# Efficient Machine Learning Techniques for Big Data Analytics

## A Thesis

*submitted for the award of the degree of*

## Doctor of Philosophy

in

**Computer Science and Engineering Department**

Submitted by

**Gaurav Sharma**
(Reg no: 901503007)

Under the Guidance of

**Dr. Seema Bawa**
Professor

**Dr. Prashant Singh Rana**
Associate Professor


**Thapar Institute of Engineering and Technology, Patiala,**
**Punjab - 147004, India**
**October 2021**

# Certificate

I hereby certify that the work, which is being presented in the thesis, entitled "Efficient Machine Learning Techniques for Big Data Analytics", in partial fulfillment of the requirements for the award of the degree of Doctor of Philosophy and submitted to the institution is an authentic record of my work carried out under the supervision of Dr. Seema Bawa and Dr. Prashant Singh Rana. I have cited the reference about the text(s)/figure(s)/table(s) from where they have been taken.
The matter presented in this thesis has not been submitted either in-part or full to any other University/Institute for the award of any other degree.

(Gaurav Sharma)
Registration No. 901503007

This is to certify that the above statements made by the candidate are correct and true to the best of my knowledge.

Verified by:

(Dr. Seema Bawa)
Supervisor
Computer Science and Engineering Department
Thapar Institute of Engineering and Technology, Patiala, Punjab, India

(Dr. Prashant Singh Rana)
Co - Supervisor
Computer Science and Engineering Department
Thapar Institute of Engineering and Technology, Patiala, Punjab, India

*Dedicated to my Spiritual Masters of Past and Present*

# Acknowledgements

I would like to express my deep gratitude to my supervisors Dr. Seema Bawa and Dr. Prashant Singh Rana for their invaluable advice, enthusiastic encouragement and useful critiques of this research work. Without their unfailing support and belief in me, this thesis would not have been possible. I would like to thanks Dr. Seema Bawa for introducing me to this interesting world of Big Data and suggesting this research topic. I am also grateful to my other supervisor Dr. Prashant Singh Rana for providing valuable guidance and sharing his machine learning expertise all the time during this work.

The contribution of my supervisors to this thesis goes well beyond their role as an academic supervisor and includes constant support on a personal level without which this journey may never have been completed. And for this, I am truly grateful. They are great mentor for my life as well.

I would like to express my gratitude to Dr. O. P. Pandey (former Dean of Research) for his constant motivation and encouragement. I am thankful to Dr. Maninder Singh (Head CSE Department) for being a source of motivation. I want to thank the members of my thesis committee: Dr. Rajendra Kumar Sharma, Dr. Shalini Batra and Dr. Harish Garg. They generously gave their time to offer insightful comments towards improving my work.

I am thankful to Dr. Ankit Kotia and Dr. Subrata Kumar Ghosh for sharing their experimental data and subject related expertise in the field of nano-lubricants. I would also extend my gratitude to Dr. Mohamed Kamal Ahmed Ali for guiding us to relate this interdisciplinary research work to industry problems.

I thank my fellow labmates for not only their stimulating discussions and valuable suggestions; but also for all the fun we shared. I truly appreciate my colleagues, including those from other disciplines, for enriching me by sharing their experiences. I'm thankful to the graduate and post-graduate students at TIET. My life would have been very dull without their fun interactions.

My hardworking parents deserve a special mention here. I would like to pay high regards to my father Mr. Vijay Sharma and my mother Mrs. Meera Sharma for their love, affection and blessings. My warmest thanks to my dear wife Asmita Pandey whose unconditional support during all these years is so appreciated. She inspired me in all dimensions of life and instilled confidence in me to successfully complete this journey and finally lots of love to my son Aadvik Sharma.

# Abstract

World's data is increasing at a tremendous rate, and many domains are becoming data-rich. New technological trends like the internet of things, cloud computing, smart devices etc. are responsible for this unprecedented data growth in several domains. Every domain is interested in gaining valuable insights by implementing knowledge discovery methods on the generated data to improve overall outcome or for some scientific breakthrough. However, gaining valuable insight from this big data comes with several challenges due to its inherent properties like carrying heterogeneous formats like structured, semi-structured or unstructured, growth rate and huge volume. The traditional machine learning and predictive analytics techniques face some significant limitations in terms of efficiency and accuracy when it comes to big data. The limitations of traditional tools and techniques have opened up vast opportunities for researchers worldwide to develop efficient machine learning techniques for big data problems.

There is no single machine learning algorithm that fits all scenarios, so there is a vast amount of research developing efficient machine learning techniques for different big data problems. Researchers are using different approaches like ensemble or a hybrid approach for developing a more accurate, efficient and reliable machine learning system for the problem in hand. Hybrid approaches usually involve integrating one machine learning technique with some other machine learning, heuristic, meta-heuristic or soft computing technique. On the other hand, ensemble machine learning techniques are build by combining various machine learning algorithms using grouping techniques like bagging, boosting and stacking.

In this thesis, hybrid and ensemble machine learning techniques are developed for big data problems in bioinformatics, material science and particle physics domains. In the first case study, hybrid machine learning techniques are developed to predict different types of human T-cell lymphotropic virus (HTLV) from semi-structured data, compromising protein sequences of different HTLVs and non-HTLV viruses. Hybrid machine learning techniques are built by combining supervised and unsupervised machine learning algorithms with greedy search and heuristic techniques. The machine learning system developed in this case study aims to assist the current diagnostic system for detecting HTLV-1 virus and gaining better insights about the virus by exploring the protein sequences' physicochemical properties extracted in this work.

In the second case study multi-criteria decision making (MCDM) based machine learning techniques are developed to predict the kinematic viscosity of three commercial grades

of lubricants namely gear oil, hydraulic oil and transmission oil deployed in heavy earth-moving vehicles. The experimental data for each lubricant category was collected by adding two different types of nano-particles at varying temperature and particle volume fraction. Four different machine learning techniques were trained on each category of nano-lubricants' experimental data, and their predictive efficiency was evaluated based on different model evaluation parameters. In the final step for finding the best predictive model in each category, the ranking of machine learning techniques is done basis on the model evaluation parameters using MCDM technique called Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS).

