0.192	-0.000	1	-0.070	-0.016	-0.193	0.098	-0.110
relations...	-0.200	-0.003	0.026	-0.091	0.002	-0.070	1	0.142	0.267	0.239	-0.062
race	0.135	-0.179	-0.091	0.019	0.103	-0.016	0.142	1	0.146	0.002	-0.117
sex	0.063	0.011	-0.074	0.078	0.284	-0.193	0.267	0.146	1	-0.016	-0.080
native-co...	-0.150	0.028	0.086	-0.194	-0.021	0.098	0.239	0.002	-0.016	1	-0.065
class	0.268	0.088	-0.045	0.440	0.031	-0.110	-0.062	-0.117	-0.080	-0.065	1

Figure 4.5: Correlation matrix of adult data.

4.2. Second Experiment the implementation of SVM with Gamma=1.5

The SVM with a fixed value of gamma =1.5 and other different kernels applied to two water quality datasets (889 instances and 22parameters), the result in table3 shows that the accuracy is better in water quality dataset2 after the preprocessing is done to water quality dataset1. SVM also is trained to the adult dataset and diabetes dataset after cleaning. The diabetes dataset has been prepared to analyze factors related to readmission as well as other outcomes about patients with diabetes. With diabetes and adult datasets, the SVM algorithm was run with Radial kernel and gamma=1.5, C =0.0, convergence epsilon =0.1, the classification accuracy of diabetes is 96.40%, and with an adult dataset is 94.29%. The best efficiency for all datasets is when the gamma = 1.5 and the kernel is RBF. The result is shown in Table 4.3 and Figure 4.6 & Figure 4.7. Figure 4.8 shows the scatter plot before the classification water

dataset. Figure 4.9 shows the scatter plot after the classification water quality dataset.

Table 4.3: The accuracy of four datasets with deferent kernel types.

Dataset	Kernel gamma (γ =1.5) (radial)	Kernel type (polynomial) γ=1.0	Kernel type (dot) γ=1.0	Kernel type (radial) γ=1.0
Water quality datase1 C=0.0	69.90%	69.33%	69.11%	69.79%
Water quality dataset2 C=1.0	95.60%	69.90%	90.76%	95.15%
Adult dataset C=0.0	94.29%	78.01%	76.66%	93.40%
Diabetes C=0.0	96.40%	93.04%	57.01%	94%

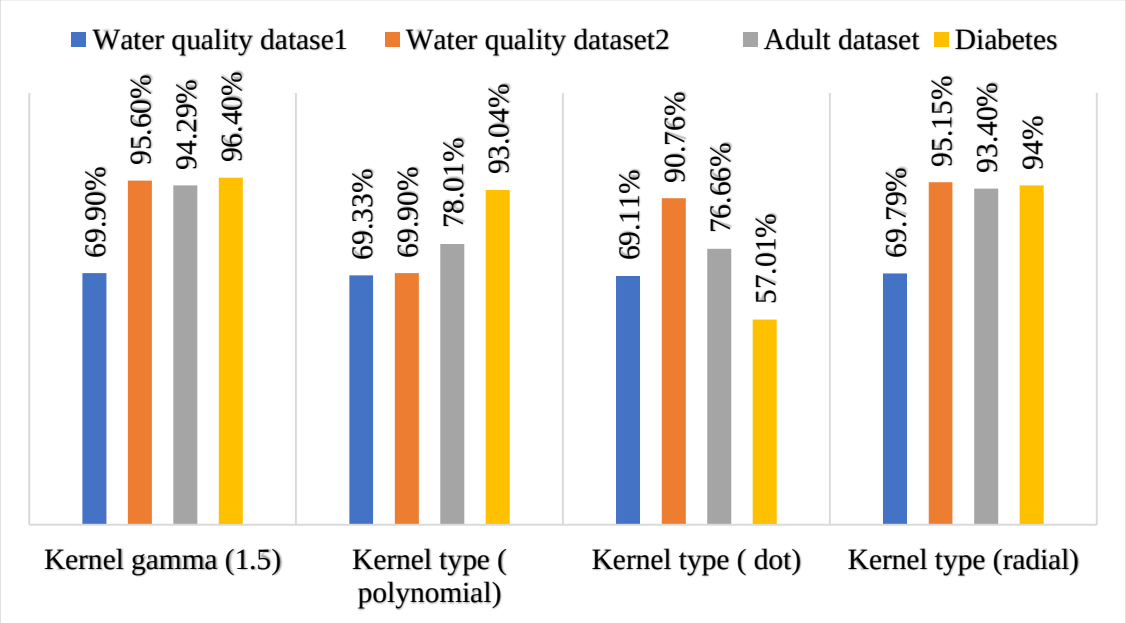


Figure 4.6: Results accuracy of SVM with Gamma=1.5.

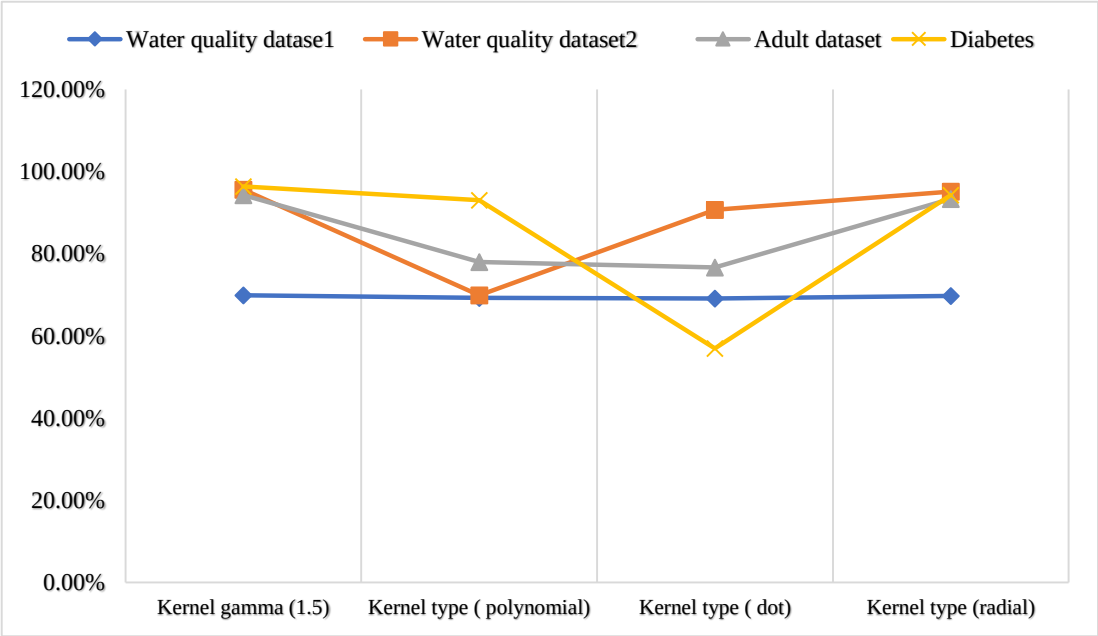


Figure 4.7: Accuracy curve of SVM with Gamma=1.5.

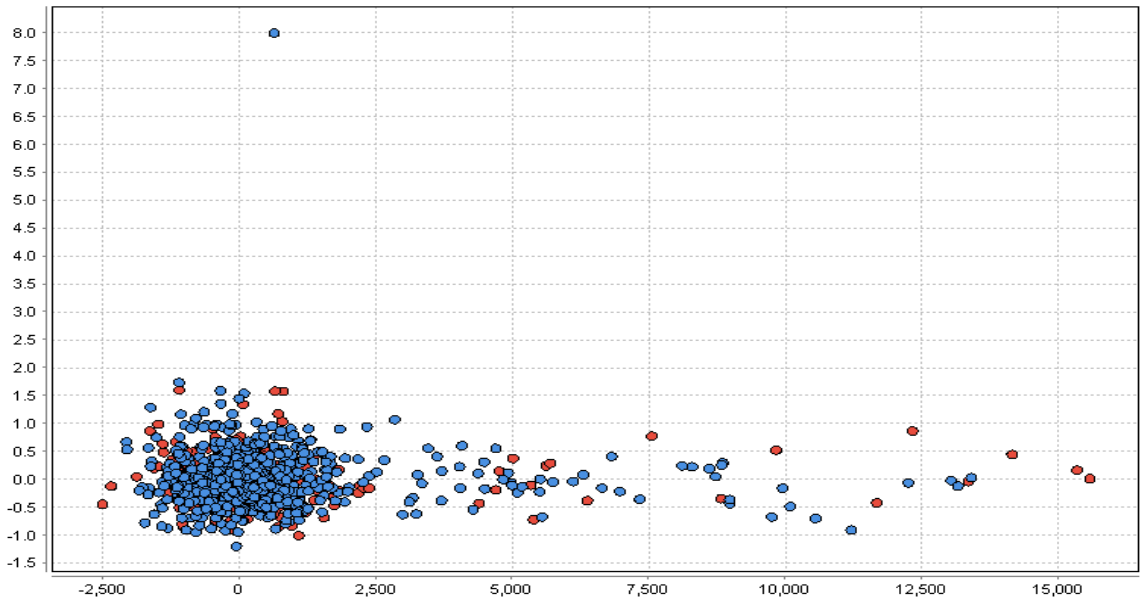


Figure4.8: scatter plot before the classification water dataset.

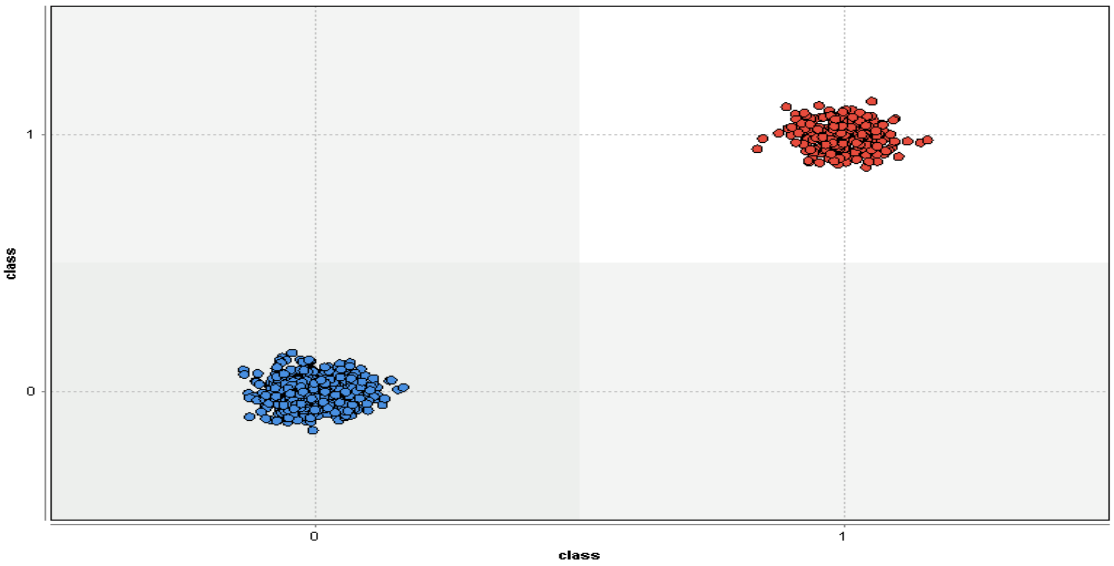


Figure 4.9: scatter plot after the classification water quality dataset.

4.3. Third Experiment the implementation of k-means clustering applied to SVM

The k-means clustering algorithm is applied to SVM algorithm. First the k-means implements with max run =10, and parameter k=2, this is shown in Figure 4.10. The results are shown in Table 4.4 and Figure 4.11. Figure 4.12: (a) and (b) shows the cover type dataset before classification by regression and after preprocessing respectively, while Figure 4.13 shows the scatter after classification cover type data. Confusion Matrix of the Classification by regression of cover type dataset is shown in Figure 4.14.

Figure 4.10: K-mean parameters.

Table 4.4: Results accuracy of SVM with k-mean (k=2)

Kernel	Dot	Polynomial	Radial	Value of k	Computation Time (in sec)
Water quality	70.45%	69.79%	74.52%	2	73.48
Diabetes	81.12%	76.61%	97.00%	2	430.12
adult	84.45%	87.12%	96.40%	2	980.18
Cover type	86.77%	87.12%	91.17%	2	1444.58

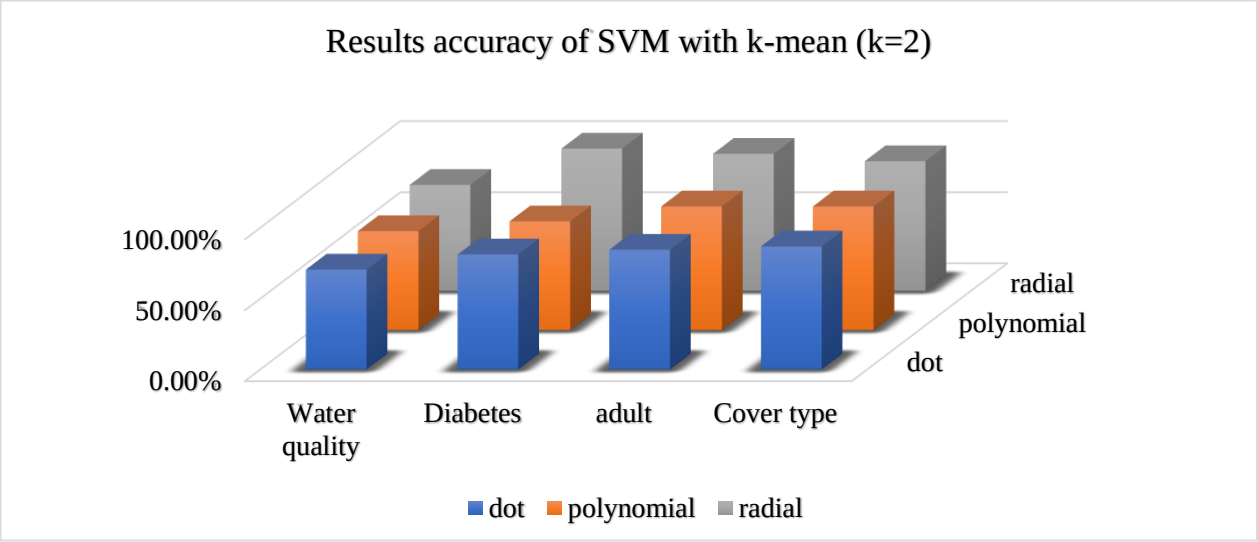


Figure4.11: Results accuracy of SVM with the k-mean (k=2).

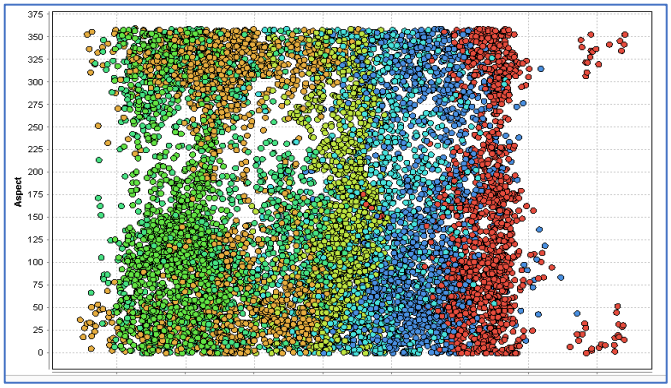


Fig12:(a)

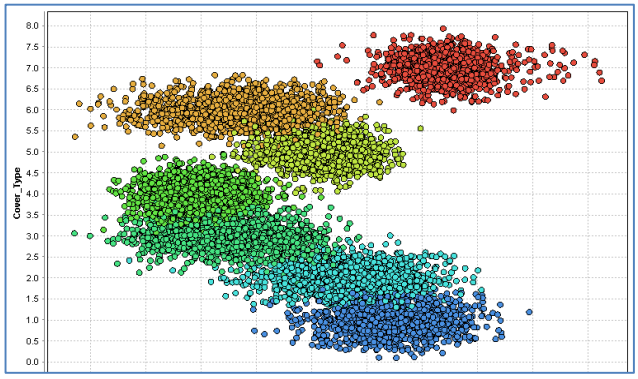


Fig12:(b)

Figure 4.12: (a) Cover type dataset before classification. (b) After preprocessing

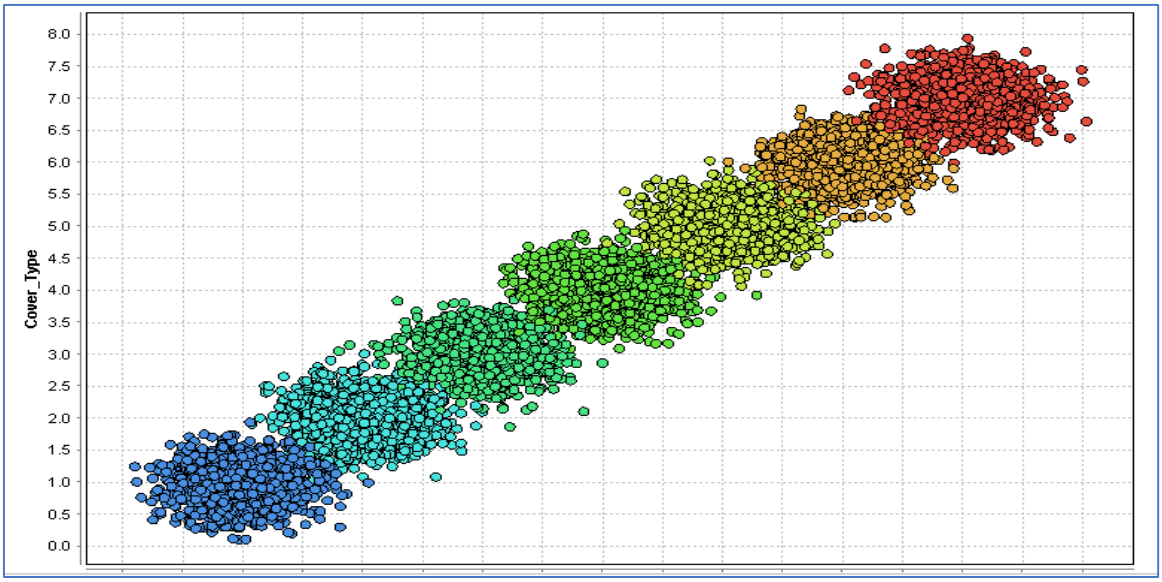


Figure 4.13: Cover type dataset after classification.

accuracy: 95.26%

	true 5	true 2	true 1	true 7	true 3	true 6	true 4	class precision
pred. 5	1558	56	13	0	2	4	0	95.41%
pred. 2	6	1349	40	0	0	0	0	96.70%
pred. 1	1	34	1301	6	0	0	0	96.94%
pred. 7	0	2	20	1343	0	0	0	98.39%
pred. 3	10	9	0	0	1106	44	10	93.81%
pred. 6	8	11	1	0	87	1278	19	91.03%
pred. 4	0	1	0	0	67	23	1591	94.59%
class recall	98.42%	92.27%	94.62%	99.56%	87.64%	94.74%	98.21%	

Figure 4.14: Classification by regression. Confusion Matrix of cover type dataset.

4.4. Result Comparison of SVM and k-mean applied to SVM

- Using single SVM model, it is found that when the number of instances increases, the training time also increases leading to weak performance. So, k-mean clustering is used to reduce the number of these instances producing better accuracy.
- The RBF shows the best result over the other two kernels with best accuracy. For this reason, the compression between the two models is done in term of the RBF. Table4.5, Figure 4.15, Figure 4.16 show the results of this compression.