In the third case study, multilevel ensemble classifier is developed for dealing with the binary classification problem in the massive volume of data generated by particle colliders like Large Hadron Collider (LHC). In this work, four different supervised machine learning techniques are stacked to build ensemble classifier. Moreover, for dealing with the massive volume of data, the ensemble classifier is implemented using popular big data distributed platform Apache Spark on the AWS cloud. The multilevel ensemble classifier's efficiency is evaluated based on different model evaluation parameters, and comparative analysis of the results is done with the existing benchmark techniques.

The results obtained in all three case studies have proved the efficiency of hybrid and machine learning techniques developed for the respective problem in hand.

***Keywords:*** Big Data, Machine Learning, Ensemble Models, Hybrid Models, Heuristic Techniques, MCDM, Apache Spark

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

This chapter carries the introduction of the task conducted in this thesis. It underlines big data's fundamental concepts, challenges involved in handling big data, big data analytics, introduction to machine learning techniques and tools; their limitations in dealing with big data. The chapter also includes the brief introduction recently prevailing machine learning approaches like ensemble learning and cluster computing environments to deal with big data problems.

## 1.1 Background

The beginning of this new century, the intensity of data growth surpassed Moore's Law, and the better research paradigm was born as Data-Intensive Scientific Discovery (DISD), also known as the Big Data problem. Approximately every field, from economic and business practises to policy making, from national security to scientific research, from healthcare to evolutionary biology, from social networking to astronomy, involve the Big Data problems [1]. Data sets are multiplying, from diverse data provisions, such as sensor networks, telescopes, science studies, high-performance tools, business data, tablets, and the Internet. Similarly, vast amounts of publications classify the data as a structured, semi-structured or unstructured.

With a fall in prices of data storing devices and wider accessibility of high-performance computers, there are a significant increase in usage of machine learning (ML) techniques in various fields and sectors, including healthcare, bioinformatics, material sciences, banking, and finance, law enforcement, entertainment, and multimedia. It is seen that machine learning tools are increasingly becoming an integral part of many business and scientific operations. Machine learning helps make future predictions by an enabled smart system that learns from the existing past or current events. In the case of a supervised machine learning process there are three phases: create the model, testing and fine tuning the model and then bring it into development. The data is the basis that drives the models of machine learning, which is at the forefront of applications for science and industry [2].

The exponential growth of data in this last decade has given rise to data-intensive computing problems commonly, known as Big Data problems. It is observed that 90% of the world's data was generated in the last five years and this rapid data growth is making trouble to people in all the sectors ranging from business to public administration, scientific research to sentimental social analysis and many more. As predicted by International Data Corporation, data on our planet's storage devices is estimated to reach 44 zettabytes by 2020, and it is said to be ten times larger than it was in 2013. However, the massive volume of data generated by diversified heterogeneous sources like sensors, the internet, trade databases, government databases, high throughput scientific experiments, etc., possess significant hidden values and knowledge. This massive data need to be explored by efficient and fast analytics systems for evolutionary breakthroughs and increased productivity in different fields [3] [4].

Machine learning techniques possess the characteristics that can help discover knowledge, hidden values and making an automatic decision from this data. However, because of its distributed storage, this excessive data accumulation presents a challenge to existing machine learning techniques to effectively process and learn from big data. Commonly used machine learning tools like R or Weka do not support distributed storage and processing [2]. Apache Hadoop, an open-source implementation of the most popular Big Data processing framework MapReduce, helps solve scalability problems by offering distributed storage and processing [5]. Although Hadoop is a big data processing architecture and now has a RHadoop kit, which is just a R programming language API that can be deployed at the top of the Hadoop ecosystem to apply the machine learning algorithms available on the R platform to create better machine learning models from high-dimensional distributed data. Apache Mahout is another library on the top of the Hadoop ecosystem, which is mainly developed for creating scalable machine learning systems [6]. However, due to the iterative nature of machine learning algorithms and the MapReduce framework's inefficiency in dealing with iterative tasks, these big data machine learning platforms lack performance.

On the other hand, Apache Spark another open-source big data platform capable of efficiently dealing with streaming data and iterative tasks with its inherent architectural properties of in-memory computation is gaining more popularity these days over Hadoop. Moreover, Spark provides the flexibility of writing applications in different programming languages by providing APIs in Scala, Java, Python, and R [7]. Spark also has a robust machine learning library called MLLib, which includes commonly used supervised, semi-supervised and unsupervised machine learning algorithms.

### 1.1.1 Big Data

Big Data is a vast volume of data about a single system which is too huge and too complicated to manage. In 2012, Gartner defined Big Data as a high-volume, high-speed and high-variety information resource that requires a new processing mode to enable improved decision-making, insight-making and process optimization [8]. More broadly, where it is impossible to collect, curate, interpret and model data utilising state-of-the-art data analysis methods or technologies, a dataset can be called as Big Data. In doing business, management and science, Big Data has totally modified the way its been applied. This data-intensive computation has appeared in the world and promises to have the resources needed to deal with Big Data issues. Data-intensive research appears as the fourth empirical model after the preceding three, namely empirical science, theoretical science, and computational science [1, 9]. It has arisen because we live in a world where data-intensive systems are increasingly used. Often it becomes Big Data because we have a huge number of dataset resources that are very complicated and difficult to process on-hand information processing tools. Laney [10] used volume, velocity and variety, defined as 3V's (Figure1.1), to define Big Data and claimed that another V might be extended according to the particular requirements of the data analytics domain. The fourth V can be value, veracity, volatility, or validity. Following are the fundamental characteristics of the dataset to be called Big Data:

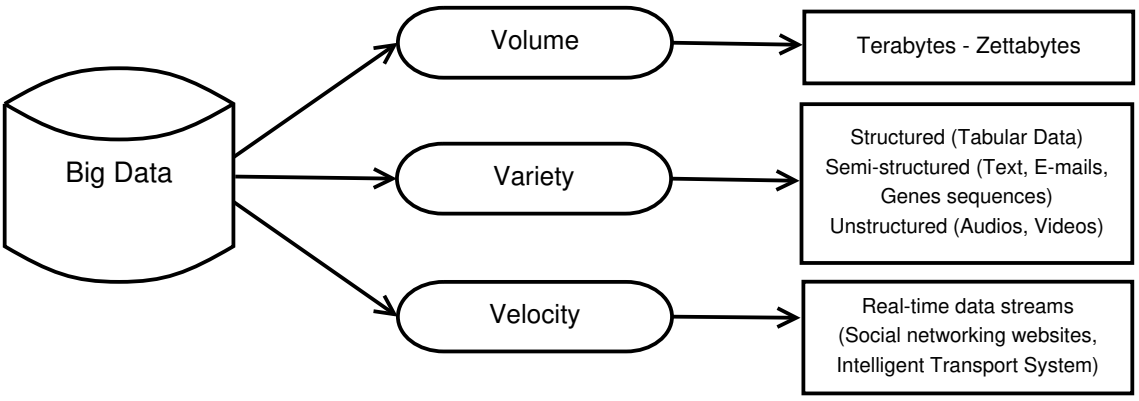



Figure 1.1: 3V's of Big Data

1. Volume - The amount of data that is generated is significant. It is the size of data, which determines the data values, and the potential size of the data underuse, and whether it can actually be considered Big Data or not. Entitled "Big Data," which itself contains a related feature size and, therefore, it is a term. When it is in the context of the size of data, the datasets range from petabytes to zettabytes.

2. Variety - Big Data's next feature is its diversity. Variety refers to both the various

forms of data and the various origins of data. It is possible to structure, unstructure and semi-structure. This variety of data available in the form of texts, audio, video, tabular and json format carries hidden knowledge and insights. This diversified variety in data creates issues for researchers and data scientists for storage, mining and study.

3. Velocity - Velocity refers to the velocity or rate of induced or processed data. Datasets expand exponentially with different data provisions, such as sensor networks, telescopes, science studies, high-throughput equipment, cell phones, social networks, and the internet. Big Data is often used in web-based applications; social computing hot spots is a suitable example of this case; it involves study of social networks, online forums, recommendation mechanisms, credibility systems, and markets for prediction. In addition, there are various sensors around us that create sum-less sensor information that needs to be used; for example, the Intelligent Transport System (ITS) is focused on vast quantities of complex sensor data.

4. Veracity: The biases, noise, and abnormality in data are referred to as Veracity. The data is reserved and extracted under the problem that is being analyzed. The reality is about the quality of the data, but as people are using Big Data, they are still committed and able to participate in optimally cleaning up data at the source.

5. Validity: When data has become accurate by correcting it following the intended use, it becomes valid data, and the term is referred to as validity. It plays a significant role in making the right decision as it guarantees the uncorrupted transmission of data.

6. Variability: The inconsistent flow of data with periodic event-triggered peaks is referred to as variability. The case gets more complicated when unstructured data involved.

7. Volatility: It refers to how long the data can be stored. It has to deal with the retention policy of structured data that we implement every day in our businesses. We will quickly destroy it until the preservation time expires. For instance, if an online e-commerce firm may not wish to retain a one year purchasing history for consumers. Since there is little chance that these data would ever be recovered after one year and default warranty expires on their items.

8. Value: As a consequence of extracting value from large data, it has a low-value density. Useful information and a large volume of data must be retrieved from every data type. For this, we need to check for the true value of knowledge in which the value of data must outweigh its expense or management. Therefore, priority needs

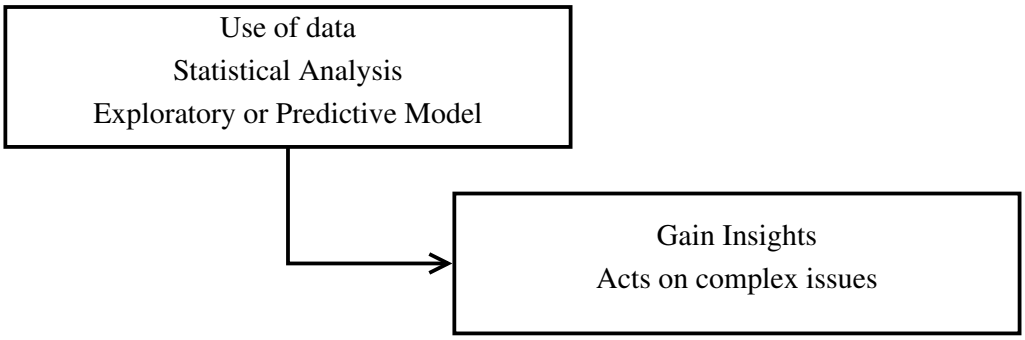



Figure 1.2: A brief overview of data analysis process

to be paid to the investment in data storage. At the time of acquisition, storage can be cost-effective and comparatively inexpensive, but such under-investment may destroy precious records. For example, for inexpensive and insecure storage, saving clinical trial data for a new medication will save money today and place data at risk tomorrow.

9. Visualization: Big Data's hard part makes all the massive volume of knowledge clear and easy to comprehend and read. In order to make data-driven decisions simpler, data visualisation allows data interpretation with the aid of graphical representation of the raw data so that it becomes understandable.