Table4.5: Comparison of result of both models.

datasets	Water quality	Adult	Diabetes	Cover type
SVM	69.79%	94.29%	96.40%	74.52%
SVM with k-mean kernel (k=2)	74.52%	96.40%	97.00%	91.17%

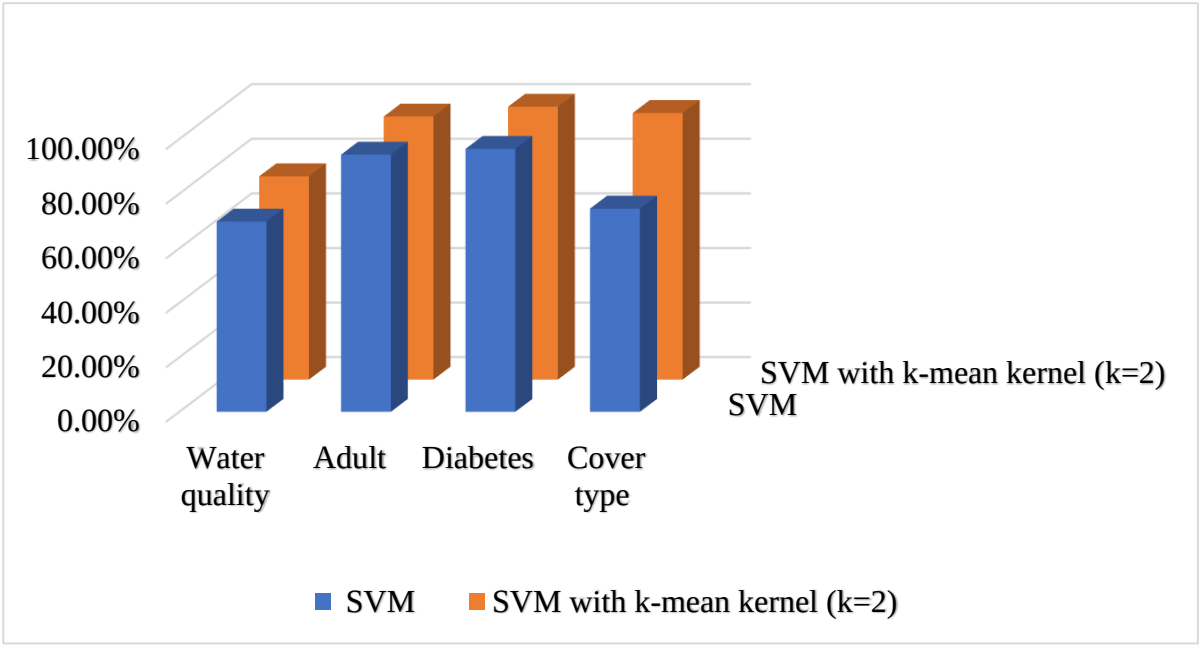


Figure 4.15: Results accuracy of SVM and SVM with k-mean kernel (k=2)

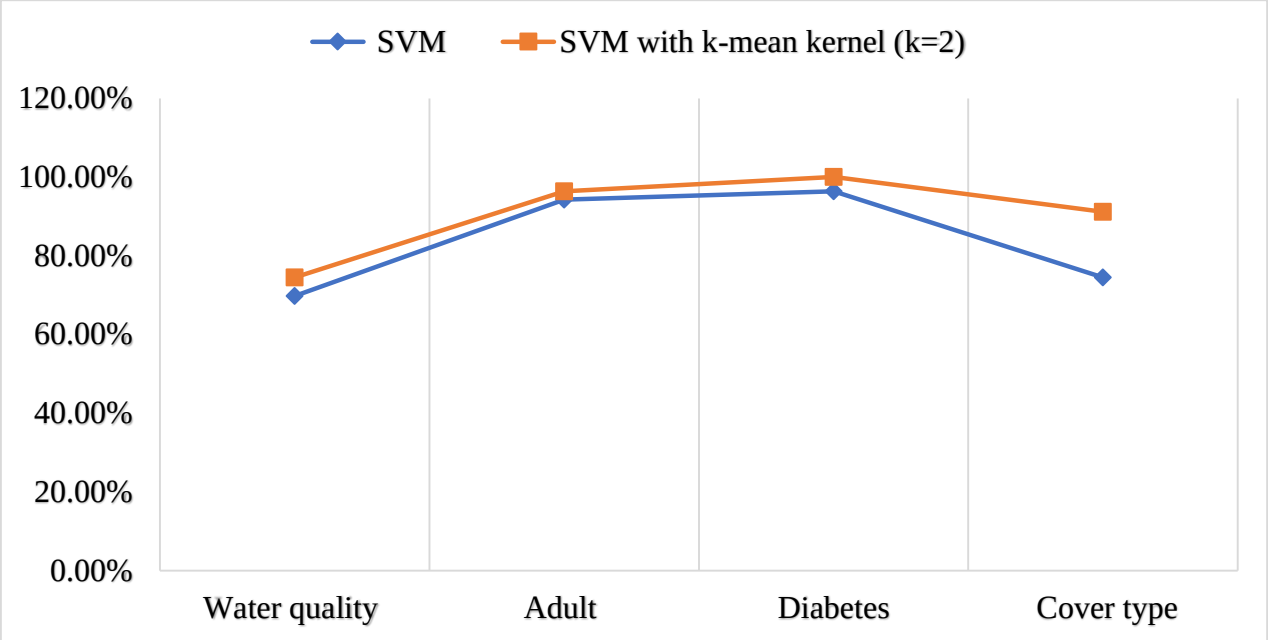


Figure 4.16: Curve of SVM and SVM with k-mean kernel (k=2)

Figure 4.17 and Figure 4.18 show the statistical analysis of the two classes of an adult dataset and Diabetes dataset after the classification process using PSVM and data cleaning and filter of attributes.

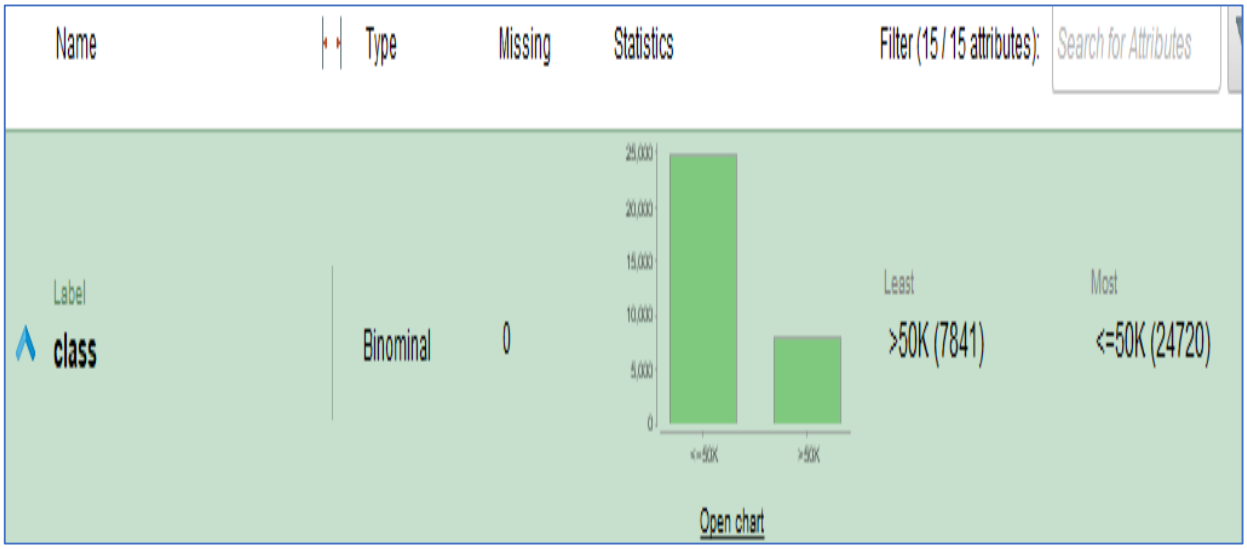


Figure 4.17: Statistical analysis of the two classes. After classification of adult dataset

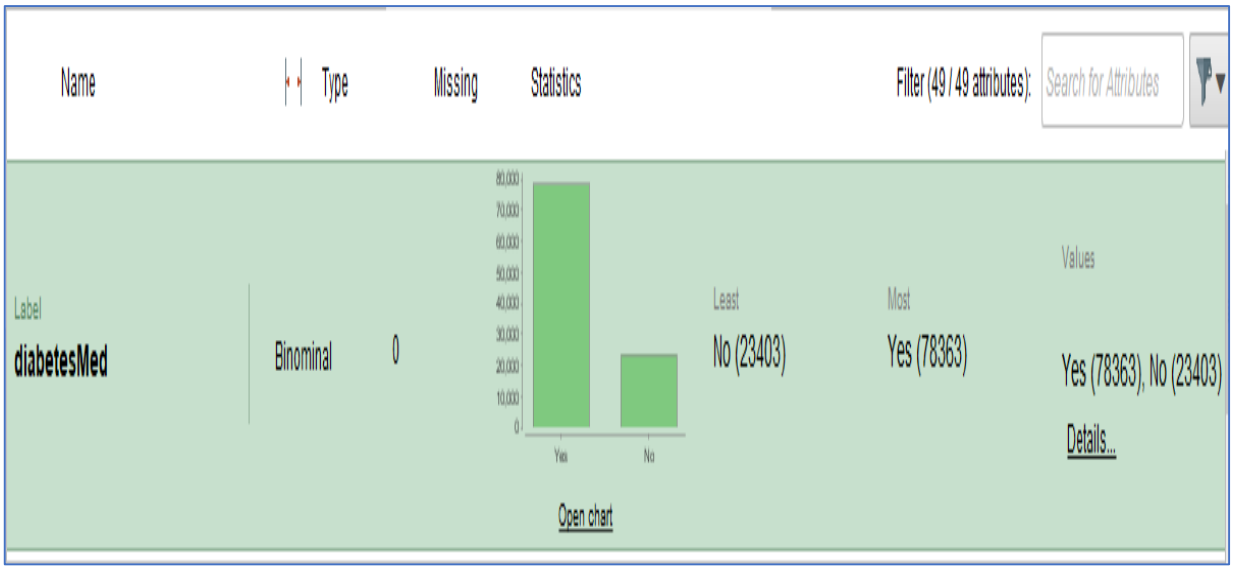


Figure 4.18: Statistical analysis of the two classes. After classification of Diabetes dataset.

The most critical issue in big data processing is the execution time that the algorithm will take. Table4.6 and Figure 4.19 show the comparison of the execution time of SVM and k-means applied to SVM. It shows that when the k-means applied to SVM, it results in a better speed up, although if the data size grows.

Table4.6: Compare Execution Time for SVM algorithms and k-means applied to SVM

Execution Time		**Data Size**			
		888	48842	100000	581012
Algorithms	**SVM**	56m 30s	4h,54m,61s	1 h and 40m	1 days and 22 h
	k-means with SVM	73.48s	630.12s	980.18s	1444.58s

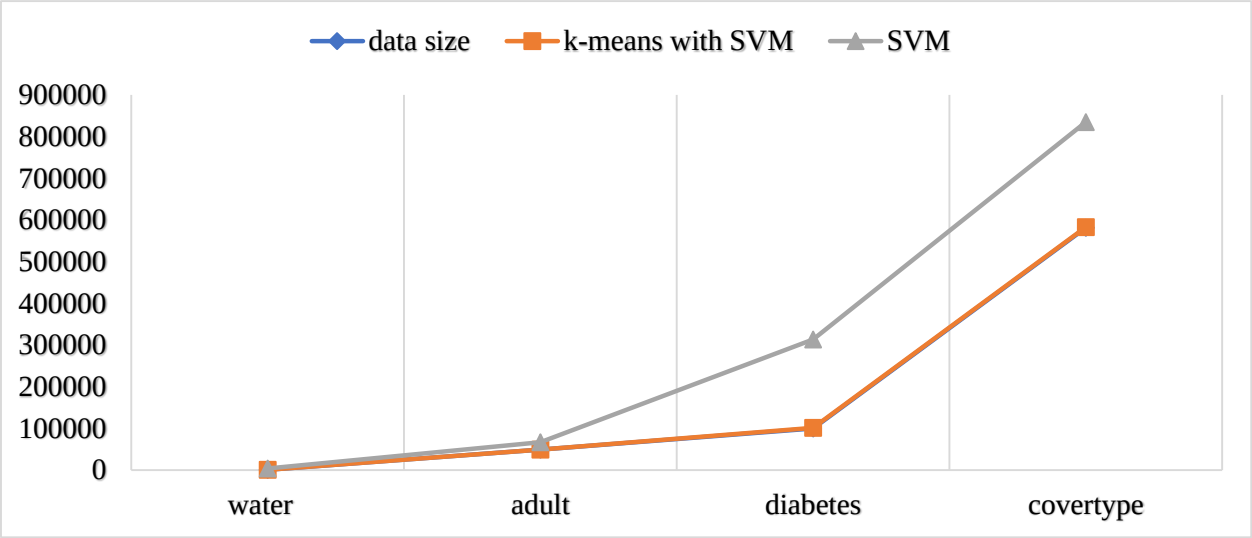


Figure 4.19: Comparison of execution Time for SVM and k-means applied to SVM.

4.5. Forth Experiment of PSVM on Hadoop cluster

The experiments were generated using the Hadoop framework due to the fact that one its main component MapReduce has the same characteristics as the PSVM approach. This software has been setup in HPC in Sudan using. Hadoop cluster hardware and software configuration are shown in Table 4.7. The Hadoop configuration is done in RapidMiner using mange Radoop connection setting as shown in Figure 4.20 (a, b).

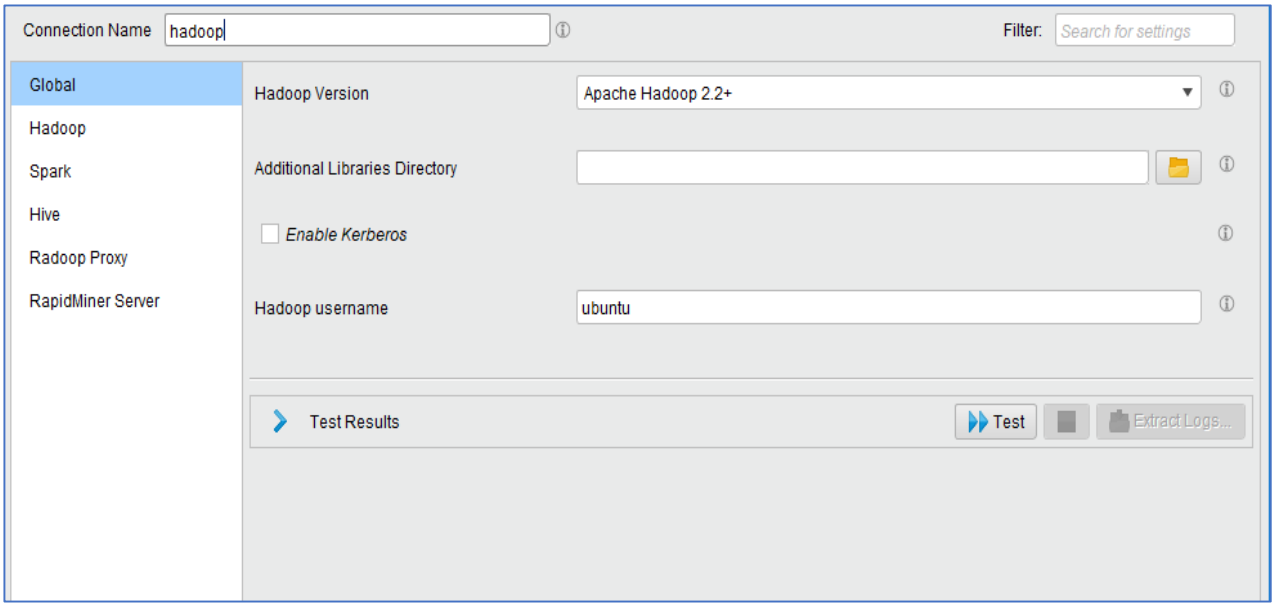


Figure 4.20: (a).The configration of the Hadoop in RapidMine.

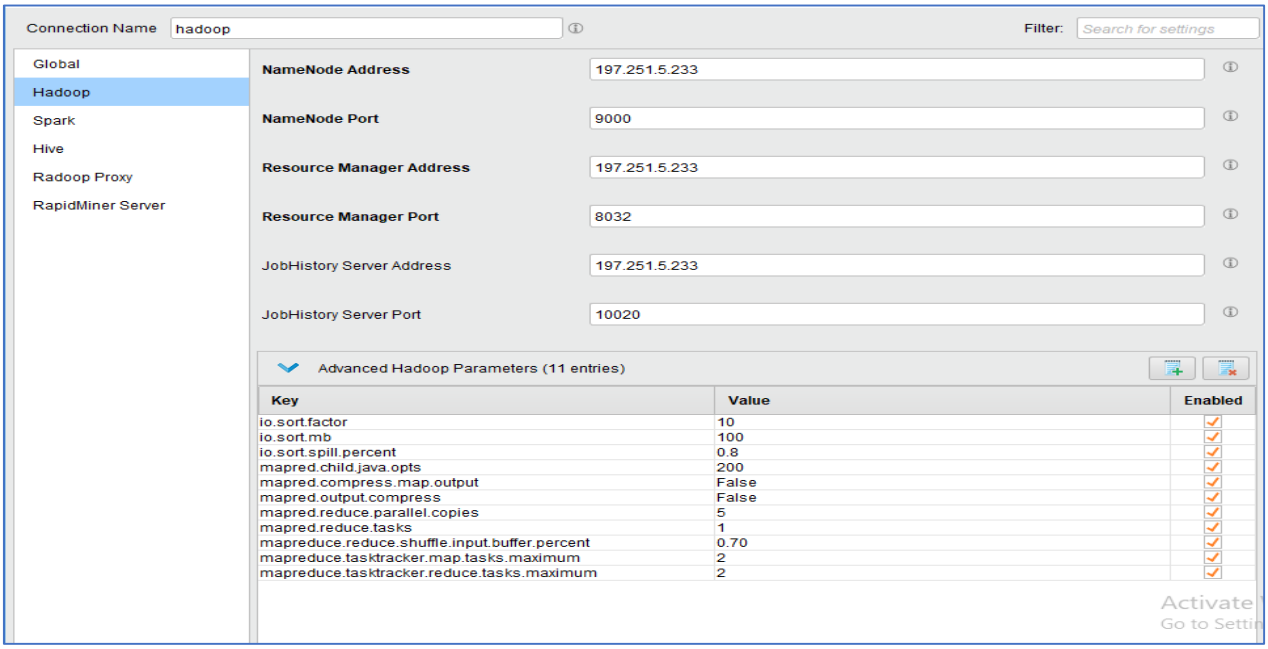


Figure 4.20:(b).The configration of the Hadoop in RapidMiner.

4.6. The Efficiency of the Parallel SVM

The SVM algorithm was re-modeled for testing on a MapReduce Hadoop cluster. It provided in RapidMiner to paralyzed, configured as a MapReduce job. The efficiency of the Parallel SVM is done as follows: The SVM algorithm was re-modeled for testing on a MapReduce Hadoop cluster. It is provided in RapidMiner paralyzed and configured as a MapReduce job. The configuration of the Hadoop cluster with the resources (software and hardware) is shown in Table 4.7.

Table4.7: Hadoop cluster configuration with resources.

Hardware Environment		
	CPU	**RAM**
Node 1, 2, 3, & 4	Intel Core i5	8 GB
Client	Intel Core i7(64-bit OS)	8 GB
Software Environment		
SVM	RapidMiner 9.4	
OS	Ubuntu 16.04	
Hadoop	Apache Hadoop 2.2+	

The experiment is carried out by taking RBF kernel function, penalty parameter C =1 and gamma = 0.01. Experiment is carried out by 4 nodes on Hadoop cluster in HPC center in Sudan. The results of four datasets are shown in Table 4.8 and Figure 4.21.

Table 4.8: Hadoop Cluster Results

Dataset	Accuracy	Computation Time (in sec)	Number of nodes
Water quality	90.05%	15.05	4
adult	96%	530.74	4
Diabetes	97.47%	630.74	4
Cover type	97%	690.91	4

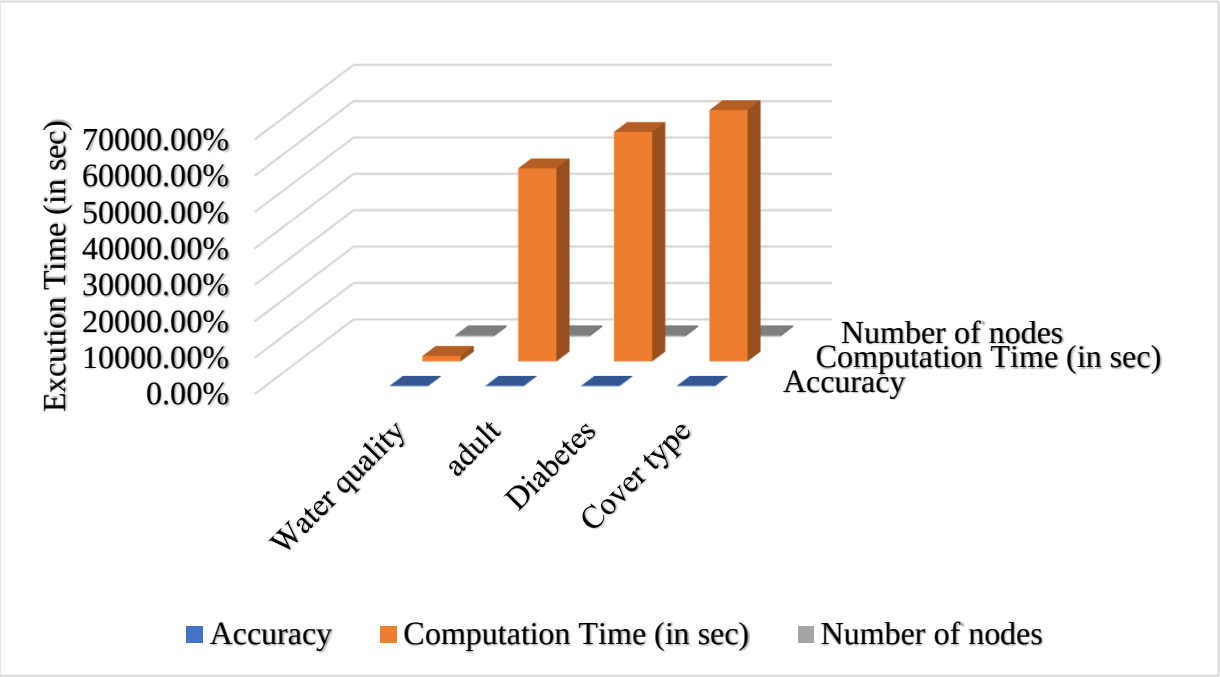


Figure 4.21: Result accuracy of PSVM.

4.7. First Experiment versus forth Experiment

- By using single SVM model, it is found that when the number of instances increases, the training time also increases leading to weak

performance. So, PSVM is used to reduce and spilt the number of these instances producing better accuracy and time consuming.

- By using sequential SVM is challenging and difficult to work with large scale data set. The parallel SVM works efficiently on large datasets as compared to the sequential SVM. The advantage of using PSVM is a distributed model that spilt the data and excite it in parallel. The number of nodes that are used on the Hadoop cluster is four; it is worked in a parallel manner. Table 4.9, Figure22, and Figure23 shows the accuracy result.

Table 4.9: Accuracy comparison of SVM and PSVM.