### 1.1.2 Big Data Analytics

Analytics is typically the method of finding and sharing useful data patterns, important in areas rich in historical knowledge. To measure efficiency, analytics depends on the simultaneous applications of statistics, computer programming, and operations analysis. In order to communicate insight, analytics also prefer data visualisation. The method of storing, arranging, and reviewing vast collections of data to find correlations and other valuable knowledge [11] applies to big data analytics. Big Data were analysed to discover secret patterns, unidentified associations and other important knowledge that can be used to make smarter decisions. Big data analytics can help companies better interpret the data found in the data and help determine the data most important to the corporate domain for future business decisions [12]. Figure 1.2 gives a brief overview of the data analysis process. Analysis of large data can be further divided into the following types:

1. Descriptive or Exploratory Analytics: It is also called a fact analysis and has the lowest complexity level as compared to another kind of data analytics. In this analysis, past events and performances, including successes and failures, are taken

into account to know what had happened.

2. Diagnostic Analytics: It is another kind of data analysis and has a more complex level than descriptive analytics. In this type of analysis, past events and performances, including successes and failures, are taken into account to know why that event has happened.

3. Predictive analytics: This kind of analysis uses historical information and facts to predict the future outcome. It has a higher complexity level than the previous two data analytics techniques and is used to know what will happen based on records. Predictive analysis is quite beneficial in several business domains to gain insight into future opportunities and risks.

4. Prescriptive Analytics: This type of analysis has the highest complexity level, and it makes use of factual information to recommend specific action about the situation that has already happened or will happen in the future. Several business domains and natural resource management organizations use this type of analysis to increase their productivity and take safety measures for upcoming natural disasters and climatic change.

### 1.1.3 Applications of Big Data Analytics

In order to increase productivity in industries and evolutionary breakthroughs in scientific sciences, big data is highly valuable; this gives more possibilities in many areas, such as:

#### 1.1.3.1 Big Data Analytics for bioinformatics and healthcare

The healthcare and human welfare industry is the sector of application where Big Data analytics is most important. This area shows some of the most significant and biggest datasets available. Seemingly, the global scale of clinical data in 2011 stood at about 150 exabytes with an expected growth of 1.2 to 2.4 exabytes annually [13]. Medical data consists mainly of, electronic medical records (EMR) and imaging data. Just a few categories of data exist for healthcare: prescription data, such as medication molecules and composition, drug goals, bio-molecular data and clinical studies, data on specific activities and interests, financial or action reports. Integrated data will include changes in intervention delivery, well-being and quality of life. According to McKinsey, healthcare analytics would add about $300 billion in revenue, annually per analyst forecast [14]. In reference to healthcare data analytics, the problem is how to combine heterogeneous databases situated in various areas of the world with differing access capacities. In addi-

tion to data consistency, timeliness, and data privacy and protection, a third major issue is the appropriateness of data sets.

#### 1.1.3.2 Big Data Analytics for scientific research and natural processes

As analytical sciences progress, many academic disciplines are turning into highly data-driven fields. The foundation for data-intensive scientific study is astronomy, computer science, computational biology, and others. Significant quantities of data are produced within these regions over time. Even on a single day, the Large Synoptic Survey Telescope (LSST) has 30 trillion bytes of data. Scientists utilise statistical methods and advanced analysis strategies to learn how the cosmos came to be. Likewise, 60 TB of data is produced per day by the Large Hadron Collider (LHC). In different academic areas such as oceanography, geology, genetics, and sociology, several potential e-science projects are arriving [15]. To discover knowledge from the databases generated effectively, a centralised repository and organised analysis method is needed.

Activities such as polar ice cap mapping, glaciers, and severe weather conditions involve collecting large databases from satellite imagery, atmospheric radars, and terrestrial surveillance and sensing equipment. A different collection of considerations are included in analysing insights from these datasets. Space-time scale, the magnitude of data and validation of long-term predictions focused on models are some of the most critical problems for these datasets [13]. Another barrier to data mining is a timely transfer of processed data into distributed networks for performing analysis activities.

#### 1.1.3.3 Big Data Analytics for business and economic systems

In a landscape of data, even huge companies have come across Big Data issues. It is projected that by 2020 the amount of market data would double every 1.2 years. Businesses handle large volumes of multi-modal data through costumer orders, order tracking, shop related video streams, advertising, consumer perceptions and feelings, revenue management infrastructure and financial reports [16]. For example, 267 million transactions happen per day inside 6,000 Walmart retail outlets across the world. To take advantage of machine learning strategies they use vast amounts of data to benefit about what people are involved in purchasing.

#### 1.1.3.4 Big Data Analytics for government and public sectors

Public government still needs to work with Big Data. In 2011, the Library of Congress received 3 Terabytes, or 3 trillion bytes, of records. Underneath the Obama presidency, the US government launched a big data research and development programme. This

programme requires the involvement of six independent agencies. Successes were also registered in Europe. Public agencies across the globe are deeply involved in proper public management. For instance, the annual health care fraud costs $60 billion to the US tax payers according to data gathered by the FBI [17]. Identifying tax avoidance and irregular purchases through electronic means presents the government with an ability to raise its tax revenue.According to a study by The McKinsey, Big Data functionality, such as reserving expertise and information habits, gives a public sector an ability to boost competitiveness and enhance total performance and effectiveness.

## 1.2 Machine Learning

Machine learning is the study of learning frameworks that incorporate principles and strategies from the fields of bothstatistics and mathematics. The area addresses the creation and the implementation of intelligent systems capable of learning from data without being directly programmed. The models are used to uncover secret patterns and developments in the data that contribute to meaningful observations and helpful in making data-driven decisions [18]. As the model learns from data and is able to execute activities from that data, the consistency and quantity of data accessible can dictate how much the model will learn. Machine learning approaches are frequently related to data processing, predictive data analysis and computer science processes, but on the actual ground these concepts vary in nature. Data mining relates to the practise of searching to see trends in databases and data warehouses [19]. Predictive analytics refers to the use of data in order to anticipate possible results and is used mainly in corporate environments to characterise practises such as revenue forecasting and anticipating the buying behaviour of a consumer. The word data science became common at the same period that big data appeared, data science covers how to derive values from big data in addition to processing big data in a number of ways [20]. The method of identifying patterns is identical no matter what since it uses the same methods and algorithms.