Datasets	**Water quality**	**Adult**	**Diabetes**	**Cover type**
SVM (using RBF)	69.79%	94.29%	96.40%	74.52%
PSVM (using RBF)	90.05%	96.80%	97.47%	95.47%

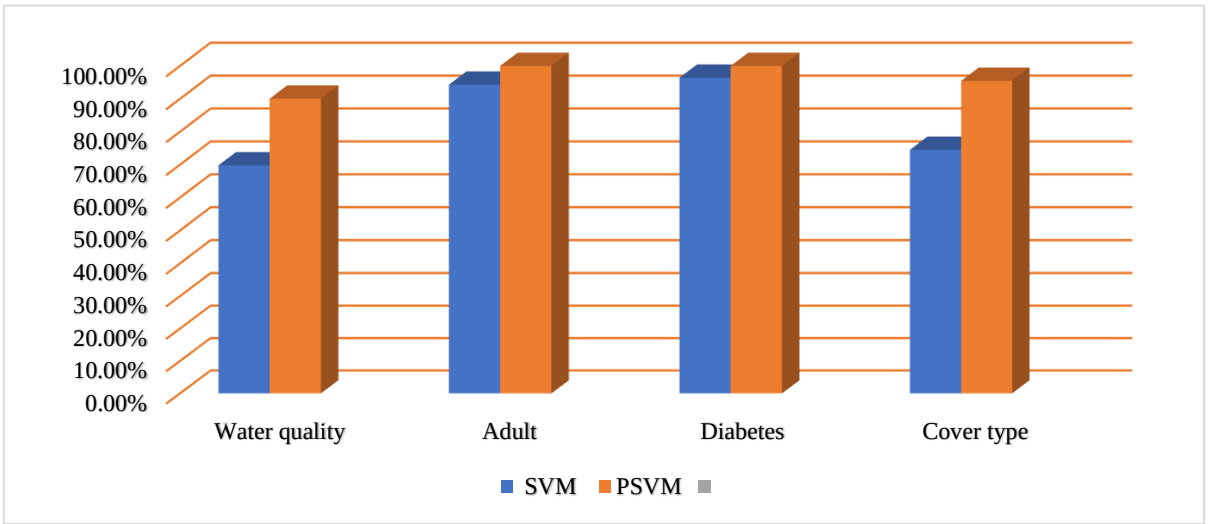


Figure 4.22**:** Results comparison accuracy of SVM and PSVM

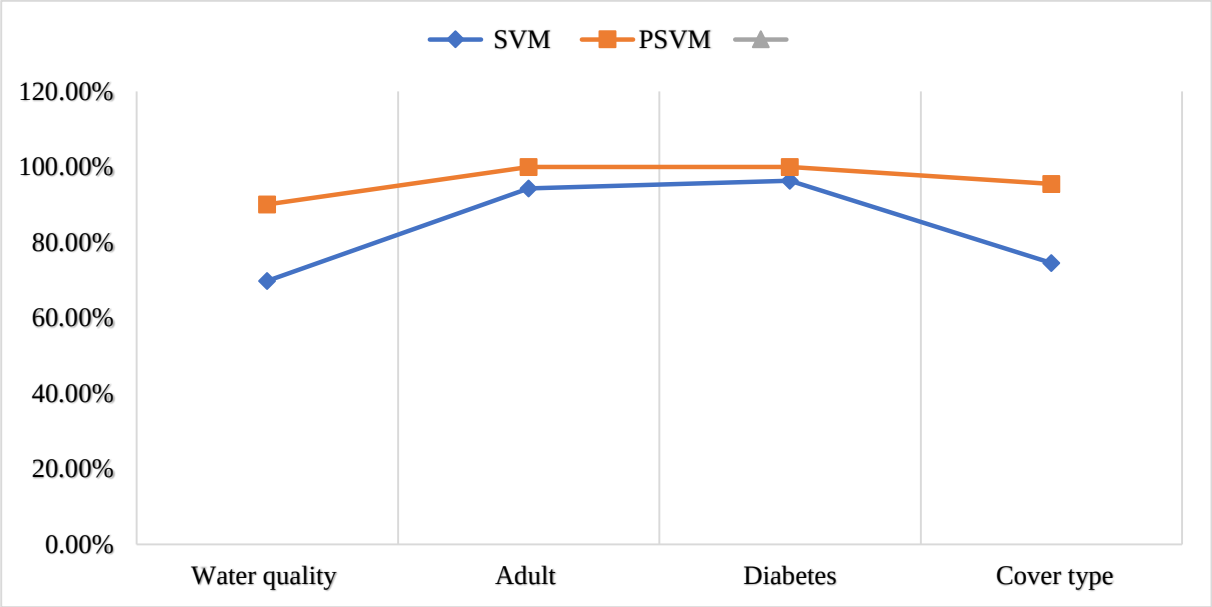


Figure 4.23: Accuracy curve comparison of SVM and PSVM.

Table 4.10 and Figure24 are predicted that the processing time of SVM increases exponentially as data size increases. On the other hand, the processing time of PSVM is incomparably low concerning that of SVM. Hence, to improve the processing speed of SVM applied to big data for classification, the parallel SVM is used.

Table 4.10: Comparison of Execution Time for SVM and PSVM algorithms

Execution Time in (Sec)		**Data size (in bytes)**			
		888(116 KB)	48842(2.32 MB)	100000(18.2 MB)	581012(109 MB)
Algorithm	**SVM**	3000.39	17000.7	6000.001	165000.6
	PSVM	15.05	530.74	630.74	690.91

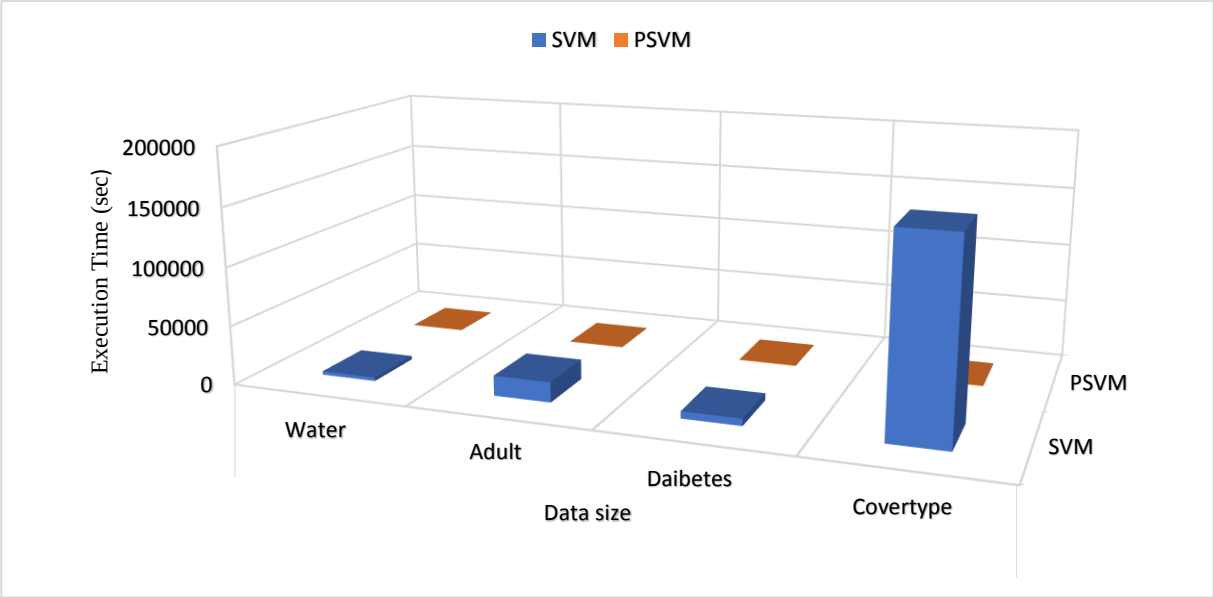


Figure 4.24: Execution time of SVM and PSVM on varying sizes of dataset.

4.8. Result comparison for, First, Second, Forth Experiments

- By using sequential SVM is challenging and difficult to work with large scale data set. The MapReduce based parallel SVM works efficiently on large datasets as compared to the sequential SVM. The advantage of using MapReduce based SVM is the core components of the Hadoop framework HDFS and MapReduce distributed programming model provides data awareness between the NameNode and DataNode. The number of nodes that are used on the Hadoop cluster is four, it is worked in parallel manner.
- Using single SVM model, it is found that when the number of instances increases, the training time also increases leading to weak performance. So, k-mean clustering is used to reduce the number of these instances producing better accuracy.

- The RBF shows the best result over the other two kernels with best accuracy. For this reason, the compression between the two models is done in term of the RBF. Table 4.11, Figure25, Figure26 show the results of this compression.

Table 4.11: Comparison of SVM, SVM with k-mean, and PSVM.

Datasets	**Water quality accuracy**	**Adult accuracy**	**Diabetes accuracy**	**Cover type accuracy**
SVM	69.79%	94.29%	96.40%	74.52%
SVM with k-mean	74.52%	96.40%	97.00%	91.17%
PSVM	90.05%	96.80%	97.47%	95.47%

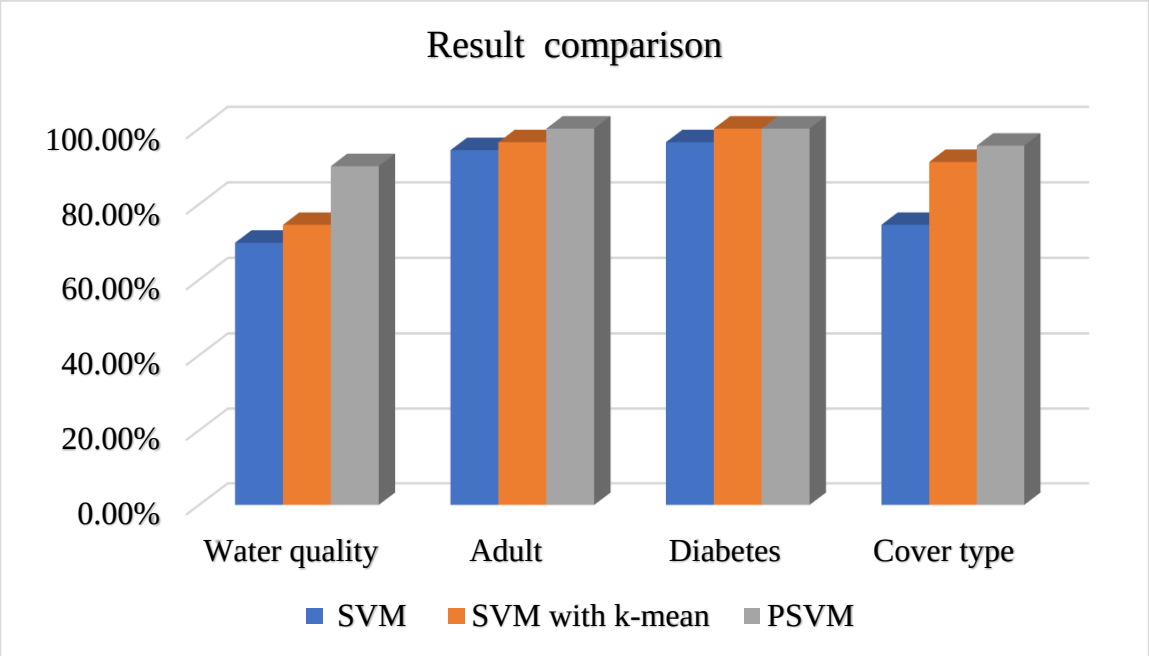


Figure 4.25: Comparison of SVM, SVM with k-mean, PSVM.

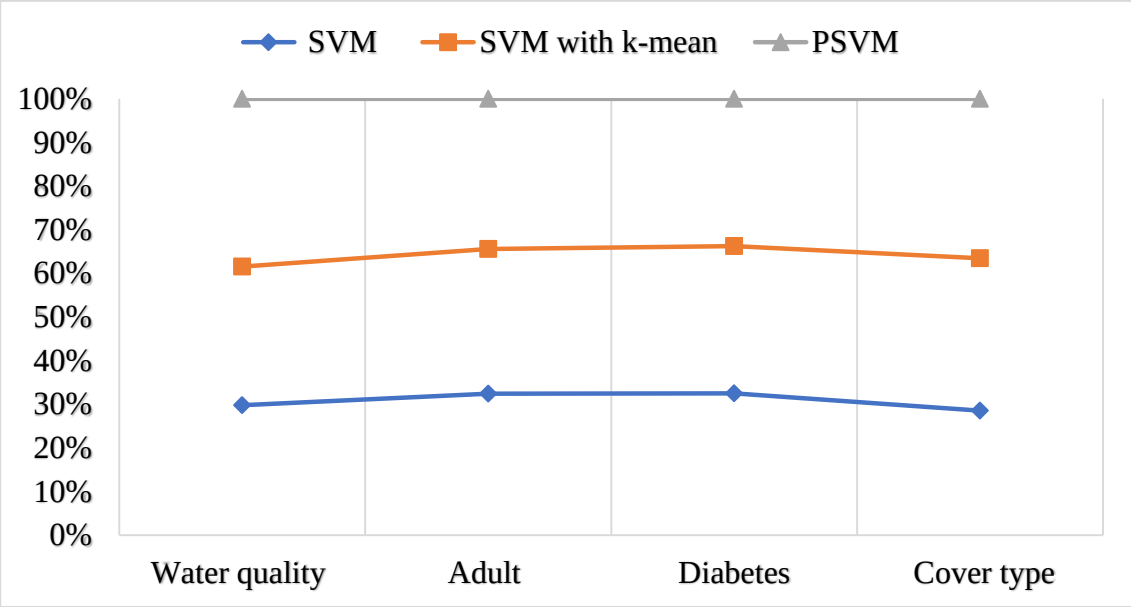


Figure 4.26: Curve of SVM, SVM with k-mean, PSVM.

Table 4.12 and Figure 4.27. Predict that the processing time of k- means with SVM increases exponentially as data size increases. On the other hand, the processing time of PSVM is incomparably low concerning that of k- means with SVM. Hence, to improve the processing speed of SVM applied to big data for classification, the parallel SVM is used. Table 4.12, Figure 4.28, and Figure 4.29 :(a, b). Show the comparison of four datasets classification accuracy using the two algorithms.

Table 4.12: Comparison of Execution Time for k-means applied to SVM and PSVM algorithms

Execution Time (Sec)		**Data Size**			
		888	48842	100000	581012
Algorithm	**k-means applied to SVM**	73.48	630.12	980.18	1444.58
	PSVM	15.05	530.74	630.74	690.91

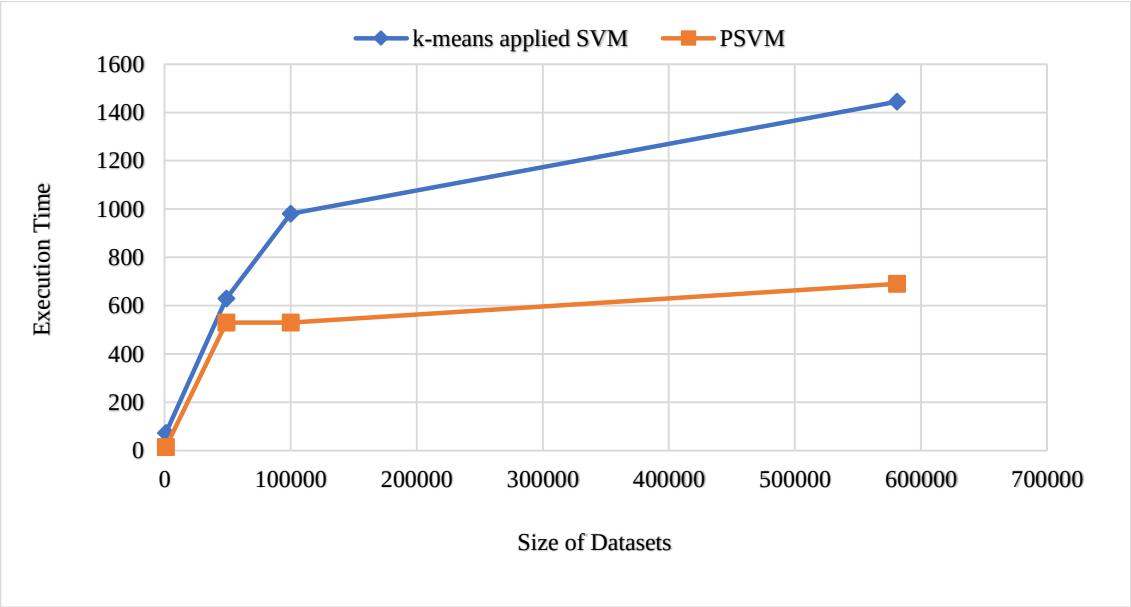


Figure 4.27: Execution time of SVM and PSVM on varying sizes of dataset.

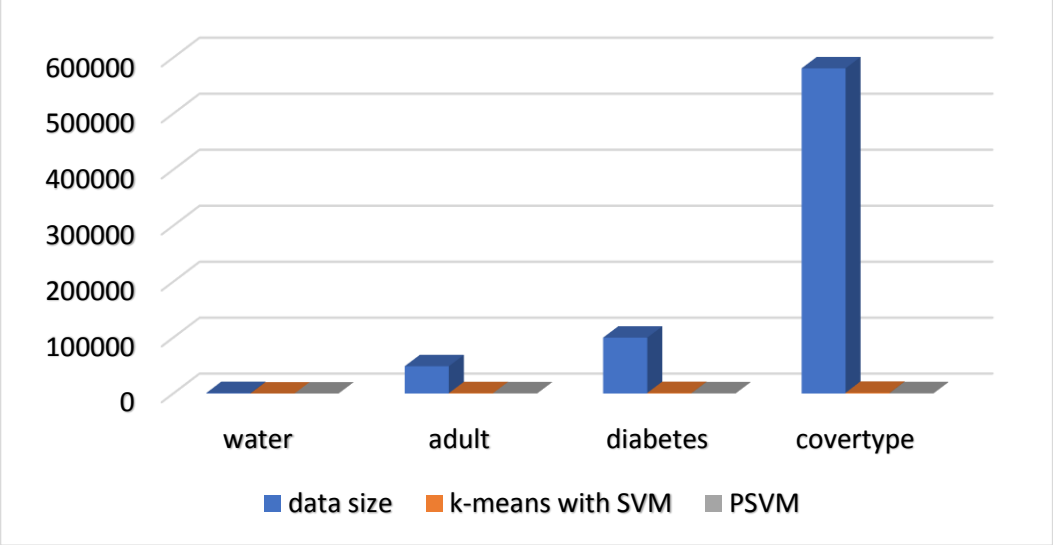
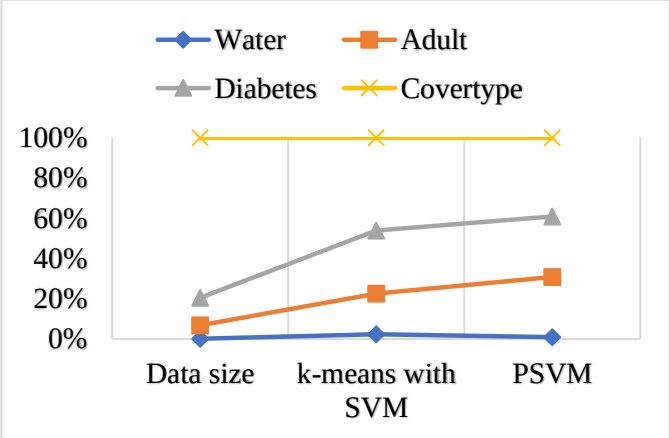


Figure 4.28: Comparison of execution time for K-means applied to SVM and PSVM with data sizes.



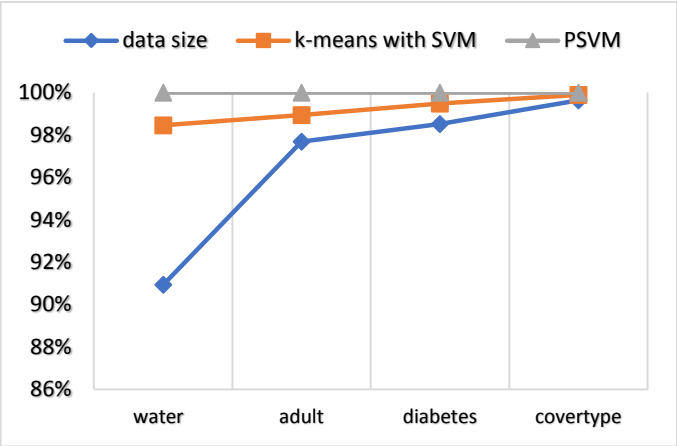


Figure29. (a)

Figure 4.29 :(a, b). Comparison curve of execution time for K-means applied to SVM and PSVM.

Figure 29. (b)

4.9 Chapter summary

In this chapter presented all the results of the classifications models built by single algorithms and parallel support vector machine used in the experiments. Also, it provides a comparison between all classification models results to validate the PSVM model. It explains the use of Support Vector Machines (SVM) and k-mean clustering applied to the SVM framework. Parallel support vector machine for big data classification in which big data technology has played an important role in classification, which gives more accuracy. The accuracy results show that the SVM classifier is a perfect classifier for different data sets, although if it is big data, but when we paralyzed the hyperplane of SVM.

CHAPTER FIVE

CONCLUSION AND FUTURE WORK

Big data faces many problems when dealing with machine learning techniques. SVM is a robust classifier with a high computational cost. The computational cost increases with an increase in data size. Parallel computation is found to be ideal with big data analysis, for it divides the data into smaller partitions to increase the efficiency and accuracy and reduce the computation time.

This thesis introduced the parallel support vector machine algorithm and the importance of using the PSVM in the implementation of big data. Many of the traditional support vector machine and different methods are described. Next, the k-means clustering algorithm and its applications in big data fields are introduced. The definition, implementation, and architecture of Hadoop framework resources are discussed. Furthermore, the complexity of dealing and implementing large data sets and the resources or environment associated with Hadoop are determined and identified. Then, a description of data mining techniques used to implement four data sets, each of the three algorithms used for the classification tasks. Finally, results are provided.

Conclusions

Researchers have a different opinion about big data processing and analysis. Various algorithms and tools provide an excellent framework to implement the PSVM. The work presented in this thesis aims to increase the performance and perform time-consuming. This research had conducted many steps:

- The first step was data collection. Four data sets are involved in this research, three of them from the UCI repository. The fourth one is real data from the ministry of health in Sudan.

- The second step is the preprocessing of data. Missing values were replaced, filtering and changing the data type of the attributes were made for all datasets.
- The third step is the designing of three models were used in this research, and their results are compared.
- The first model was an SVM model applied to the data set for binary classification.
- The second model is first the datasets are partitioning using the k-means clustering algorithm, then it is applied to the SVM algorithm.
- The third model is PSVM based on Hadoop MapReduce. It is implementing in HPC center in Sudan.
- The results of the three models were compared, the compression was done through four steps:
    1. Single SVM is compared to k-means clustering applied to SVM, it found that the k-means clustering applied to SVM enhanced the performance classification results.
    2. The k-means clustering applied to SVM compared to PSVM based on Hadoop MapReduce model, it found that the PSVMs model gives better accuracy and reduce computation time of the big data set during the execution process.
    3. The PSVM based on Hadoop MapReduce model compared with traditional SVM, it is found that the using of PSVM model reduce the number of instance and computational time.
    4. The three models are compared, it is found that the PSVM based on Hadoop MapReduce model was enhance the performance specially it reduces the execution time and give higher accuracy.

The publication papers

The research done during the PH.D. studies has been presented in the following papers:

1- *Abd Elkarim, Iatimad Satti, and Johnson Agbinya. "**A Review of Parallel Support Vector Machines (PSVMs) for Big Data classification**." Australian Journal of Basic and Applied Sciences 13.12 (2019): 61-71.*

2- *Abd Elkarim, Iatimad Satti, and Atika Hussein Johnson Agbinya. "**Parallel SVM Based Classification Technique on big data: HPC center in Sudan**." Australian Journal of Basic and Applied Sciences (2020).*

Future work

As future work, there is needed to develop an ensemble model from the three implemented models. Also, it is recommended to use parallel SVM model with other algorithms and compare its results to models in other recent research papers. Also, the PSVM model can be extended for multi-classification by utilizing parallelism related to both CPUs and PSVM. So, this approach can be more beneficial with more complex datasets.

REFRENCES

1- Nandakumar, A. N., & Yambem, N. (2014). A survey on data mining algorithms on apache hadoop platform. *International*

Journal of Emerging Technology and Advanced Engineering, *4*(1), 563-565

2- The basic Big Data definition *Peter Kinnaird; Inbal Talgam-Cohen, eds. (2012). "Big Data". XRDS: Crossroads, The ACM Magazine for Students. Vol. 19 no. 1.* https://en.wikipedia.org/wiki/Big_data. 9. *ISSN 1528-4980. OCLC 779657714.*

3- Priyadarshini, A. (2015). A map reduces based support vector machine for big data classification. *International Journal of Database Theory and Application*, *8*(5), 77-98.

4- Sun, Z., & Fox, G. (2012). Study on parallel SVM based on MapReduce. In *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA)* (p. 1). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).

5- Sagiroglu, S., & Sinanc, D. Big Data: A Review Collaboration Technologies and Systems (CTS). In *2013 International Conference on big Data*.

6- Li, J., Xu, Z., Jiang, Y., & Zhang, R. (2014, August). The overview of big data storage and management. In *2014 IEEE 13th International Conference on Cognitive Informatics and Cognitive Computing* (pp. 510-513). IEEE.

7- Jadhav, D. K. (2013). Big data: the new challenges in data mining. *International Journal of Innovative Research in Computer Science & Technology*, *1*(2), 39-42.

8- Suthaharan, S. (2014). Big data classification: Problems and challenges in network intrusion prediction with machine learning. *ACM SIGMETRICS Performance Evaluation Review*, *41*(4), 70-73.

9- He, Q., Zhuang, F., Li, J., & Shi, Z. (2010, October). Parallel implementation of classification algorithms based on MapReduce. In *International Conference on Rough Sets and Knowledge *Technology* (pp. 655-662). Springer, Berlin, Heidelberg.

10- Bickson, D., Yom-Tov, E., & Dolev, D. (2008). A gaussian belief propagation solver for large scale support vector machines. *arXiv preprint arXiv:0810.1648*

11- Zanghirati, G., & Zanni, L. (2003). A parallel solver for large quadratic programs in training support vector machines. *Parallel computing*, *29*(4), 535-551.

12- Pakize, S. R., & Gandomi, A. (2014). Comparative study of classification algorithms based on MapReduce model. *International Journal of Innovative Research in Advanced Engineering (IJIRAE)*, *1*(7), 251-254.

13- Cui, W., Wang, G., & Xu, K. (2012). Parallel community mining in social network using map-reduce. *International Journal of Advancements in Computing Technology*, *4*(15), 445-453.

14- Li, Q., Salman, R., Test, E., Strack, R., & Kecman, V. (2013). Parallel multitask cross validation for support vector machine using GPU. *Journal of Parallel and Distributed Computing*, *73*(3), 293-302.

15- Çatak, F. Ö., & Balaban, M. E. (2016). A MapReduce-based distributed SVM algorithm for binary classification. *Turkish Journal of Electrical Engineering & Computer Sciences*, *24*(3), 863-873.

16- Xu, K., Wen, C., Yuan, Q., He, X., & Tie, J. (2014). A MapReduce based parallel SVM for email classification. *Journal of Networks*, *9*(6), 1640.

17- Naïve-bayes algorithm analysis

http://www.analyticsvidhya.com/blog/2015/09/naive-bayes-explained/ 30, retrieved in 2015.

18- Collobert, R., Bengio, S., & Bengio, Y. (2002). A parallel mixture of SVMs for very large-scale problems. In *Advances in Neural Information Processing Systems* (pp. 633-640).

19- Do, T. N., Nguyen, V. H., & Poulet, F. (2008, September). A fast parallel SVM algorithm for massive classification tasks. In *International Conference on Modelling, Computation and Optimization in Information Systems and Management Sciences* (pp. 419-428). Springer, Berlin, Heidelberg.

20- Prekopcsak, Z., Makrai, G., Henk, T., & Gaspar-Papanek, C. (2011, June). Radoop: Analyzing big data with rapidminer and hadoop. In *Proceedings of the 2nd RapidMiner community meeting and conference (RCOMM 2011)* (pp. 1-12).

21- Shim, K. (2012). MapReduce algorithms for big data analysis. *Proceedings of the VLDB Endowment*, *5*(12), 2016-2017

22- Rebentrost, P., Mohseni, M., & Lloyd, S. (2014). Quantum support vector machine for big data classification. *Physical review letters*, *113*(13), 130503.

23- ABDAR, M. (2015). A survey and compare the performance of IBM SPSS modeler and rapid miner software for predicting liver disease by using various data mining algorithms. *Cumhuriyet Üniversitesi Fen-Edebiyat Fakültesi Fen Bilimleri Dergisi*, *36*(3), 3230-3241.

24- Zheng, J., & Dagnino, A. (2014, October). An initial study of predictive machine learning analytics on large volumes of historical data for power system applications. In *2014 IEEE International Conference on Big Data (Big Data)* (pp. 952-959). IEEE.

25- Bello-Orgaz, G., Jung, J. J., & Camacho, D. (2016). Social big data: Recent achievements and new challenges. *Information Fusion*, *28*, 45-59.

26- Radoop, http://www.radoop.eu/, retrieved on July 2014.

27- RapidMiner, http://rapidminer.com/, retrieved on July 2014.

28- Zhu, K., Wang, H., Bai, H., Li, J., Qiu, Z., Cui, H., & Chang, E. Y. (2008). Parallelizing support vector machines on distributed computers. In *Advances in Neural Information Processing Systems* (pp. 257-264).

29- Z. Sheikh, "Proposal to implement Neural Network with backpropagation learning on Hadoop", https://issues.apache.org/jira/browse/MAHOUT- 364, retrieved on July 2014.

30- Meng, X., & Mahoney, M. (2013, February). Robust regression on mapreduce. In *International Conference on Machine Learning* (pp. 888-896).

31- Kiran, M., Kumar, A., Mukherjee, S., & Prakash, R. G. (2013). Verification and validation of MapReduce program model for parallel support vector machine algorithm on Hadoop cluster. *International Journal of Computer Science Issues (IJCSI)*, *10*(3), 317.

32- Rao, B. T., & Reddy, L. S. S. (2012). Survey on improved scheduling in Hadoop MapReduce in cloud environments. *arXiv preprint arXiv:1207.0780*.

33- Shvachko, K., Kuang, H., Radia, S., & Chansler, R. (2010, May). The hadoop distributed file system. In MSST (Vol. 10, pp. 1-10).

34- Vapnik, V. (2013). *The nature of statistical learning theory*. Springer science & business media.

35- Gunn, S. R. (1998). Support vector machines for classification and regression. *ISIS technical report*, *14*(1), 5-16.

36- Robinson, J. (2004). *The application of support vector machines to compression of digital images* (Doctoral dissertation, ResearchSpace@ Auckland).

37- Ertekin, S., Bottou, L., & Giles, C. L. (2010). Nonconvex online support vector machines. IEEE Transactions on Pattern Analysis and Machine Intelligence, 33(2), 368-381.

38- Tavara, Shirin. "Parallel computing of support vector machines: A survey." ACM Computing Surveys (CSUR) 51.6 (2019): 123.

39- Liu, Peng, et al. "Parallel naive Bayes algorithm for large-scale Chinese text classification based on spark." Journal of Central South University 26.1 (2019): 1-12.

40- Kyrkou, Christos, et al. "Embedded hardware-efficient real-time classification with cascade support vector machines." IEEE transactions on neural networks and learning systems 27.1 (2015): 99-112.

41- Do, Thanh-Nghi. "Non-linear classification of massive datasets with a parallel algorithm of local support vector machines." Advanced Computational Methods for Knowledge Engineering. Springer, Cham, (2015). 231-241.

42- Chen, Zhi, et al. "A parallel genetic algorithm-based feature selection and parameter optimization for support vector machine." Scientific Programming 2016 (2016).

43- Singh, Dinesh, Debaditya Roy, and C. Krishna Mohan. "DiP-SVM: distribution preserving kernel support vector machine for big data." IEEE Transactions on Big Data 3.1 (2016): 79-90.

44- Phu, Vo Ngoc, Vo Thi Ngoc Chau, and Vo Thi Ngoc Tran. "SVM for English semantic classification in parallel environment." International Journal of Speech Technology 20.3 (2017): 487-508.

45- Kshirsagar, Nabha, and N. Z. Tarapore. "GPU Parallel Computing of Support Vector Machines as applied to Intrusion Detection System." International Journal of Computer Science and Information Security (IJCSIS) 16.6 (2018).

46- Singh, Dinesh, and C. Krishna Mohan. "Projection-SVM: Distributed Kernel Support Vector Machine for Big Data using Subspace Partitioning." 2018 IEEE International Conference on Big Data (Big Data). IEEE, 2018.

47- C.-J. Hsieh, S. Si, and I. Dhillon, "A Divide-and-Conquer Solver for Kernel Support Vector Machines," in Proc. of Int. Conf. Machine Learning (ICML), Beijing, 21-26 Jun 2014, pp. 566–574.

48- Y. You, J. Demmel, K. Czechowski, L. Song, and R. Vuduc, "Design and Implementation of a Communication-Optimal Classifier for Distributed Kernel Support Vector Machines," IEEE Trans. Parallel and Distributed Systems, vol. 28, no. 4, pp. 974–988, 2017.

49- Thanigaivasan, Vivekanandan, Swathi J. Narayanan, and N. Ch Sriman Narayana Iyengar. "Analysis of Parallel SVM Based Classification Technique on Healthcare using Big Data Management in Cloud Storage." Recent Patents on Computer Science 11.3 (2018): 169-178.

50- Ksiaâ, Walid, Fahmi Ben Rejab, and Kaouther Nouira. "Big Data Classification: A Combined Approach Based on Parallel and Approx SVM." International Conference on Intelligent Interactive Multimedia Systems and Services. Springer, Cham, 2018.

51- Sadasivam, G. Sudha, et al. "Crop Disease Protection Using Parallel Machine Learning Approaches." Classification in BioApps. Springer, Cham, 2018. 227-259.

52- Goyal, Aarti, and T. Meenpal. "Kinship verification from facial images using feature descriptors." Cognitive Informatics and Soft Computing. Springer, Singapore, 2019. 371-380.

53- Rezvani, Salim, Xizhao Wang, and Farhad Pourpanah. "Intuitionistic Fuzzy Twin Support Vector Machines." *IEEE Transactions on Fuzzy Systems* (2019).

54- Xie, Juanying, et al. "An Efficient Global K-means Clustering Algorithm." *JCP* 6.2 (2011): 271-279.

55- Lin, Yujun, et al. "An improved clustering method based on k-means." *2012 9th International Conference on Fuzzy Systems and Knowledge Discovery*. IEEE, 2012.

56- Sookoian, Silvia, and Carlos J. Pirola. "The genetic epidemiology of nonalcoholic fatty liver disease: toward a personalized medicine." *Clinics in liver disease* 16.3 (2012): 467-485.

57- Gunnar Ratsch, “A Brief Introduction into Machine Learning”, Friedrich Miescher Laboratory of the Max Planck Society, 2004

58- Zhou, Lina, et al. "Machine learning on big data: Opportunities and challenges." *Neurocomputing* 237 (2017): 350-361

59- Zanghirati, Gaetano, and Luca Zanni. "A parallel solver for large quadratic programs in training support vector machines." *Parallel computing* 29.4 (2003): 535-551.

60- Yu, Hwanjo, Jiong Yang, and Jiawei Han. "Classifying large data sets using SVMs with hierarchical clusters." *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2003.

61- Arun, K., and L. Jabasheela. "Big data: review, classification and analysis survey." *International Journal of Innovative Research in Information Security (IJIRIS)* 1.3 (2014): 17-2.

62- A. Gandimi and M. Haider, “Beyond the hype: Big data concepts, methods and analytics”, Intl. J. Inform. Manage., Vol. 35, pp. 13714, 2015.

63- Mierswa, Ingo, et al. "Yale: Rapid prototyping for complex data mining tasks." *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006.

64- Zong, Yu, et al. "A Clustering Algorithm based on Local Accumulative Knowledge." *JCP* 8.2 (2013): 365-371.

65- RapidMiner Tool data mining and machine learning procedures https://docs.rapidminer.com 2006

66- Rostam Niakan Kalhori, Sharareh. *An Integrated Supervised and Unsupervised Learning Approach to Predict the Outcome of Tuberculosis Treatment Course*. Diss. The University of Manchester (United Kingdom), 2011. pp. 734-737, 2012.

67- Rebentrost, P., M. Mohseni, and S. Lloyd. "Quantum support vector machine for big feature and big data classifica-tion. CoRR, vol. abs/1307.0471, 2014." (2012): 76.

68- Shrivastava, Naveeen Kumar, Praneet Saurabh, and Bhupendra Verma. "An efficient approach parallel support vector machine for classification of diabetes dataset." *International Journal of Computer Applications* 36.6 (2011): 19-24.

69- Zhanquan, Sun, and Geoffrey Fox. "Large Scale Classification Based on Combination of Parallel SVM and Interpolative MDS." *Digital Sci. Center Publications, Tech. Rep* (2012).

70- Yao, Yukai, et al. "K-SVM: An Effective SVM Algorithm Based on K-means Clustering." *JCP* 8.10 (2013): 2632-2639.

71- Xiao, Han. "Towards parallel and distributed computing in large-scale data mining: A survey." *Technical University of Munich, Tech. Rep* (2010).

72- Graf, Hans P., et al. "Parallel support vector machines: The cascade svm." *Advances in neural information processing systems*. 2005,17, 521–528.

73- Joachims, Thorsten. "Making large-scale support vector machine learning practical, Advances in Kernel Methods." *Support vector learning* (1999).

74- Zhao, Hai-xiang, and Frédéric Magoules. "Parallel support vector machines on multi-core and multiprocessor systems." *11th International Conference on Artificial Intelligence and Applications (AIA 2011)*. IASTED, 2011.

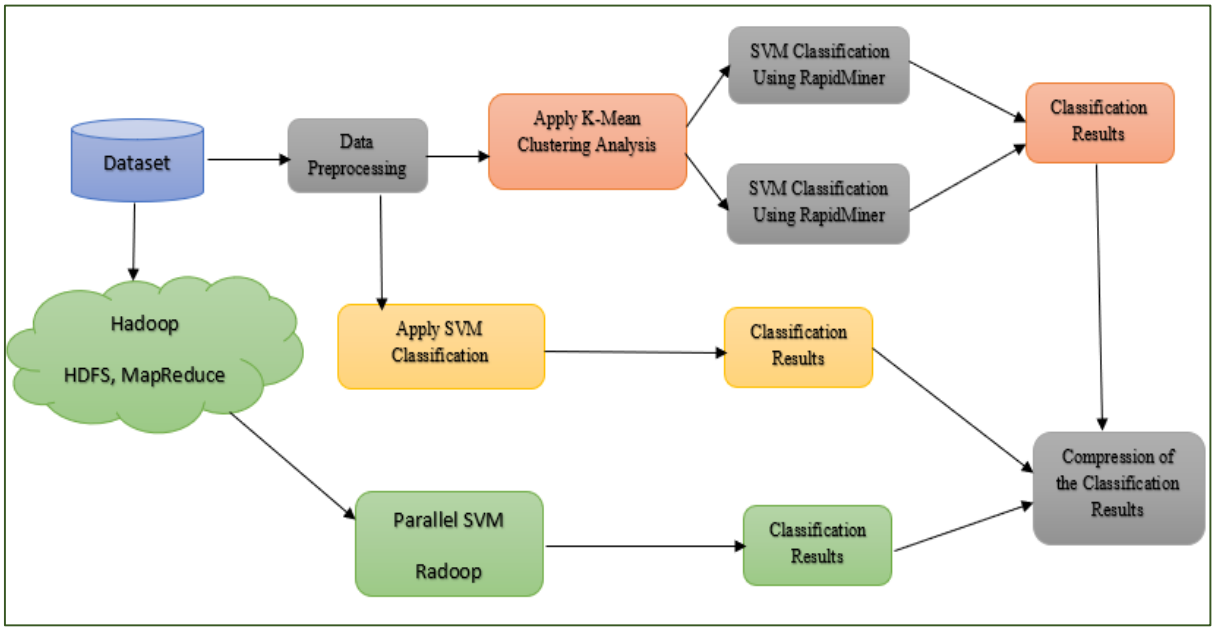
75- Dhillon, Supreet, and Kamaljit Kaur. "Comparative Study of Classification Algorithms for Web Usage Mining." *International Journal of Advanced Research in Computer Science and Software Engineering* 4.7 (2014): 137-140.

76- Bache, Kevin, and Moshe Lichman. "UCI Machine Learning Repository [http://archive. ics. uci. edu/ml]. Irvine, CA: University of California." *School of information and computer science* 28 (2013).

77- ICDM 2006 Panel on Top 10 Data mining algorithms. 12/21/2006, Coordinators: Xindong Wu and Vipin Kumar.

Appendix I

RapidMiner



Row No.	class	Turbidity	Odor	PH	Temperature	Conductivity	TDS	TSS	T.Alkalinty	ph.ph alkalin...	T.H
1	0	11.600	0	7.800	19	174.400	87	0	95	0	57
2	1	10.400	0	7.800	20.500	170.900	85	0	95	0	52
3	0	15.300	0	7.800	22.900	196.400	98.100	0	100	0	57
4	0	7.900	0	7.700	23.600	213	106	9	95	0	10
5	0	7.900	0	7.900	28.500	208	114.400	14	105	0	11
6	1	11	0	7.700	28.700	248	124	11	95	0	10
7	0	10.800	0	7.800	22.800	303	151.800	13	110	0	13
8	0	16.700	0	7.700	25.500	229	114.500	18	90	0	10
9	0	10.900	0	7.700	25.200	243	121.100	17	80	0	10
10	1	11	0	7.800	27.200	214	106.700	17	85	0	88
11	1	13.600	0	8.300	28.100	223	111.900	15	95	5	10
12	0	15.100	0	7.600	23.800	207	103.500	20	85	0	56
13	1	12.400	0	7.600	21.500	222	111	14	100	0	60

Importing Water quality dataset.

Row No.	class	age	workclass	fnlwgt	education	education-n...	marital-status	occupation	relationship	race
1	<=50K	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White
2	<=50K	50	Self-emp-not...	83311	Bachelors	13	Married-civ-s...	Exec-manag...	Husband	White
3	<=50K	38	Private	215646	HS-grad	9	Divorced	Handlers-cle...	Not-in-family	White
4	<=50K	53	Private	234721	11th	7	Married-civ-s...	Handlers-cle...	Husband	Black
5	<=50K	28	Private	338409	Bachelors	13	Married-civ-s...	Prof-specialty	Wife	Black
6	<=50K	37	Private	284582	Masters	14	Married-civ-s...	Exec-manag...	Wife	White
7	<=50K	49	Private	160187	9th	5	Married-spo...	Other-service	Not-in-family	Black
8	>50K	52	Self-emp-not...	209642	HS-grad	9	Married-civ-s...	Exec-manag...	Husband	White
9	>50K	31	Private	45781	Masters	14	Never-married	Prof-specialty	Not-in-family	White
10	>50K	42	Private	159449	Bachelors	13	Married-civ-s...	Exec-manag...	Husband	White
11	>50K	37	Private	280464	Some-college	10	Married-civ-s...	Exec-manag...	Husband	Black
12	>50K	30	State-gov	141297	Bachelors	13	Married-civ-s...	Prof-specialty	Husband	Asian-Pac-Is...

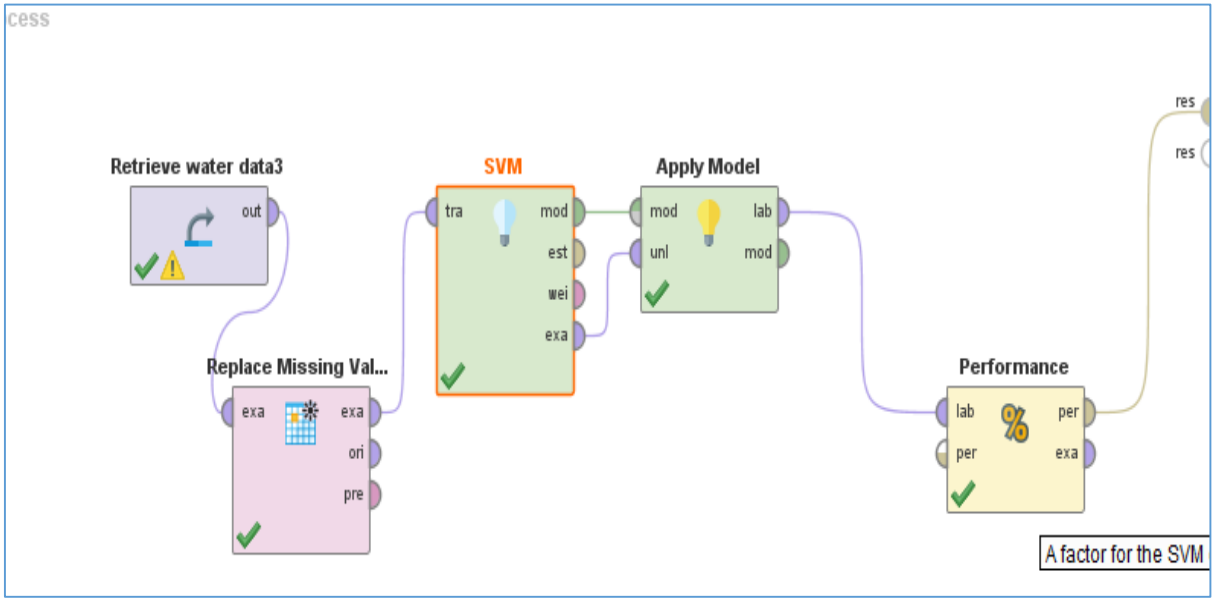
Importing Adult dataset.