Machine learning techniques are broadly classified as unsupervised or supervised learning techniques depending on the learning process of the model (Figure 1.3). Classification and regression are supervised methods where the labelled instances are used to train the algorithm, and the outcomes are expected from the model. However, clustering and association analysis are unsupervised learning methods in which the unlabelled data is used by the learning algorithm to prepare the model. Supervised approaches involves classification and regression analysis, while unsupervised strategies include clustering and association analysis.

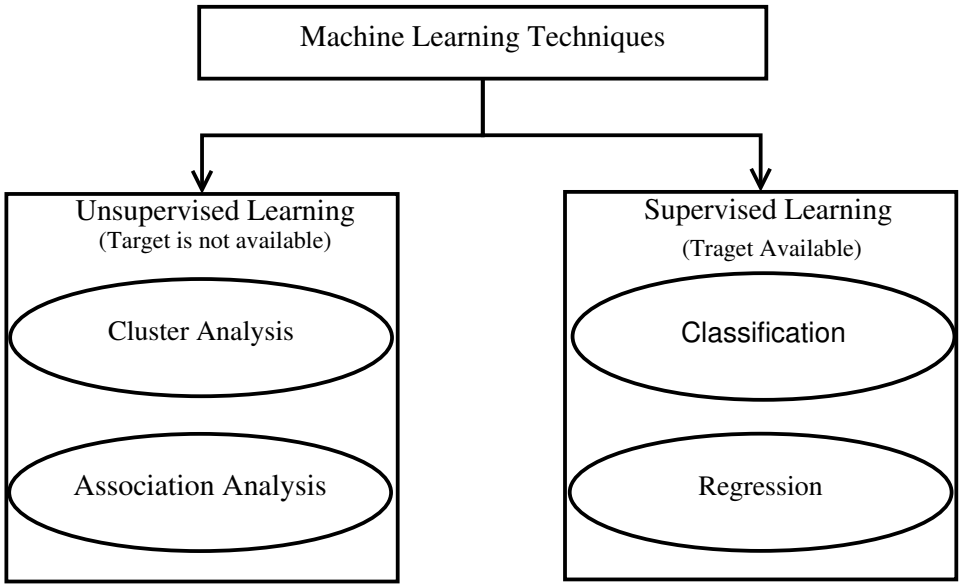



Figure 1.3: A broad categorization of Machine Learning Techniques

### 1.2.1 Supervised Learning

In the classification problems, where an algorithm is to be predicted, the inputs are introduced to the machine learning model. The goal is to predict the algorithm relative to the information. The classification assignment's objective is to achieve a categorical variable to indicate the target's class provided the input data. There are many descriptions of how computers might forecast the weather for a given day and decide potential values to be bright, windy, snowy, and gloomy. The input data can consist of weather data, such as the air pressure, temperature, humidity levels, peak or moderate wind speed, peak or average wind direction, etc. So, with any specified values for air pressure, temperature, relative humidity, and the day of the week, the model would inform us whether or not it's going to be bright, windy, snowy, or gloomy. Another widespread concern in the field of offences, particularly when it comes to credit cards, is the difficulty of discriminating between valid and fraudulent purchases [19]. A classification problem may be binary or multiclass. In a scenario weather forecast problem, there are more than two alternatives (sunny, windy, snowy, or cloudy). The outcome of the prediction problem is binary (sunny, windy, or cloudy). The issue with credit card purchases was that there are just two different categories in when and when an individual buys.

In the model, the predictors have to reveal a numeric meaning instead of only categories. This helps the model to predict a numeric value. An indication of regression is to be able to predict the price of a stock. Instead of having a set of addresses, a stock number is used. This is something like a regression task, or in other words, a sorting task instead of a classification task. In comparison to this argument, may be determining how much stock

would pay in the future, will be a classification problem. Provided the stock's actual price is a regression challenge, which refers to the significant discrepancy between estimating the stock's actual cost and past stock results. Some other examples of regression estimate the demand for a product based on the time of the year. Forecasting a score on a test, calculating how successful a drug will be for a particular patient, predicting the amount of rain for a region.

### 1.2.2 Unsupervised Learning

In clustering or cluster analysis, the aim is to arrange related objects into categories.Each data point in group may have a stronger similarity in the characteristics of the artefacts than the other cluster. To be able to organise the data into clusters, some method for grouping data is used. Many widely employed methods of similarity include Euclidean distance, Manhattan distance, and Cosine similarity. Clustering has a broad variety of uses ranging from market or customer segmentation to enterprise system planning [21]. One of the most common application of clustering is to split a client into categories depending on their shopping background and then to give them specialised deals based on that section. Any other forms of cluster analysis include characterising various weather trends within a city, grouping the current news stories into topics to recognise the trending topics of the day and identifying hot spots for different forms of crime from media records in order to have appropriate police presence for troubled areas.

The aim in association analysis is to establish a collection of rules that will help predict correlations between items or events. Special rules are used to decide when things occur concurrently. A common application of association analysis is market basket analysis is used to understand customer's shopping pattern. The supermarket chain discovered an association by using data analysis to examine the relation between two apparently different products. They noticed that many people who shop late on Sunday night to buy diapers often prefer to buy beer. This data was also used to put beer and diapers together and they noticed an increase in sales in both goods. Other applications of association analysis include suggesting related products to consumers based on search background, or buying behaviour. Look for clusters of products that are normally bought together and providing discounts on certain similar items at the same time to boost sales on both [20]. Identifying websites that are frequently viewed together, so that you can put up a consolidated list of similar websites.