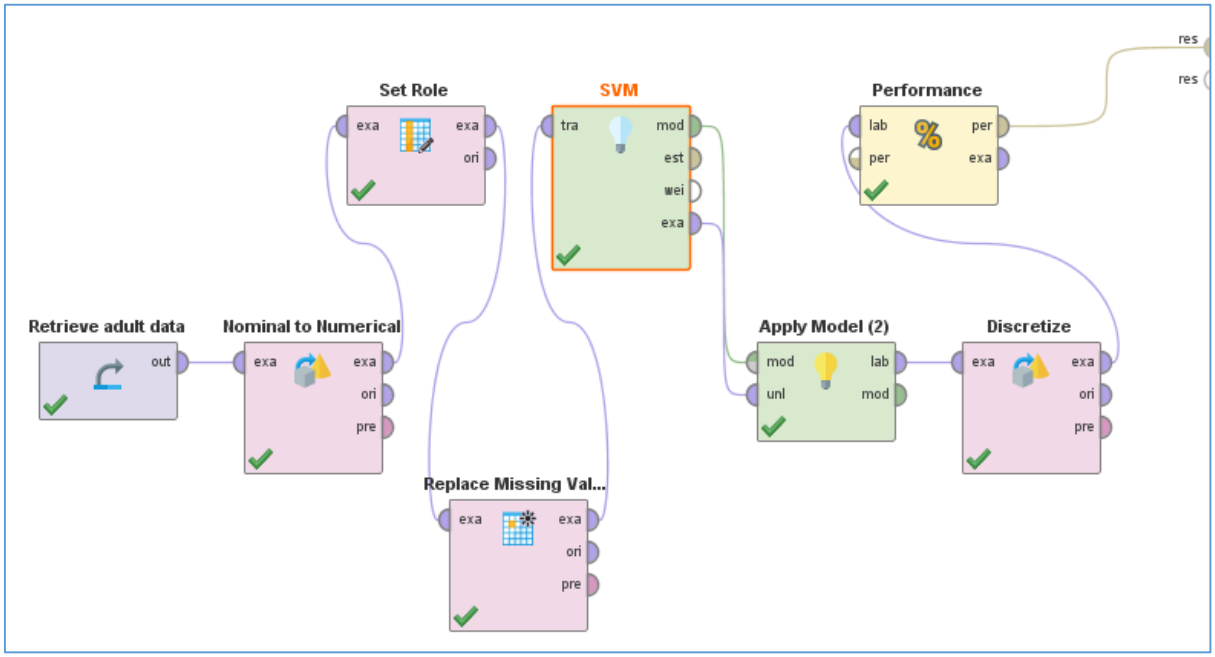
Row No.	diabetesMed	encounter_id	patient_nbr	race	gender	age	weight	admission_t...	discharge_d...	admission_...	tin
1											

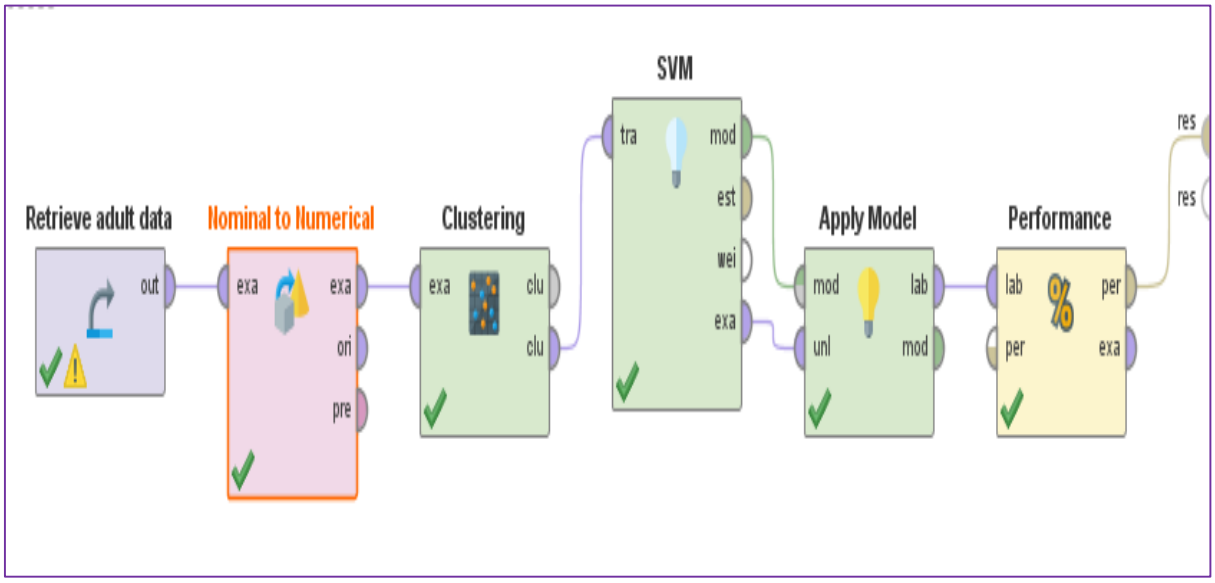
Row No.	Cover_Type	Elevation	Aspect	Slope	Horizontal_...	Vertical_Dis...	Horizontal_...	Hillshade_9...	Hillshade_N...	Hillshade_3...	Horizontal
1	5	2596	51	3	258	0	510	221	232	148	6279
2	5	2590	56	2	212	-6	390	220	235	151	6225
3	2	2804	139	9	268	65	3180	234	238	135	6121
4	2	2785	155	18	242	118	3090	238	238	122	6211
5	5	2595	45	2	153	-1	391	220	234	150	6172
6	2	2579	132	6	300	-15	67	230	237	140	6031
7	5	2606	45	7	270	5	633	222	225	138	6256
8	5	2605	49	4	234	7	573	222	230	144	6228
9	5	2617	45	9	240	56	666	223	221	133	6244
10	5	2612	59	10	247	11	636	228	219	124	6230
11	5	2612	201	4	180	51	735	218	243	161	6222
12	2	2886	151	11	371	26	5253	234	240	136	4051

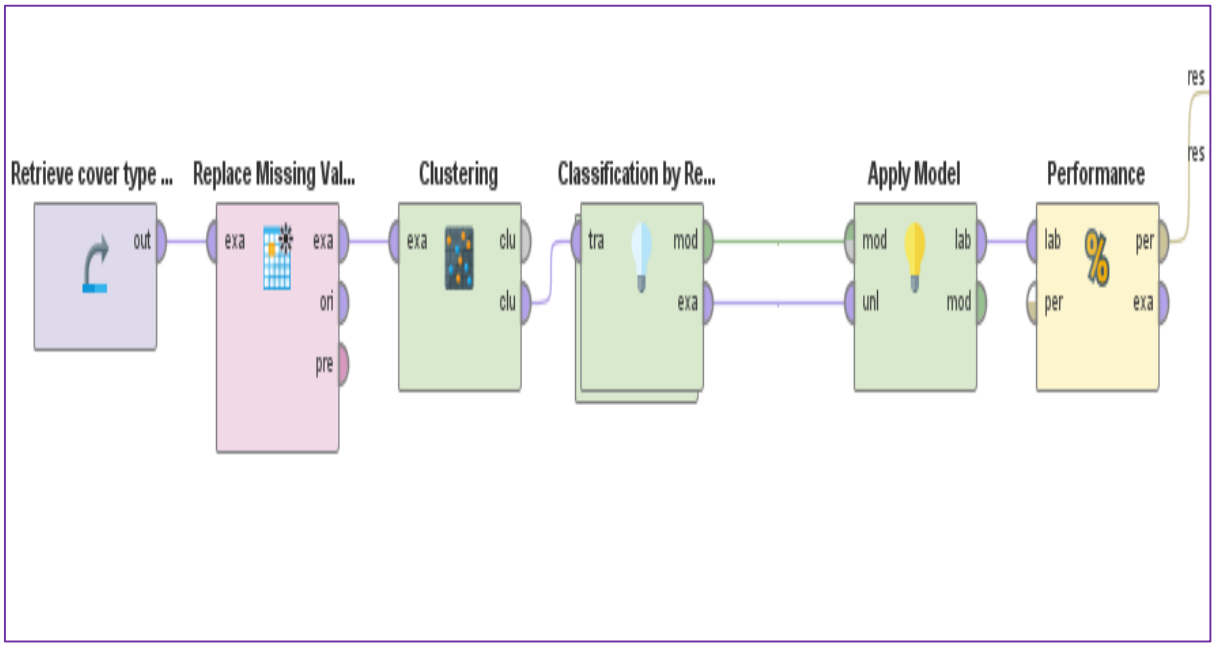
Importing Diabetes dataset.

importing Cover type dataset.

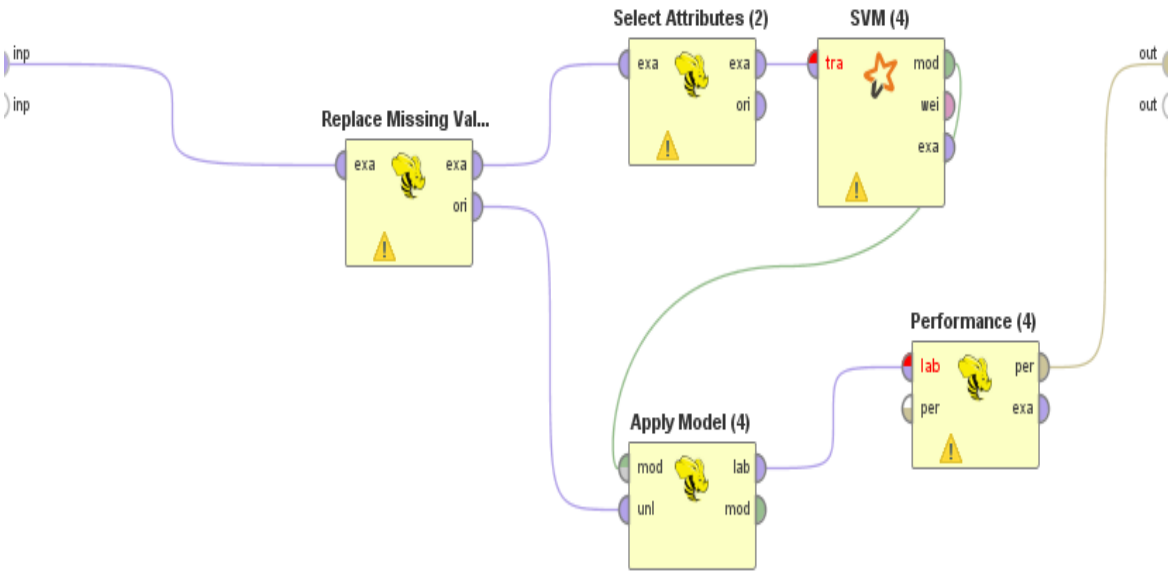

cess
Retrieve water data3
out
SVM
tra
mod
est
wei
exa
Apply Model
mod
lab
unl
mod
Replace Missing Val...
exa
exa
ori
pre
Performance
lab
per
per
exa
res
res
A factor for the SVM


Set Role
exa
exa
ori
SVM
tra
mod
est
wei
exa
Performance
lab
per
per
exa
res
res
Retrieve adult data
out
Nominal to Numerical
exa
exa
ori
pre
Apply Model (2)
mod
lab
unl
mod
Discretize
exa
exa
ori
pre
Replace Missing Val...
exa
exa
ori
pre


Retrieve adult data
out
Nominal to Numerical
exa
exa
ori
pre
Clustering
exa
clu
clu
SVM
tra
mod
est
wei
exa
Apply Model
mod
lab
unl
mod
Performance
lab
per
per
exa
res
res

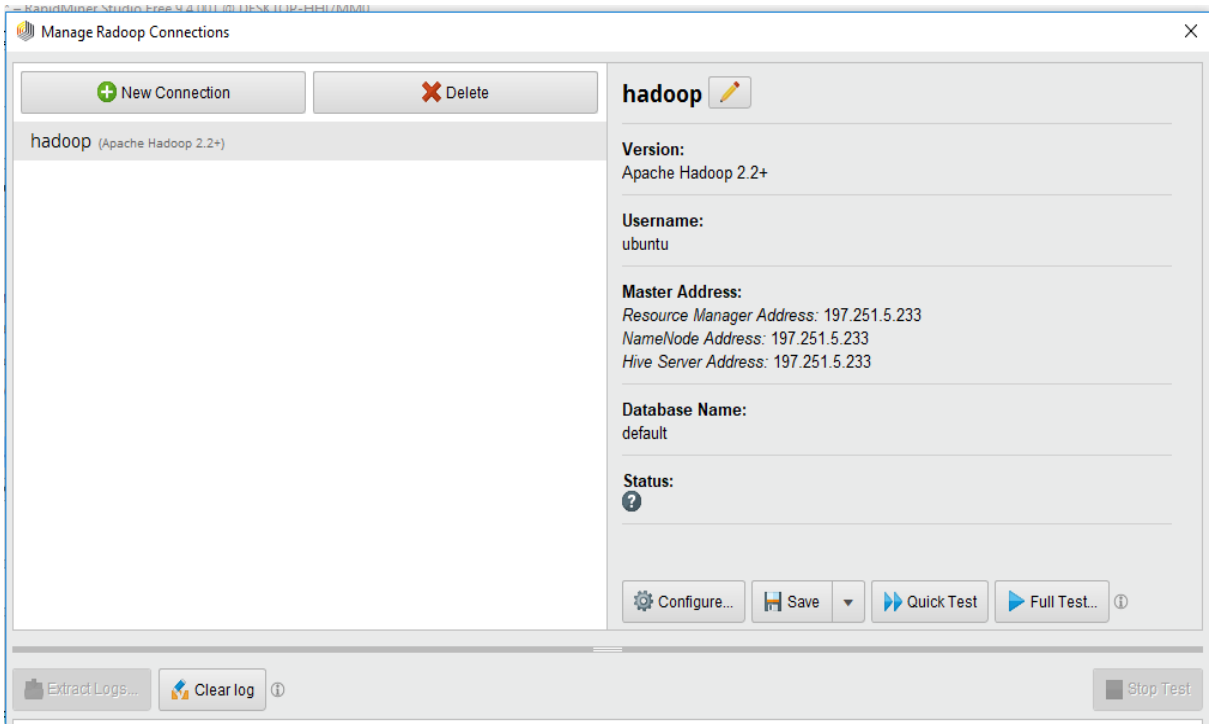

Retrieve cover type ...
out
Replace Missing Val...
exa
exa
ori
pre
Clustering
exa
clu
clu
Classification by Re...
tra
mod
exa
Apply Model
mod
lab
unl
mod
Performance
lab
per
per
exa
res
res

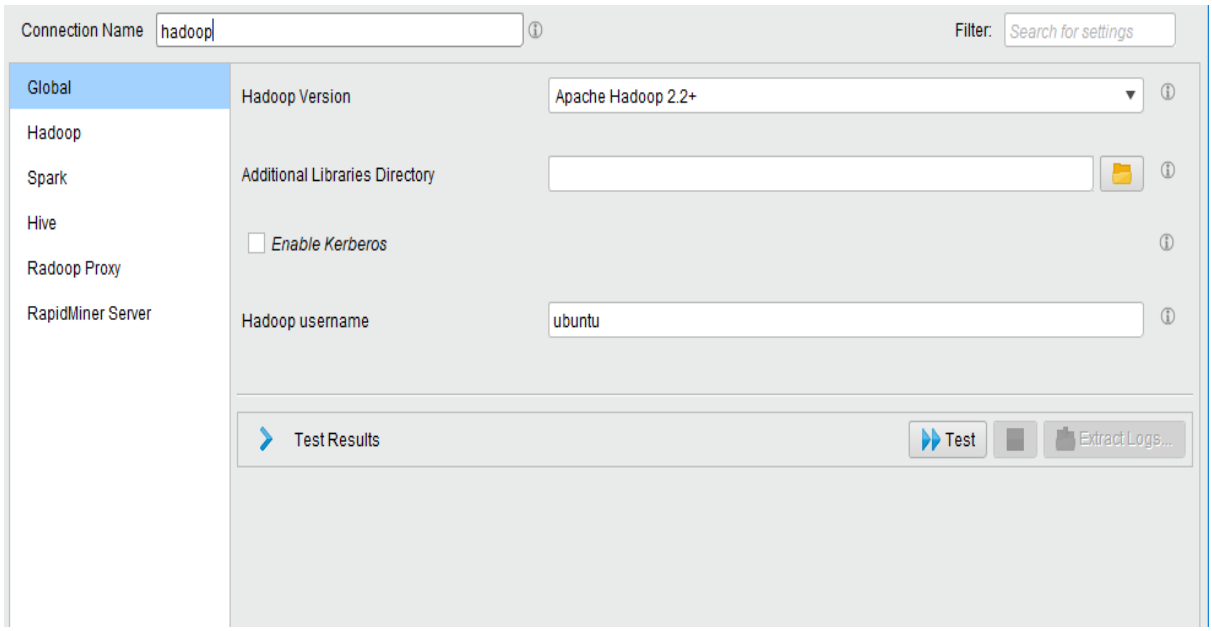

Retrieve
out
Radoop Nest
inp
inp
out
out
res
res

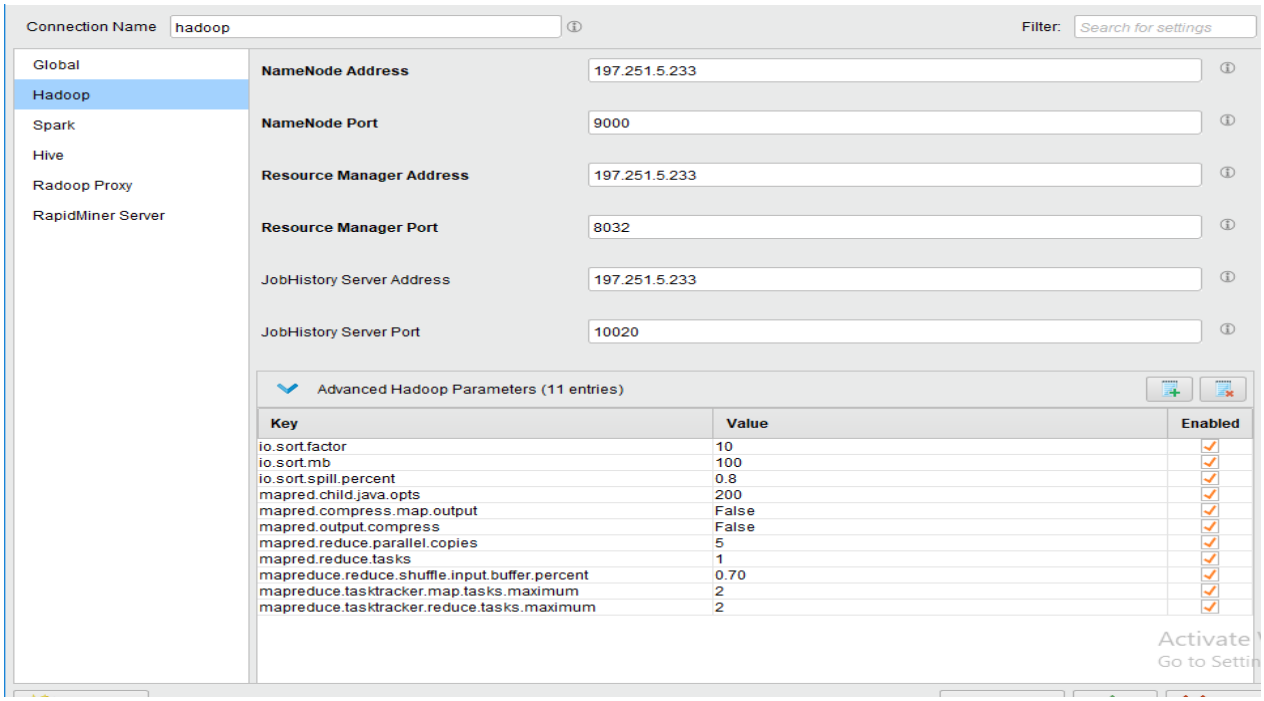


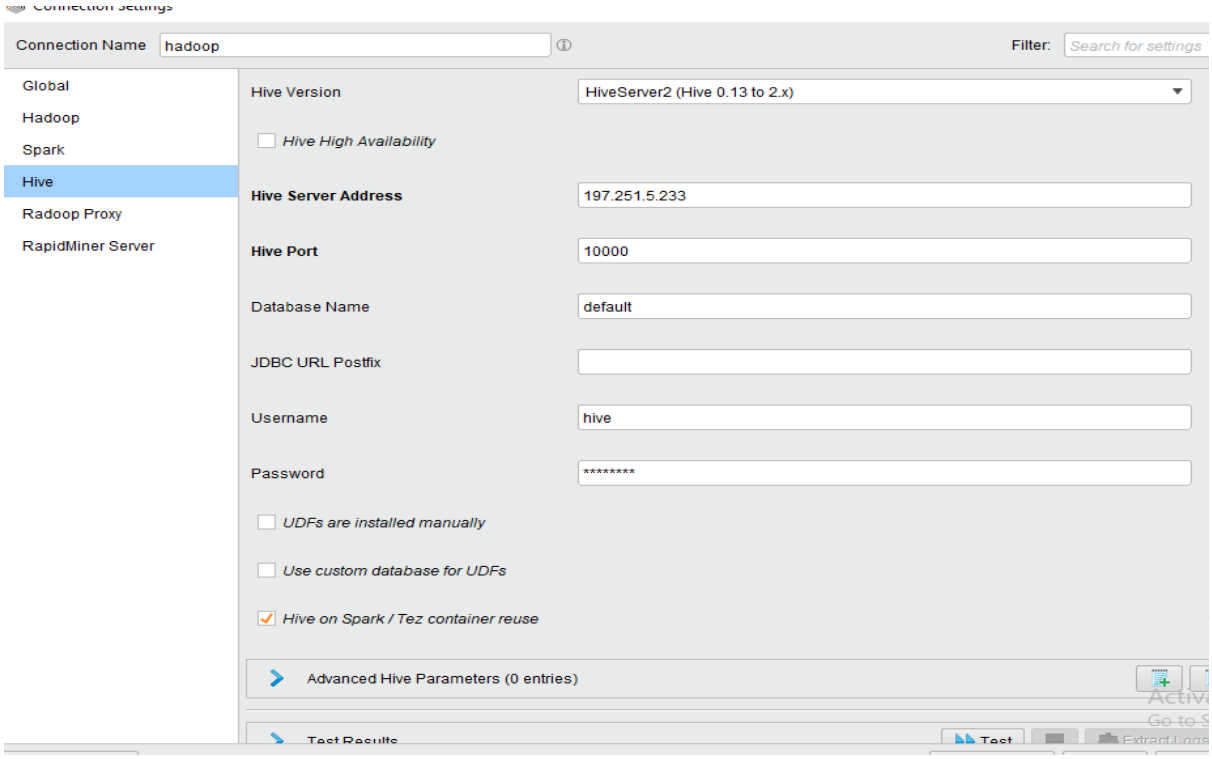
Appendix II

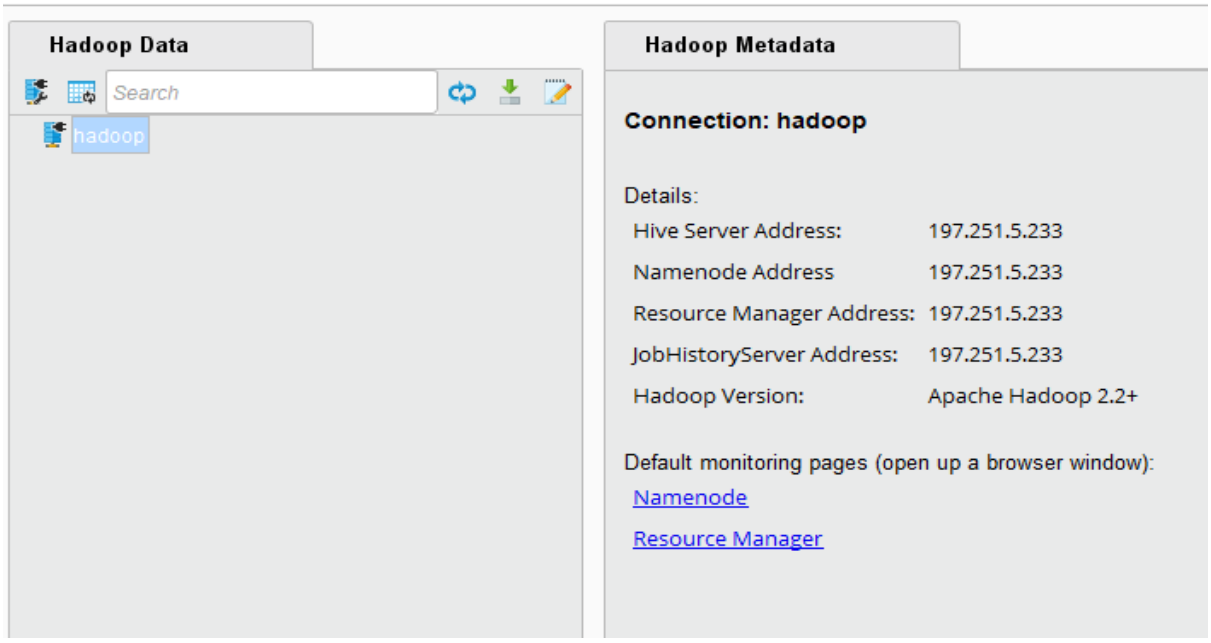
Hadoop Integration

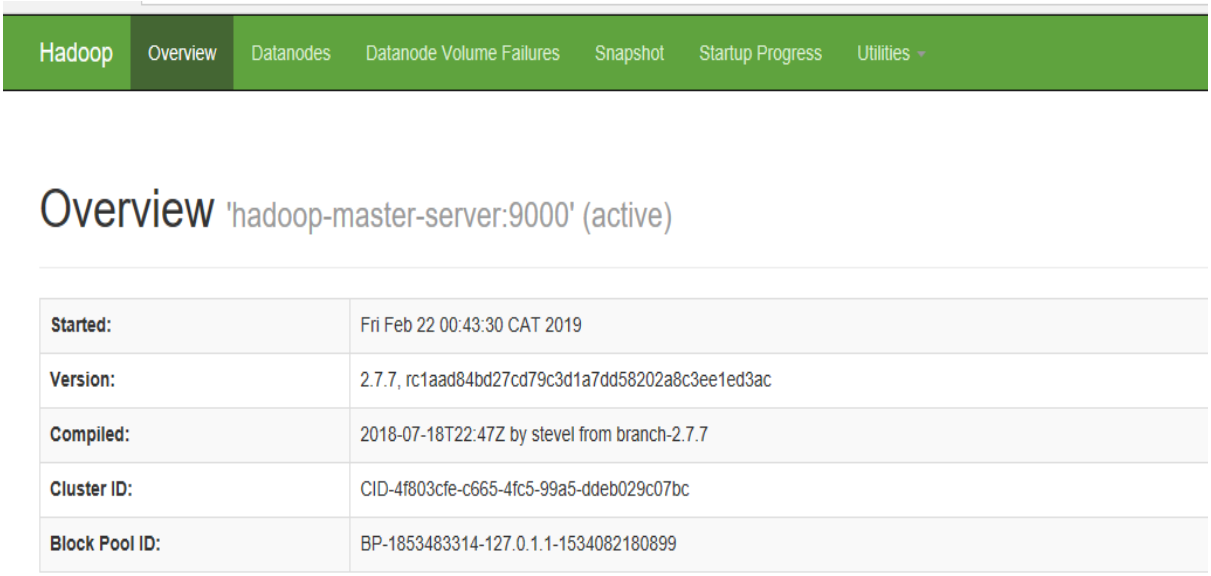

Manage Radoop Connections
New Connection
Delete
hadoop (Apache Hadoop 2.2+)
hadoop
Version:
Apache Hadoop 2.2+
Username:
ubuntu
Master Address:
Resource Manager Address: 197.251.5.233
NameNode Address: 197.251.5.233
Hive Server Address: 197.251.5.233
Database Name:
default
Status:
Configure...
Save
Quick Test
Full Test...
Extract Logs...
Clear log
Stop Test


Connection Name
hadoop
Filter:
Search for settings
Global
Hadoop
Spark
Hive
Radoop Proxy
RapidMiner Server
Hadoop Version
Apache Hadoop 2.2+
Additional Libraries Directory
Enable Kerberos
Hadoop username
ubuntu
Test Results
Test
Extract Logs...

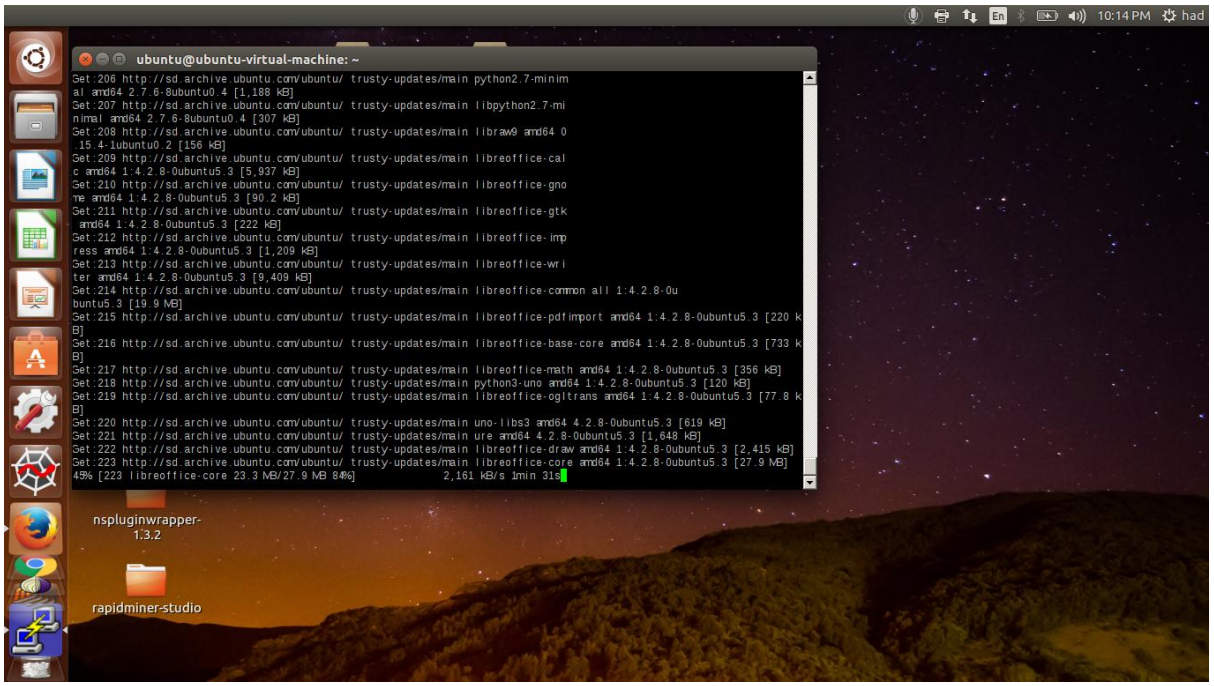

Connection Name
hadoop
Filter:
Search for settings
Global
Hadoop
Spark
Hive
Radoop Proxy
RapidMiner Server
NameNode Address
197.251.5.233
NameNode Port
9000
Resource Manager Address
197.251.5.233
Resource Manager Port
8032
JobHistory Server Address
197.251.5.233
JobHistory Server Port
10020
Advanced Hadoop Parameters (11 entries)
Key
Value
Enabled
io.sort.factor 10
io.sort.mb 100
io.sort.spill.percent 0.8
mapred.child.java.opts 200
mapred.compress.map.output False
mapred.output.compress False
mapred.reduce.parallel.copies 5
mapred.reduce.tasks 1
mapreduce.reduce.shuffle.input.buffer.percent 0.70
mapreduce.tasktracker.map.tasks.maximum 2
mapreduce.tasktracker.reduce.tasks.maximum 2

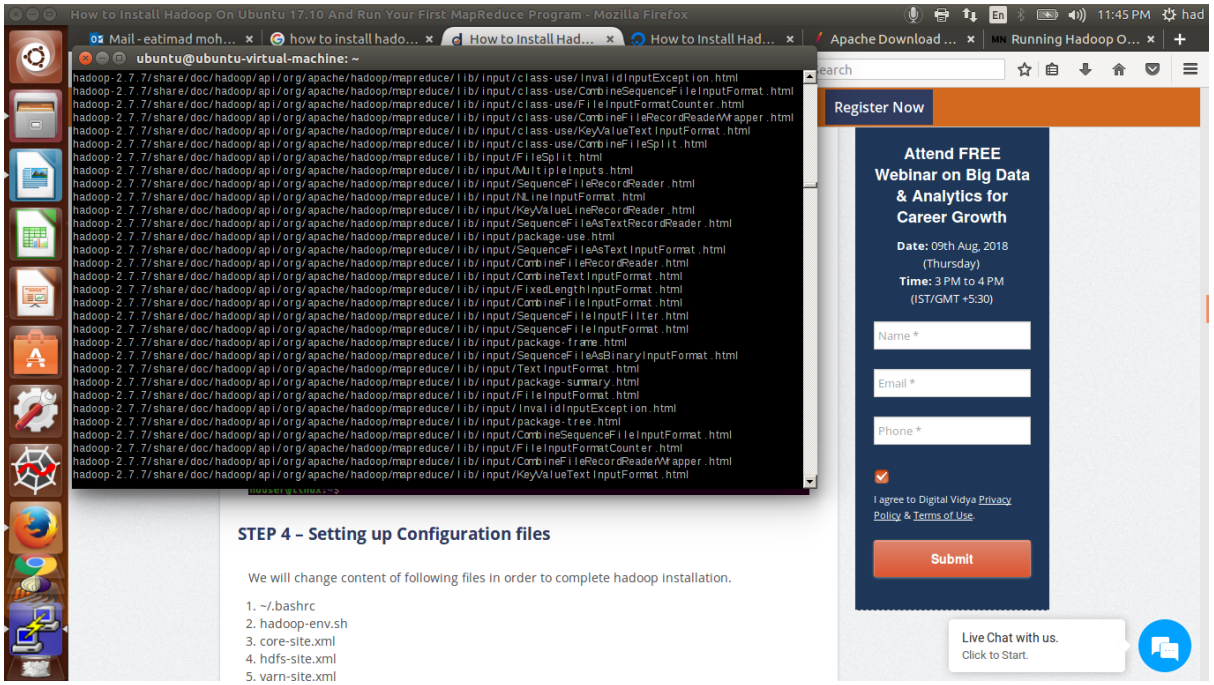

Connection Name
hadoop
Filter:
Search for settings
Global
Hadoop
Spark
Hive
Radoop Proxy
RapidMiner Server
Hive Version
HiveServer2 (Hive 0.13 to 2.x)
Hive High Availability
Hive Server Address
197.251.5.233
Hive Port
10000
Database Name
default
JDBC URL Postfix
Username
hive
Password
****
UDFs are installed manually
Use custom database for UDFs
Hive on Spark / Tez container reuse
Advanced Hive Parameters (0 entries)
Test Results
Test

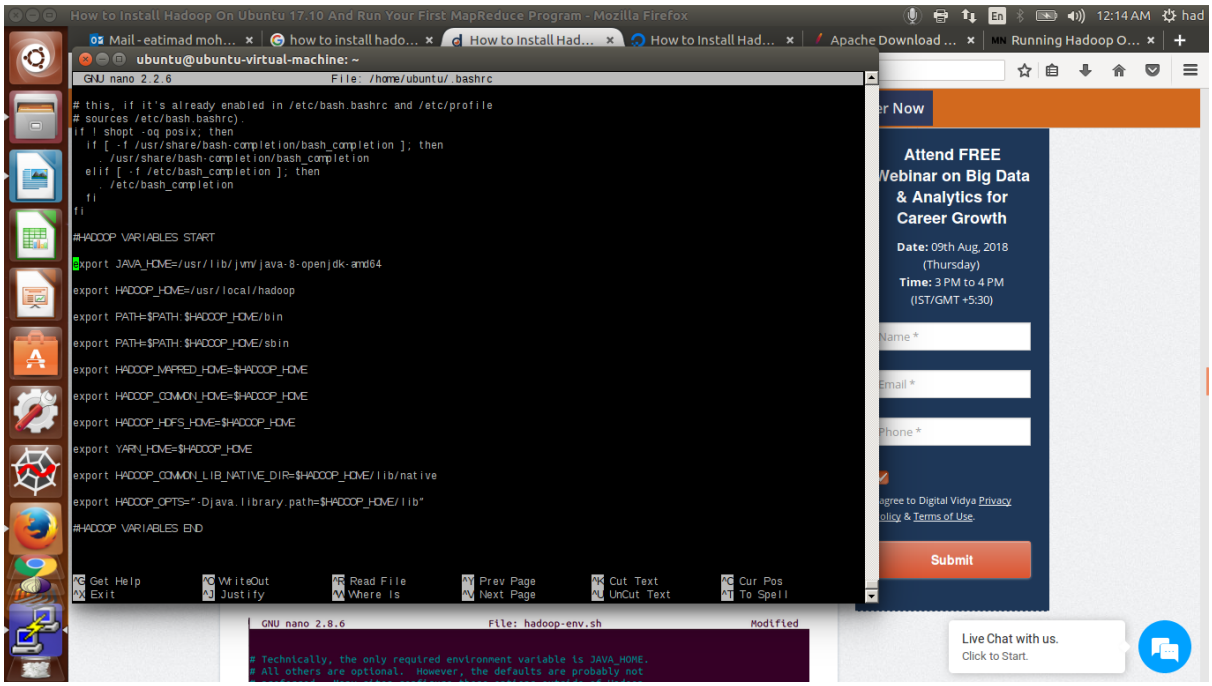




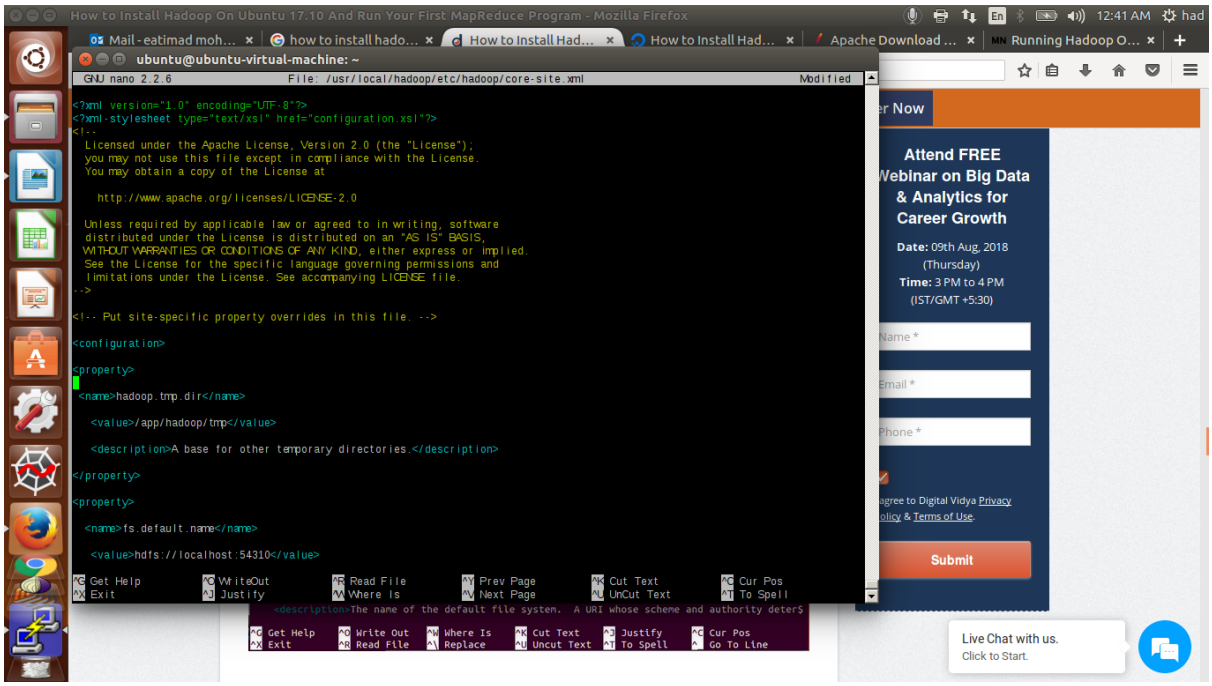
Hadoop installation

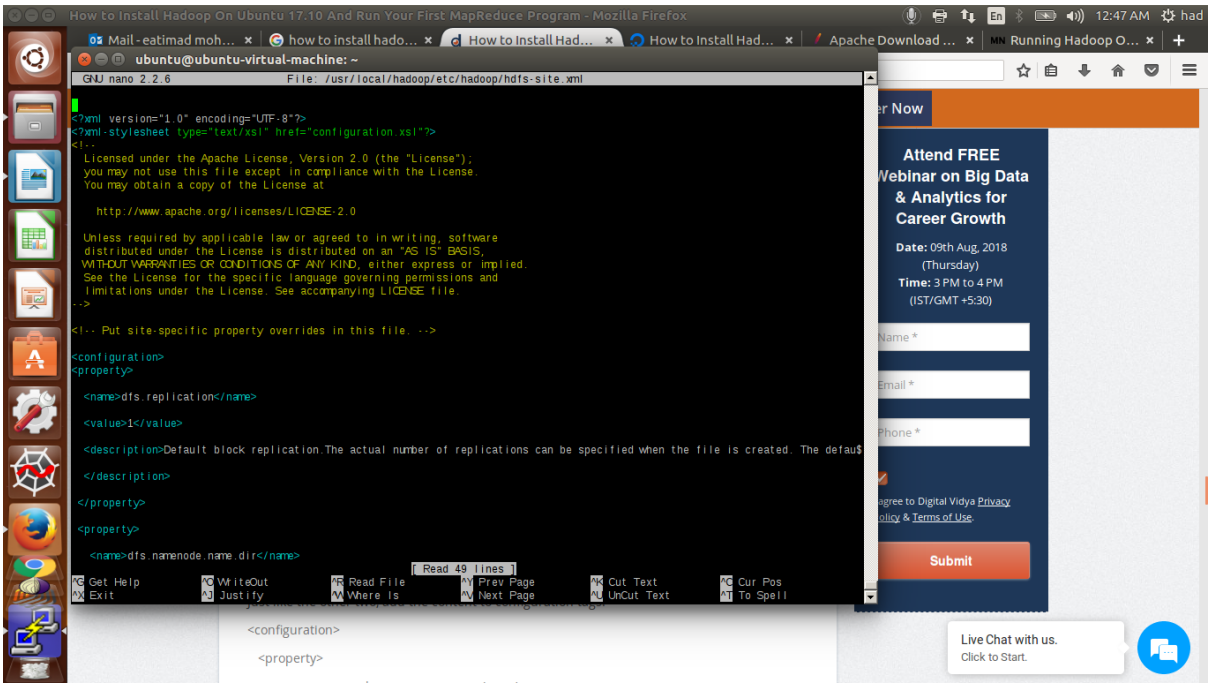

ubuntu@ubuntu-virtual-machine: ~
nspluginwrapper-1.3.2
rapidminer-studio

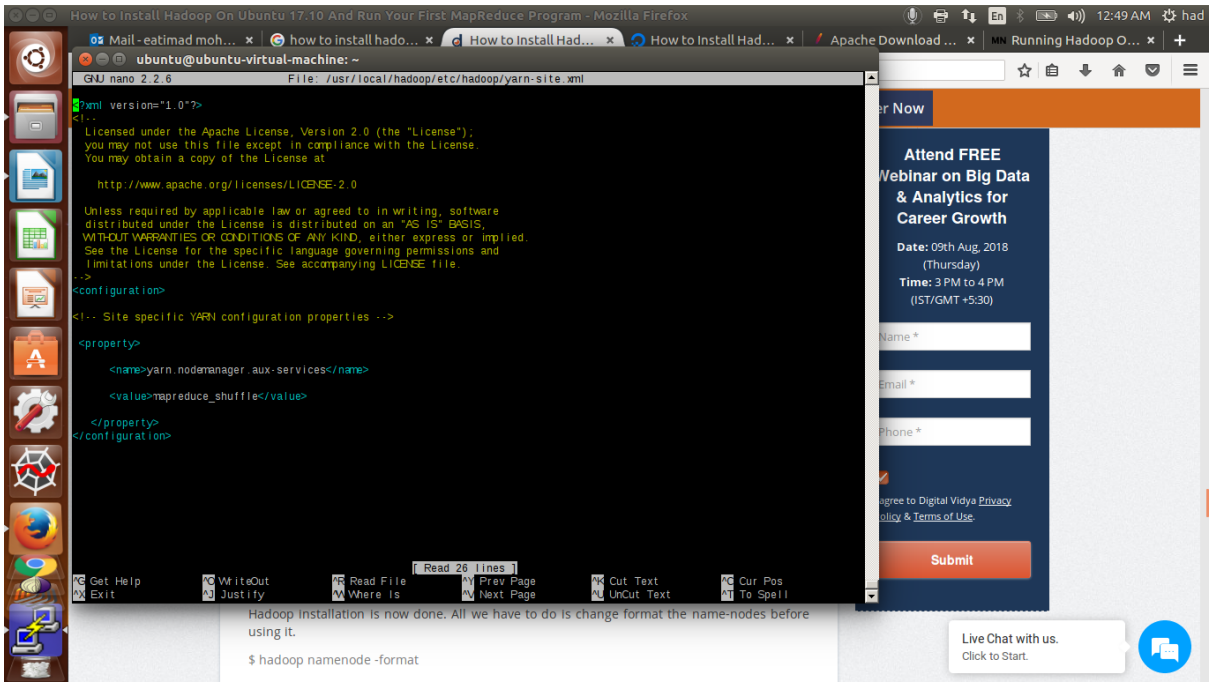

How to Install Hadoop On Ubuntu 17.10 And Run Your First MapReduce Program - Mozilla Firefox
ubuntu@ubuntu-virtual-machine: ~
STEP 4 – Setting up Configuration files
We will change content of following files in order to complete hadoop installation.
1. ~/.bashrc
2. hadoop-env.sh
3. core-site.xml
4. hdfs-site.xml
5. varn-site.xml
Register Now
Attend FREE Webinar on Big Data & Analytics for Career Growth
Date: 09th Aug, 2018 (Thursday)
Time: 3 PM to 4 PM (IST/GMT +5:30)
Name *
Email *
Phone *
I agree to Digital Vidya Privacy Policy & Terms of Use.
Submit
Live Chat with us.
Click to Start.