## 1.3 Machine Learning techniques for data modeling

As various types of input data are collected, different models are used to solve complex problems. Earlier onwards, statisticians used statistical approaches in which they noticed that data mining was the creation of efficient statistical models from which the underlying distributions could be obtained. E.g., we've found a collection of data and a statistician found that the data comes from the Gaussian distribution that would be considered a data model. The researcher can then use different Gaussian parameters, such as mean, standard deviation, to represent the results' nature. Machine learning algorithms are the foundation for data mining. Modeling techniques like decision trees, secret Markov structures, etc. If deep learning tackles greater and more complicated challenges, the difficulty of concentrating on the important knowledge and discarding the remainder has become progressively meaningful. It is also regarded as function reduction. Function collection gauges the suitability of the samples for training and discarding samples that are unacceptable. The influential machine learning strategies are discussed as below:

### 1.3.1 Decision Tree

The rationale behind the decision tree's methodology is to divide data into categories that apply to only one class. The method divides the input space into pure regions where only samples from one class exists. With real data, however, certain subsets are not possible. The goal is to divide the data into subsets that have as much purity as possible. Each subset contains as many samples as possible from a single class. It is best performed by graphing the input values and identifying regions where the values are as pure as possible. These regions are separated by boundaries called as decision boundaries. The decision tree classifies the at on the behalf of these decision boundaries [22]. A decision tree is a hierarchical tree structure with nodes and directed edges. The root node is the first node in the path. The leaf nodes are at the bottom-most nodes of the tree. Nodes in between the root node and the leaf nodes are called internal nodes. For each root and internal node, there is a test condition and a class label. An intelligent decision is made by utilising the tree to find the best action. At each node, the answer to the test condition determines the route to take. When a leaf node is reached, the category at the leaf node will determine the classification decision. The depth of a node is the number of levels from the root node and other nodes. The depth of the root node is zero. The depth of a decision tree is the number of parent and child nodes in the path between the root and the leaf. The number of nodes in a decision tree is the size of the tree.

At a high level, it involves some basic decision making steps. All samples are brought together at the beginning, and then are separated by the variables being tested. It aims to create a subset of samples which contain all of the samples that belong to just one class. The subsets should be as homogeneous as possible or as pure as possible. A process for subsampling data into progressively more pure subsets is iterative until the stopping criterion is met. For the purpose of the decision tree model, an induction algorithm is utilised [23]. A greedy algorithm is used in the division of the specified portion of the data. Greedy algorithms are applied to solve a portion of the problem at a time, while more comprehensive approaches are applied when the approach is complete. The tree is built by finding the optimal method to partition the current node at every step and merging these decisions with each other to form the final decision tree. The best split is determined from a set of possible splits based on the impurity measures such as Gini Index and Entropy which compare subsets of the resulting mixture. Lower the value of these impurity measures, more homogenous would be the class labels in the constructed tree.

### 1.3.2 Support Vector Machine

SVM can also be used for all classification and regression purposes. Support vector machine algorithms are ideal for linear and non-linear data classification. SVM generates a collection of hyperplanes in a high dimensional space for classification or regression purposes. The hyperplane with the largest operating margin from the training data is used as the classifier's final hyperplane. The more significant, margin lower will be the classifier's generalisation error. Transforms feature vectors through high-dimensional space features via a kernel element [24]. The SVM models depict the samples as points in the space and map them such that simple gaps can be found between various classifications. The latest research sample is expected dependent on the same representation. SVM's are widely utilised by researchers for image classification and text categorisation.

### 1.3.3 Bayesian Networks

Bayesian belief networks are guided acyclic graphs of information. Therefore, they are called Bayesian probabilistic machine learning models [25]. For purposes like classifying diseases and monitoring signs, Bayesian classification may be used. If we know the symptoms of an illness, then we can predict the risk of anyone developing the disease. Bayesian networks are used everywhere, in areas like computational biology, bio-informatics, information retrieval, semantic search, etc.

### 1.3.4 Artificial Neural Networks

Artificial neural network algorithms are widely regarded as neural networks. Neural network algorithms and structure rely on the way the human brain functions. A neural network-based computational model that involves several processing units that are connected across a communication network. Processing units are classified as neurons that function together to produce the required output. The theory functions like the way trillions of nerve cells, or synapses, function within a human brain. Neural network algorithms are typically chosen because the interaction between input features and output variables will express itself as a complex network. These subject knowledge are well developed in the area, like in speech recognition, image analysis, biology and etc. [26].

### 1.3.5 Hierarchical Clustering Techniques

Hierarchical clustering combines data objects into groups that, within such groups, integrate into broader groups, and so forth, forming a hierarchy. A tree demonstrating this hierarchical pattern is known as a dendrogram. Specific data artefacts are the leaves of the (graph) tree, and the nodes inside the clusters are nonempty. Sibling nodes on the same branch create a partition. This allows practitioners to see data at various levels of granularity. Hierarchical clustering strategies are grouped into two types: agglomerative and divisive [27]. An agglomerative strategy begins with a single-point cluster, and the most related clusters are then combined [28]. A divisive clustering begins by collecting all the data and breaking the data into a series of clusters. Clustering persists until a stopping criterion (most commonly, the requested amount k of clusters) is met [29].

### 1.3.6 Partition Relocation Clustering Techniques

Unlike the conventional hierarchical approaches, the partition relocation algorithms are able to boost clusters when new points are introduced. When suitable data are used, results are good quality clusters [27]. Iterative distribution strategies require replacing points in each of the k clusters. One way to divide data is to think of a cluster of data as a certain model whose undefined parameters have to be defined. Different mathematical methods presume that the data come from various populations whose distributions we choose to describe. One benefit of probabilistic clustering is that the effects can be easily interpreted. Getting cluster representations that are concise often makes for inexpensive calculation of intra-cluster metrics of fit as well as the derivation of a global objective function. Through evaluating a partition, an objective function can be described. Pairwise

distances or correlations may be used to distinguish clusters inside or between clusters. In the iterative improvement method, there will be so many pairwise similarities. The k-means and k-medoids also use the idea of special, distinct cluster members. The computation of an objective function is now linear with respect to the number of clusters, and the number of clusters is less than N.