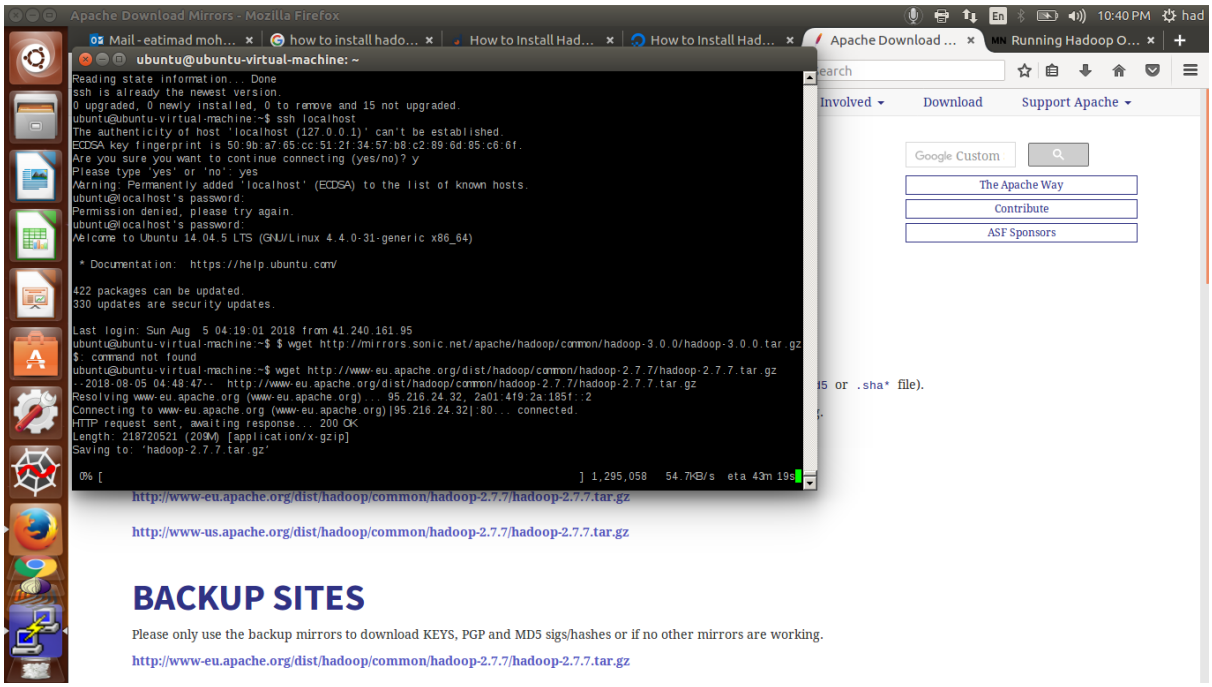

How to Install Hadoop On Ubuntu 17.10 And Run Your First MapReduce Program - Mozilla Firefox
ubuntu@ubuntu-virtual-machine: ~
GNU nano 2.2.6 File: /home/ubuntu/.bashrc
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
if [-f /usr/share/bash-completion/bash_completion]; then
. /usr/share/bash-completion/bash_completion
elif [-f /etc/bash_completion]; then
. /etc/bash_completion
fi
fi
#HADOOP VARIABLES START
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib"
#HADOOP VARIABLES END
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
GNU nano 2.8.6 File: hadoop-env.sh Modified
Technically, the only required environment variable is JAVA_HOME.
All others are optional. However, the defaults are probably not
Attend FREE Webinar on Big Data & Analytics for Career Growth
Date: 09th Aug, 2018 (Thursday)
Time: 3 PM to 4 PM (IST/GMT +5:30)
Name *
Email *
Phone *
Submit
Live Chat with us.
Click to Start.

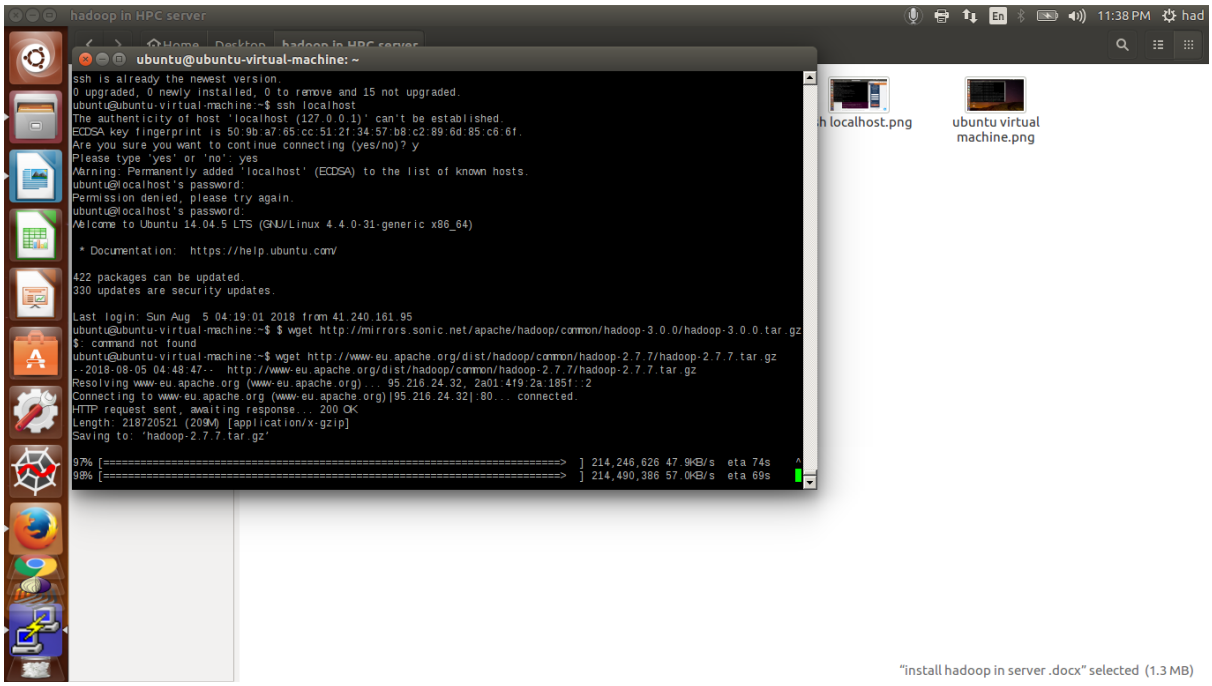

How to Install Hadoop On Ubuntu 17.10 And Run Your First MapReduce Program - Mozilla Firefox
ubuntu@ubuntu-virtual-machine: ~
GNU nano 2.2.6 File: /usr/local/hadoop/etc/hadoop/hadoop-env.sh
export JAVA_HOME=${JAVA_HOME}
Licensed to the Apache Software Foundation (ASF) under one
or more contributor license agreements. See the NOTICE file
distributed with this work for additional information
regarding copyright ownership. The ASF licenses this file
to you under the Apache License, Version 2.0 (the
"License"); you may not use this file except in compliance
with the License. You may obtain a copy of the License at
#
http://www.apache.org/licenses/LICENSE-2.0
#
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
Set Hadoop-specific environment variables here.
The only required environment variable is JAVA_HOME. All others are
optional. When running a distributed configuration it is best to
set JAVA_HOME in this file, so that it is correctly defined on
remote nodes.
The java implementation to use.
export JAVA_HOME=usr/lib/jvm/java-8-openjdk-amd64
The jsvc implementation to use. Jsvc is required to run secure datanodes
that bind to privileged ports to provide authentication of data transfer
protocol. Jsvc is not required if SASL is configured for authentication of
data transfer protocol using non-privileged ports.
#export JSVC_HOME=${JSVC_HOME}
export HADOOP_CONF_DIR=${HADOOP_CONF_DIR:-"/etc/hadoop"}
[Read 100 lines]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
<name>hadoop.tmp.dir</name>
<value>/app/hadoop/tmp</value>
<description>A base for other temporary directories.</description>
Attend FREE Webinar on Big Data & Analytics for Career Growth
Date: 09th Aug, 2018 (Thursday)
Time: 3 PM to 4 PM (IST/GMT +5:30)
Name *
Email *
Phone *
Submit
Live Chat with us.
Click to Start.


How to Install Hadoop On Ubuntu 17.10 And Run Your First MapReduce Program - Mozilla Firefox
ubuntu@ubuntu-virtual-machine: ~
GNU nano 2.2.6 File: /usr/local/hadoop/etc/hadoop/core-site.xml Modified
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License. See accompanying LICENSE file.
-->
<!-- Put site-specific property overrides in this file. -->
<configuration>
<property>
<name>hadoop.tmp.dir</name>
<value>/app/hadoop/tmp</value>
<description>A base for other temporary directories.</description>
</property>
<property>
<name>fs.default.name</name>
<value>hdfs://localhost:54310</value>
Attend FREE Webinar on Big Data & Analytics for Career Growth
Date: 09th Aug, 2018 (Thursday)
Time: 3 PM to 4 PM (IST/GMT +5:30)
Submit
Live Chat with us.
Click to Start.

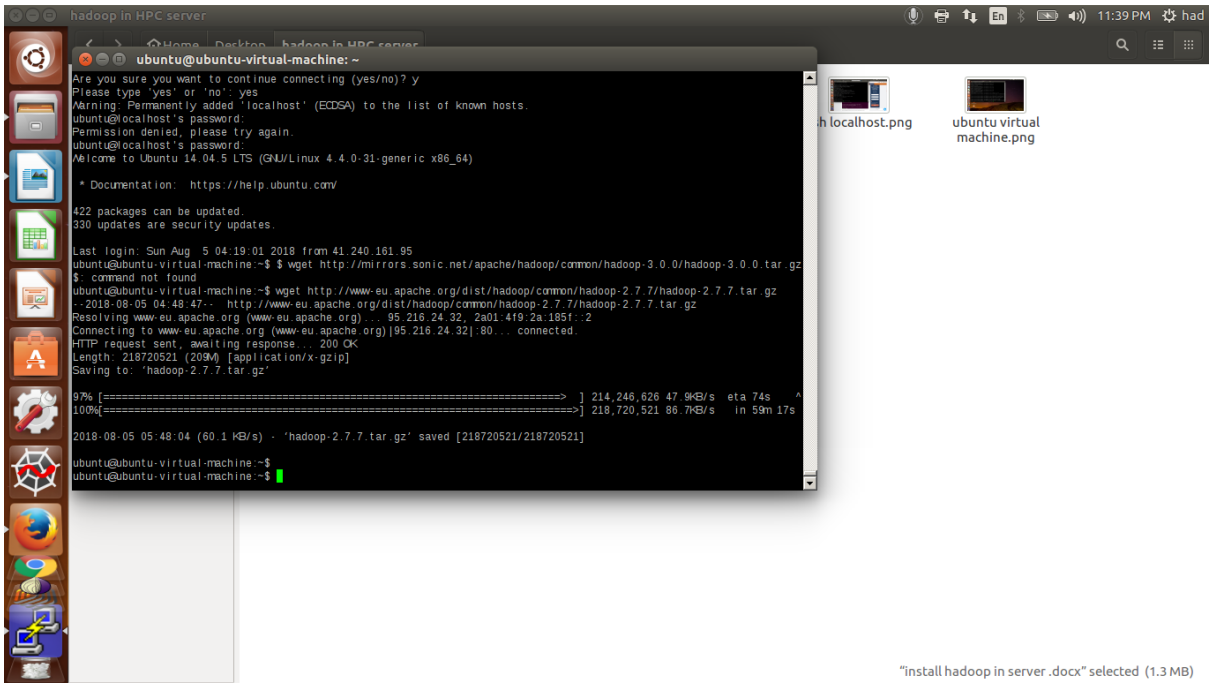

How to Install Hadoop On Ubuntu 17.10 And Run Your First MapReduce Program - Mozilla Firefox
ubuntu@ubuntu-virtual-machine: ~
GNU nano 2.2.6 File: /usr/local/hadoop/etc/hadoop/hdfs-site.xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License. See accompanying LICENSE file.
-->
<!-- Put site-specific property overrides in this file. -->
<configuration>
<property>
<name>dfs.replication</name>
<value>1</value>
<description>Default block replication.The actual number of replications can be specified when the file is created. The defau$
</description>
</property>
<property>
<name>dfs.namenode.name.dir</name>
[Read 49 lines]
Attend FREE Webinar on Big Data & Analytics for Career Growth
Date: 09th Aug, 2018 (Thursday)
Time: 3 PM to 4 PM (IST/GMT +5:30)
Submit
Live Chat with us.
Click to Start.


How to Install Hadoop On Ubuntu 17.10 And Run Your First MapReduce Program - Mozilla Firefox
ubuntu@ubuntu-virtual-machine: ~
GNU nano 2.2.6 File: /usr/local/hadoop/etc/hadoop/yarn-site.xml
<?xml version="1.0"?>
<!--
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License. See accompanying LICENSE file.
-->
<configuration>
<!-- Site specific YARN configuration properties -->
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
</configuration>
[Read 26 lines]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
Attend FREE Webinar on Big Data & Analytics for Career Growth
Date: 09th Aug, 2018 (Thursday)
Time: 3 PM to 4 PM (IST/GMT +5:30)
Submit
Hadoop installation is now done. All we have to do is change format the name-nodes before using it.
$ hadoop namenode -format
Live Chat with us.
Click to Start.

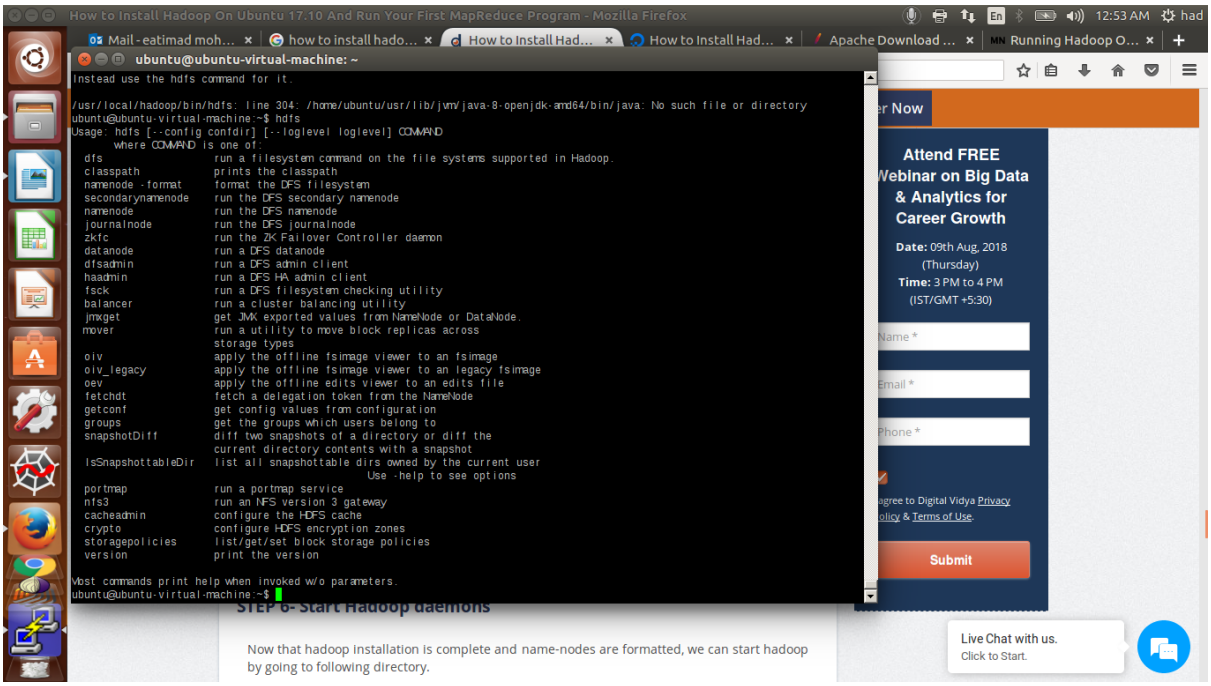

Apache Download Mirrors - Mozilla Firefox
ubuntu@ubuntu-virtual-machine: ~
Reading state information... Done
ssh is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 15 not upgraded.
ubuntu@ubuntu-virtual-machine:~$ ssh localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is 50:9b:a7:65:cc:51:2f:34:57:b8:c2:89:6d:85:c6:6f.
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
ubuntu@localhost's password:
Permission denied, please try again.
ubuntu@localhost's password:
Welcome to Ubuntu 14.04.5 LTS (GNU/Linux 4.4.0-31-generic x86_64)
* Documentation: https://help.ubuntu.com/
422 packages can be updated.
330 updates are security updates.
Last login: Sun Aug 5 04:19:01 2018 from 41.240.161.95
ubuntu@ubuntu-virtual-machine:~$ $ wget http://mirrors.sonic.net/apache/hadoop/common/hadoop-3.0.0/hadoop-3.0.0.tar.gz
$: command not found
ubuntu@ubuntu-virtual-machine:~$ wget http://www-eu.apache.org/dist/hadoop/common/hadoop-2.7.7/hadoop-2.7.7.tar.gz
--2018-08-05 04:48:47-- http://www-eu.apache.org/dist/hadoop/common/hadoop-2.7.7/hadoop-2.7.7.tar.gz
Resolving www-eu.apache.org (www-eu.apache.org)... 95.216.24.32, 2a01:4f9:2a:185f::2
Connecting to www-eu.apache.org (www-eu.apache.org)|95.216.24.32|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 218720521 (209M) [application/x-gzip]
Saving to: 'hadoop-2.7.7.tar.gz'
0% [ ] 1,295,058 54.7KB/s eta 43m 19s
Involved Download Support Apache
Google Custom
The Apache Way
Contribute
ASF Sponsors
http://www-eu.apache.org/dist/hadoop/common/hadoop-2.7.7/hadoop-2.7.7.tar.gz
http://www-us.apache.org/dist/hadoop/common/hadoop-2.7.7/hadoop-2.7.7.tar.gz
BACKUP SITES
Please only use the backup mirrors to download KEYS, PGP and MD5 sigs/hashes or if no other mirrors are working.
http://www-eu.apache.org/dist/hadoop/common/hadoop-2.7.7/hadoop-2.7.7.tar.gz

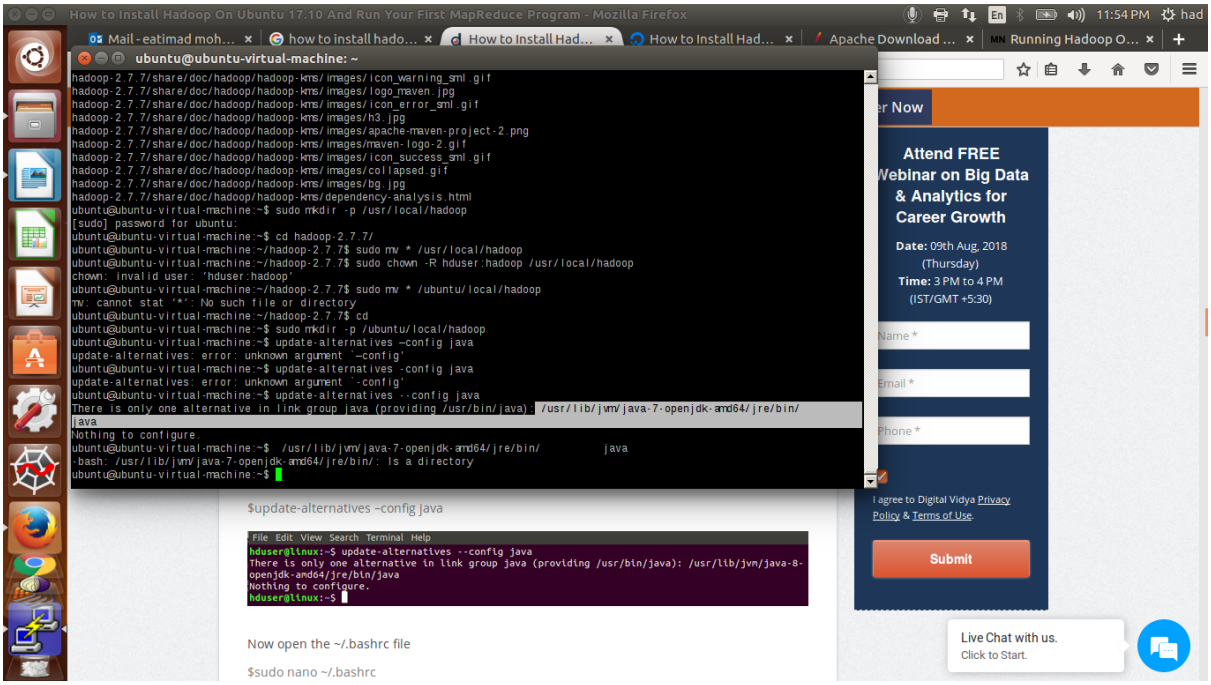


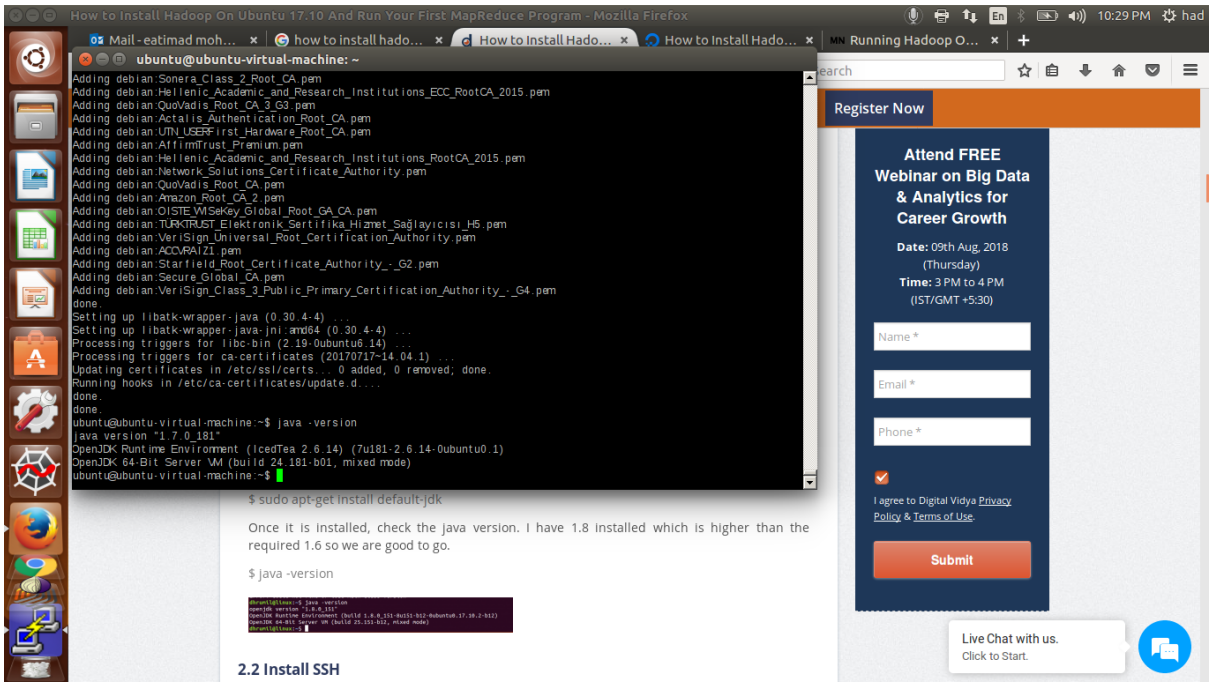
"install hadoop in server .docx" selected (1.3 MB)

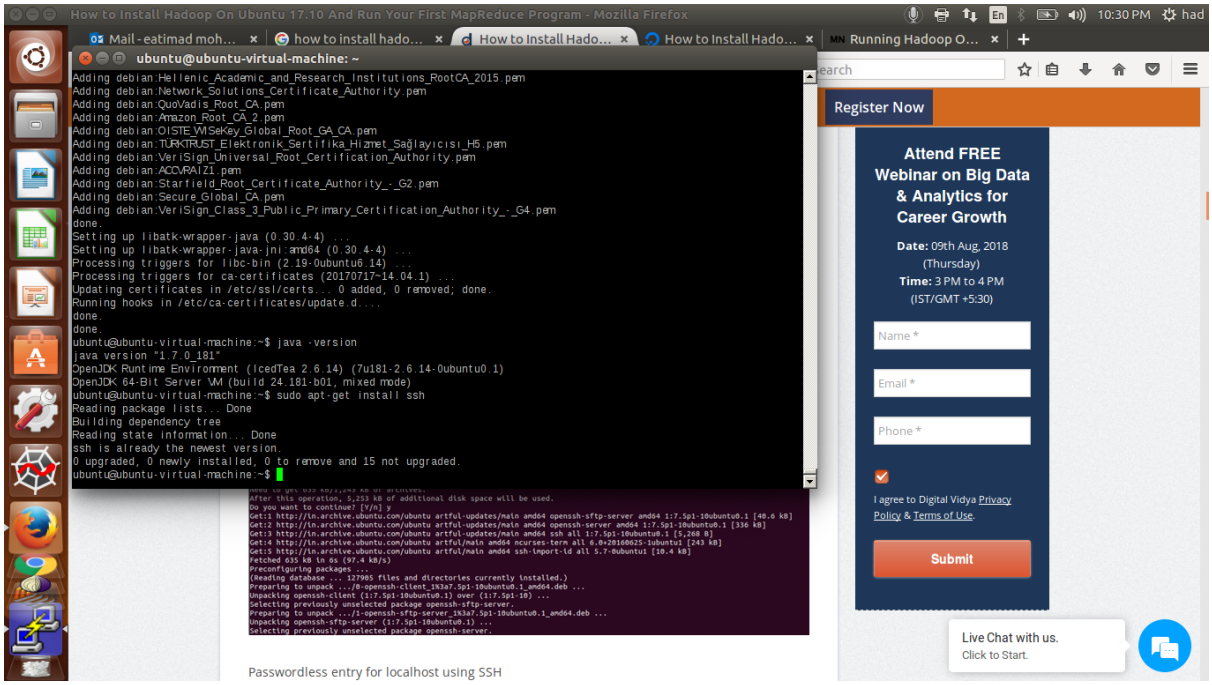


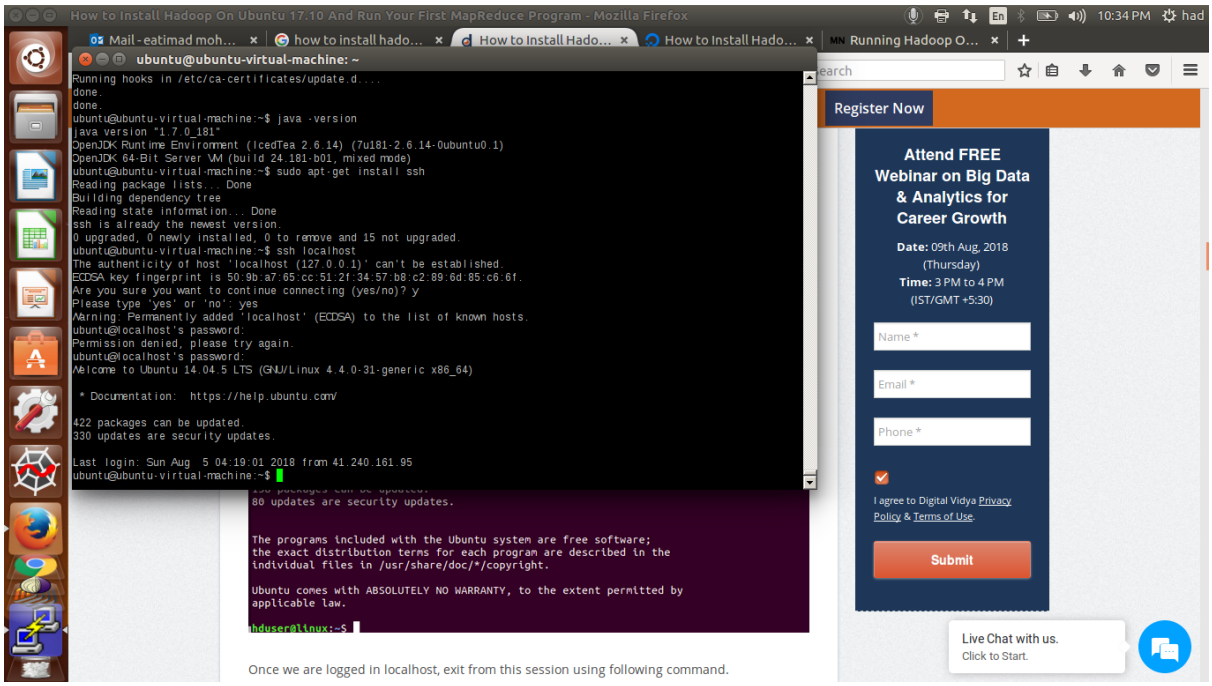
"install hadoop in server .docx" selected (1.3 MB)


How to Install Hadoop On Ubuntu 17.10 And Run Your First MapReduce Program - Mozilla Firefox
ubuntu@ubuntu-virtual-machine: ~
Instead use the hdfs command for it.
/usr/local/hadoop/bin/hdfs: line 304: /home/ubuntu/usr/lib/jvm/java-8-openjdk-amd64/bin/java: No such file or directory
ubuntu@ubuntu-virtual-machine:~$ hdfs
Usage: hdfs [--config confdir] [--loglevel loglevel] COMMAND
where COMMAND is one of:
dfs run a filesystem command on the file systems supported in Hadoop.
classpath prints the classpath
namenode -format format the DFS filesystem
secondarynamenode run the DFS secondary namenode
namenode run the DFS namenode
journalnode run the DFS journalnode
zkfc run the ZK Failover Controller daemon
datanode run a DFS datanode
dfsadmin run a DFS admin client
haadmin run a DFS HA admin client
fsck run a DFS filesystem checking utility
balancer run a cluster balancing utility
jmxget get JMX exported values from NameNode or DataNode.
mover run a utility to move block replicas across
storage types
oiv apply the offline fsimage viewer to an fsimage
oiv_legacy apply the offline fsimage viewer to an legacy fsimage
oev apply the offline edits viewer to an edits file
fetchdt fetch a delegation token from the NameNode
getconf get config values from configuration
groups get the groups which users belong to
snapshotDiff diff two snapshots of a directory or diff the
current directory contents with a snapshot
lsSnapshottableDir list all snapshottable dirs owned by the current user
Use -help to see options
portmap run a portmap service
nfs3 run an NFS version 3 gateway
cacheadmin configure the HDFS cache
crypto configure HDFS encryption zones
storagepolicies list/get/set block storage policies
version print the version
Most commands print help when invoked w/o parameters.
ubuntu@ubuntu-virtual-machine:~$
Attend FREE Webinar on Big Data & Analytics for Career Growth
Date: 09th Aug, 2018 (Thursday)
Time: 3 PM to 4 PM (IST/GMT +5:30)
Submit
Live Chat with us.
Click to Start.
Now that hadoop installation is complete and name-nodes are formatted, we can start hadoop by going to following directory.


How to Install Hadoop On Ubuntu 17.10 And Run Your First MapReduce Program - Mozilla Firefox
ubuntu@ubuntu-virtual-machine: ~
hadoop-2.7.7/share/doc/hadoop/hadoop-kms/images/icon_warning_sml.gif
hadoop-2.7.7/share/doc/hadoop/hadoop-kms/images/logo_maven.jpg
hadoop-2.7.7/share/doc/hadoop/hadoop-kms/images/icon_error_sml.gif
hadoop-2.7.7/share/doc/hadoop/hadoop-kms/images/h3.jpg
hadoop-2.7.7/share/doc/hadoop/hadoop-kms/images/apache-maven-project-2.png
hadoop-2.7.7/share/doc/hadoop/hadoop-kms/images/maven-logo-2.gif
hadoop-2.7.7/share/doc/hadoop/hadoop-kms/images/icon_success_sml.gif
hadoop-2.7.7/share/doc/hadoop/hadoop-kms/images/collapsed.gif
hadoop-2.7.7/share/doc/hadoop/hadoop-kms/images/bg.jpg
hadoop-2.7.7/share/doc/hadoop/hadoop-kms/dependency-analysis.html
ubuntu@ubuntu-virtual-machine:~$ sudo mkdir -p /usr/local/hadoop
[sudo] password for ubuntu:
ubuntu@ubuntu-virtual-machine:~$ cd hadoop-2.7.7/
ubuntu@ubuntu-virtual-machine:~/hadoop-2.7.7$ sudo mv * /usr/local/hadoop
ubuntu@ubuntu-virtual-machine:~/hadoop-2.7.7$ sudo chown -R hduser:hadoop /usr/local/hadoop
chown: invalid user: 'hduser:hadoop'
ubuntu@ubuntu-virtual-machine:~/hadoop-2.7.7$ sudo mv * /ubuntu/local/hadoop
mv: cannot stat '*': No such file or directory
ubuntu@ubuntu-virtual-machine:~/hadoop-2.7.7$ cd
ubuntu@ubuntu-virtual-machine:~$ sudo mkdir -p /ubuntu/local/hadoop
ubuntu@ubuntu-virtual-machine:~$ update-alternatives –config java
update-alternatives: error: unknown argument `–config'
ubuntu@ubuntu-virtual-machine:~$ update-alternatives -config java
update-alternatives: error: unknown argument `-config'
ubuntu@ubuntu-virtual-machine:~$ update-alternatives --config java
There is only one alternative in link group java (providing /usr/bin/java): /usr/lib/jvm/java-7-openjdk-amd64/jre/bin/java
Nothing to configure.
ubuntu@ubuntu-virtual-machine:~$ /usr/lib/jvm/java-7-openjdk-amd64/jre/bin/ java
-bash: /usr/lib/jvm/java-7-openjdk-amd64/jre/bin/: Is a directory
ubuntu@ubuntu-virtual-machine:~$
Attend FREE Webinar on Big Data & Analytics for Career Growth
Date: 09th Aug, 2018 (Thursday)
Time: 3 PM to 4 PM (IST/GMT +5:30)
Submit
Live Chat with us.
Click to Start.
$update-alternatives –config java
File Edit View Search Terminal Help
hduser@linux:~$ update-alternatives --config java
There is only one alternative in link group java (providing /usr/bin/java): /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java
Nothing to configure.
hduser@linux:~$
Now open the ~/.bashrc file
$sudo nano ~/.bashrc


How to Install Hadoop On Ubuntu 17.10 And Run Your First MapReduce Program - Mozilla Firefox
ubuntu@ubuntu-virtual-machine: ~
Adding debian:Sonera_Class_2_Root_CA.pem
Adding debian:Hellenic_Academic_and_Research_Institutions_ECC_RootCA_2015.pem
Adding debian:QuoVadis_Root_CA_3_G3.pem
Adding debian:Actalis_Authentication_Root_CA.pem
Adding debian:UTN_USERFirst_Hardware_Root_CA.pem
Adding debian:AffirmTrust_Premium.pem
Adding debian:Hellenic_Academic_and_Research_Institutions_RootCA_2015.pem
Adding debian:Network_Solutions_Certificate_Authority.pem
Adding debian:QuoVadis_Root_CA.pem
Adding debian:Amazon_Root_CA_2.pem
Adding debian:OISTE_WISeKey_Global_Root_GA_CA.pem
Adding debian:TURKTRUST_Elektronik_Sertifika_Hizmet_Sağlayıcısı_H5.pem
Adding debian:VeriSign_Universal_Root_Certification_Authority.pem
Adding debian:ACCVRAIZ1.pem
Adding debian:Starfield_Root_Certificate_Authority_-_G2.pem
Adding debian:Secure_Global_CA.pem
Adding debian:VeriSign_Class_3_Public_Primary_Certification_Authority_-_G4.pem
done.
Setting up libatk-wrapper-java (0.30.4-4) ...
Setting up libatk-wrapper-java-jni:amd64 (0.30.4-4) ...
Processing triggers for libc-bin (2.19-0ubuntu6.14) ...
Processing triggers for ca-certificates (20170717~14.04.1) ...
Updating certificates in /etc/ssl/certs... 0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d....
done.
done.
ubuntu@ubuntu-virtual-machine:~$ java -version
java version "1.7.0_181"
OpenJDK Runtime Environment (IcedTea 2.6.14) (7u181-2.6.14-0ubuntu0.1)
OpenJDK 64-Bit Server VM (build 24.181-b01, mixed mode)
ubuntu@ubuntu-virtual-machine:~$
$ sudo apt-get install default-jdk
Once it is installed, check the java version. I have 1.8 installed which is higher than the required 1.6 so we are good to go.
$ java -version
2.2 Install SSH
Register Now
Attend FREE Webinar on Big Data & Analytics for Career Growth
Date: 09th Aug, 2018 (Thursday)
Time: 3 PM to 4 PM (IST/GMT +5:30)
Name *
Email *
Phone *
I agree to Digital Vidya Privacy Policy & Terms of Use.
Submit
Live Chat with us.
Click to Start.


How to Install Hadoop On Ubuntu 17.10 And Run Your First MapReduce Program - Mozilla Firefox
ubuntu@ubuntu-virtual-machine: ~
Adding debian:Hellenic_Academic_and_Research_Institutions_RootCA_2015.pem
Adding debian:Network_Solutions_Certificate_Authority.pem
Adding debian:QuoVadis_Root_CA.pem
Adding debian:Amazon_Root_CA_2.pem
Adding debian:OISTE_WISeKey_Global_Root_GA_CA.pem
Adding debian:TURKTRUST_Elektronik_Sertifika_Hizmet_Sağlayıcısı_H5.pem
Adding debian:VeriSign_Universal_Root_Certification_Authority.pem
Adding debian:ACCVRAIZ1.pem
Adding debian:Starfield_Root_Certificate_Authority_-_G2.pem
Adding debian:Secure_Global_CA.pem
Adding debian:VeriSign_Class_3_Public_Primary_Certification_Authority_-_G4.pem
done.
Setting up libatk-wrapper-java (0.30.4-4) ...
Setting up libatk-wrapper-java-jni:amd64 (0.30.4-4) ...
Processing triggers for libc-bin (2.19-0ubuntu6.14) ...
Processing triggers for ca-certificates (20170717~14.04.1) ...
Updating certificates in /etc/ssl/certs... 0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d....
done.
done.
ubuntu@ubuntu-virtual-machine:~$ java -version
java version "1.7.0_181"
OpenJDK Runtime Environment (IcedTea 2.6.14) (7u181-2.6.14-0ubuntu0.1)
OpenJDK 64-Bit Server VM (build 24.181-b01, mixed mode)
ubuntu@ubuntu-virtual-machine:~$ sudo apt-get install ssh
Reading package lists... Done
Building dependency tree
Reading state information... Done
ssh is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 15 not upgraded.
ubuntu@ubuntu-virtual-machine:~$
Passwordless entry for localhost using SSH
Register Now
Attend FREE Webinar on Big Data & Analytics for Career Growth
Date: 09th Aug, 2018 (Thursday)
Time: 3 PM to 4 PM (IST/GMT +5:30)
Name *
Email *
Phone *
I agree to Digital Vidya Privacy Policy & Terms of Use.
Submit
Live Chat with us.
Click to Start.


How to Install Hadoop On Ubuntu 17.10 And Run Your First MapReduce Program - Mozilla Firefox
10:34 PM
Mail - eatimad moh...
how to install hado...
How to Install Hado...
How to Install Hado...
Running Hadoop O...
ubuntu@ubuntu-virtual-machine: ~
Running hooks in /etc/ca-certificates/update.d....
done.
done.
ubuntu@ubuntu-virtual-machine:~$ java -version
java version "1.7.0_181"
OpenJDK Runtime Environment (IcedTea 2.6.14) (7u181-2.6.14-0ubuntu0.1)
OpenJDK 64-Bit Server VM (build 24.181-b01, mixed mode)
ubuntu@ubuntu-virtual-machine:~$ sudo apt-get install ssh
Reading package lists... Done
Building dependency tree
Reading state information... Done
ssh is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 15 not upgraded.
ubuntu@ubuntu-virtual-machine:~$ ssh localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is 50:9b:a7:65:cc:51:2f:34:57:b8:c2:89:6d:85:c6:6f.
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
ubuntu@localhost's password:
Permission denied, please try again.
ubuntu@localhost's password:
Welcome to Ubuntu 14.04.5 LTS (GNU/Linux 4.4.0-31-generic x86_64)
* Documentation: https://help.ubuntu.com/
422 packages can be updated.
330 updates are security updates.
Last login: Sun Aug 5 04:19:01 2018 from 41.240.161.95
ubuntu@ubuntu-virtual-machine:~$
80 updates are security updates.
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
hduser@linux:~$
Once we are logged in localhost, exit from this session using following command.
Register Now
Attend FREE Webinar on Big Data & Analytics for Career Growth
Date: 09th Aug, 2018 (Thursday)
Time: 3 PM to 4 PM (IST/GMT +5:30)
Name *
Email *
Phone *
I agree to Digital Vidya Privacy Policy & Terms of Use.
Submit
Live Chat with us.
Click to Start.