### 1.3.7 Density Based Partitioning Clustering Techniques

An open set in space may be separated into its related elements, or points. The application of this concept involves a diversity of ideas in fields of consistency, density, accessibility and boundaries. Definitions of these terms are in near proximity to one another. A cluster, described as a linked dense part, grows in any direction that the density leads. Therefore, shape-based clustering algorithms can detect clusters with arbitrary shapes [30] [31]. Often, these characteristics offer the best defence against outliers. A graph reveals certain cluster shapes that pose problems for k-means, but can be mitigated by density-based algorithms. Density-based methods are modular. Inconvenient premises go together with the excellent characteristics. Since a single dense cluster composed of two neighbouring areas with substantially different densities is not very insightful, I recommend collecting height data from several places [27]. Another shortcoming is that the video loses interpretability.. There are two forms of density-based techniques. The first solution utilises one point of data to evaluate the density. Algorithms reflect such items as: DBSCAN, GDBSCAN, OPTICS and DBCLASD. The second method explores density in a basic, linear manner and is clarified in the article . DENCLUE is a linear classifier that is not negatively influenced by data dimensionality.

## 1.4 Machine Learning for Big Data Analytics

The rise in the usage of technology such as wireless sensor networks, internet of things, intelligent transport system, high throughput science experiments, social networking, and e-commerce web-sites has provided hint of generation of colossal volumes of data worldwide [32]. The tremendous volume of data that is being processed, preserved and evaluated presents the major obstacle to the information and technology sector to gain benefits. With the emergence of this data specific age the mathematical intelligent machine learning systems came into limelight over the past decade in large number of complex data-intensive fields such as astronomy, particle physics, material sciences, agriculture, bio-informatics, finance and economy. And as a consequence, these machine learning frameworks are not ideal for processing massive volumes of data. One inherent prop-

erty is that most non-parametric and model-free methods, when scaled to deal with high-dimensional results, are found to be computationally costly [33]. The need for development of more effective and reliable machine learning techniques to enhance energy modelling for data-intensive areas.

### 1.4.1 Ensemble Machine Learning Methods

Ensemble learning methods blend several base models to yield a more effective and accurate approach for making predictions. The ensemble approaches have become extremely common because they do a better job than the individual models. Ensembles techniques are widely utilised at several weather prediction facilities around the globe, such as the National Center for Weather Prediction, the U.S., European Centre for Medium-Range Weather Forecasting, United Kingdom Met Office, Metro France, Climate Canada, Japanese Meteorological Service, and so forth [33]. Ensemble learning also acts as an important technique in machine learning. It utilises several classifiers and optimises the output of the base classifiers separately. While it does not always work optimally, a technique with various classifiers produces a stronger outcome than a single classifier. Choosing a particular collection of laws, including the plurality vote strategy, aids in scrutinising the possibility of producing bad outcomes from a single model.

Any of the three methods widely used to integrate various machine learning models to combine the models into one predictive model are as follows:

1. Bagging: It is based on bootstrapping. In this approach, multiple models are created by using the random sub-samples of the training data and integrating multiple homogenous models, i.e. the models that are all of the same kind. A sample is randomly produced and then pulled to create a new sample. Decreasing the scale of the data used in a predictive model decreases the uncertainty in the expected result. When decision trees, neural networks, etc., are unpredictable, minor variations in the training dataset will influence how the classifier performs.

2. Boosting: Boosting is an iterative procedure that guarantees a weight is modified after each observation is classified. If an observation was wrongly categorised in a certain field, it seeks to raise this observation's degree of significance. The method boosts poor or untrained models to stronger and better predictive models. In this method, an ensemble of models is developed using subsets of the training data and their mixture results in an average model. However, unlike the bagging that happened, the sub-samples were not uniformly dispersed. The next sub-sample generation depends on the previous models' results, i.e. it includes the records

misclassified by the previous model. The training data sub-sample generates a set of average value models, and then merge them using a particular cost feature like majority voting to improve further models. The biggest benefit of boosting is to increase accuracy. An example of an ensemble technique using this method is AdaBoost.

3. Stacking: In this technique, many process are clustered to get desired outcomes. The combining process is unique because the output of a classifier of Level (N-1) would be compared with the output of Level N's classifiers to approximate the goal function. Either prejudice or variance error is minimised when mixing learners are being used.

### 1.4.2 Scalable Machine Learning Methods

Machine learning relates to the usage of machines to learn from past or current results. The usage of machine learning algorithms to capture and gather data has driven its popularity in science and business applications. According to the Digital Universe report, the world's data by 2020 would be ten times greater than in 2013. That would be 44 zettabytes or 44 trillion gigabytes. To this degree, there is no one entity struggling with data; numerous organisations often produce data that is too large to be analysed easily through traditional techniques. With the growing amount of data, the machine learning group must learn from large data and how best to analyse and utilise it for applications. Famous machine learning frameworks like R and Weka were not initially developed to operate at these kinds of scale. The increased need to process data has caused them to revisit the architecture of machine learning systems. A successful way of coping with large data in machine learning is to operate parallel algorithms. Data parallelism is usually implemented in two ways: either the data is partitioned into more workable bits. Each subset is calculated separated into smaller chunks to allow concurrent task execution. Apache Hadoop an open source implementation of most popular Big Data processing framework MapReduce is helpful in solving scalability problem by offering distributed storage and processing. The other popular data processing engines for Big Data like Spark, Flink, Storm and $H_2O$ also have the machine learning frameworks for scalable machine learning. Few of the Big Data machine learning frameworks associated with these data processing engines are as follow:

1. Mahout: Apache Mahout is an open-source project mainly used for developing scalable machine learning algorithms that can be used in data mining projects. Mahout's core algorithms involve sorting, clustering, and collective filtering, which are successful for broad data sets. The methods include hardcoded subjects, di-

mensionality reduction, vectorization, resemblance tests, algebra, and more. One of Mahout's biggest cited assets is its versatility. It is noted for providing a large range of algorithms and successful implementations, but with long and insufficient runtimes due to the slow MapReduce engine. Mahout's most cited strengths are its elasticity, and others have found performance utilising the baseline algorithms. Mahout 0.10 was launched in summer of 2015, which aims to expand upon previous models. The update emphasises Samsara's math setting, with emphasis on linear algebra and statistical operations. The Mahout-Samsara project aims to provide the consumer with the resources to build their own algorithms, rather than simply offering pre-written implementations. Several experiments have applied Mahout-based mathematical learning approaches for functional machine learning. Honeywell [34], for instance, has its own cloud infrastructure platform where HBase, Mahout, and other analytics technologies are leveraged. They used Mahout's Random Forest and Naive Bayes' principles and algorithms to forecast auxiliary power unit outages and breakdowns. This system was able to increase the chance of executing an auto-shutoff by more than three. Using Mahout in the automotive sector produces strong results. However, it is noteworthy cases of huge data companies like Mendeley [2], and LinkedIn [35] that, as part of their big data communities, use their suggestion tools. Overstock substituted a commercial unit, bringing back on prices by a considerable sum.

2. MLlib: MLlib covers the same set of learning tasks as Mahout does, plus adds regression models that Mahout does not provide. They validated and published algorithms for topic modelling and routine pattern mining. Additional knowledge retrieval techniques require reduction of dimensions, growth characteristics, transformation and simple statistics. In general, MLlib's reliance on Spark, streaming operations and iterative batches, and in-memory computing allow it to run far faster than Mahout. Zheng and Dagnino [36] noticed that the extension of the existing algorithm or the writing of a parallel version is relatively straightforward. These support various algorithms, including Decision Trees, Random Forest, Naive Bayes, and Logistic Regression. Cluster design algorithms include k-means, Gaussian mixture, and clustering. For online research, algorithms like Linear Regression, Isotonic Regression, Alternating Least Squares and so forth are used. Streaming Logistic Regression, Linear Regression, and K-Means Clustering methods are used for online research. Internet forecasting algorithms can be learned offline from past data and applied to current streaming data. Internet forecasting algorithms may be known offline from past data and spread to new streaming data. Building sophisticated machine learning pipelines will prove difficult at times. In version 1.2, Spark Ma-

chine Learning, was implemented to resolve these problems, enabling many machine learning algorithms into one workflow. This kit contains approaches for processing and optimising algorithm. The pipeline defines a sequence of changes affecting the dataset. One illustration is studying how to transform a data frame into a format that provides forecasts and functionality. This package is structured to assist users in importing data from a source, preliminary function extraction, model training and assessment. This package is structured to help users import data from an authority, initial function extraction, model training and evaluation.

3. $H_2O$: This tool is noteworthy for its graphical user interface (GUI), deep learning applications, and other features. With Java, Python, R, and Scala, programming is possible with $H_2O$ . College students, who are not acquainted with programming, also use this function. Thanks to the availability of several pre-tuned configurations, it is easy to set up, needing less learning curve than any other free available solutions. Numerous activities, including sorting, clustering, generalized linear models, predictive analysis, assemblies, optimization approaches, data preprocessing, and deep neural networks, are covered by the machine learning assistance offered. Other algorithms and instruments that conform to the standards and requirements exist, but are still in the production stages. For incorporation of Spark and MLlib it has an API called Sparkling water[2].

4. SAMOA: SAMOA started as a project within Yahoo in 2013 and was taken into the Apache Incubator in 2014. Huge Online Analysis of Scalable Advanced applies to the study. This software framework can be run locally or on a few stream processing engines, including Storm, Samza, S4. Using a minimal API, users can easily write bindings that build new stream processors on top of the SAMOA port for a generally distributed stream processing engine. SAMOA's algorithms are described as directed, tree-like structures, or as directed graphs [37]. For sorting, clustering, regression, regular template mining and boosting, bagging, and ensemble forming, established algorithms can be exploited. Several standard protocols for streaming are usable and a platform for the consumer to write their custom distributed streaming algorithms [38]. This model is planned for large-scale database consumers with regular upgrades. Streaming frameworks are commonly utilized in programmes that aim to discover current patterns, although input happens in real-time. They have CluStream for cluster forming, and the Vertical Hoeffding Tree for grouping, which utilizes vertical parallelism on top of the Very Quick Decision Tree, or Hoeffding Tree. Many streaming classifiers are achieved using a decision tree. The Adaptive Model Rules Regressor may be achieved by regression on several

real-world instances. When using DSSFIM, PARMA is included inside the library, which is focused on DSSFIM [39]. Post-hoc measurement is also possible and helps one evaluate model consistency either from the beginning or over time. To construct classifiers for a collection of data, Bagging, Adaptive Bagging, and Boosting are used. An programme named SAMOCT [40]. Additional learning algorithms are required for the MOA clustering algorithm, which allows using the algorithm in the SAMOA platform.

## 1.5 Thesis Organization

**Chapter 1:** This chapter carries the introductory discussion on the concepts of big data, big data analytics, and machine learning techniques for data modeling and data analytics. A brief introduction of several fields producing big data and using data analytics techniques is also provided in this chapter.

**Chapter 2:** This chapter presents the exhaustive literature survey on the thesis topic. It carries a detailed description of the work done, current scenarios, significant data analytics trends, and machine learning for big data. The chapter covers a detailed study of work done and advancements in common BDA areas such as cloud computing for BDA, map-reduce paradigm, and fuzzy-based BDA. Further, this chapter covers the study of machine learning approaches and platforms for BDA.

**Chapter 3:** This chapter is a case study on the classification of the type of Human T-cell Lymphotropic Virus (HTLV), one of the prominent viruses for a life-threatening disease like adult T-cell leukemia. 64 hybrid machine learning techniques have been proposed for predicting the type of HTLV from protein sequences of different human viruses (which is semi-structured data). This chapter describes the proposed techniques, their implementation details, model evaluation, and comparison. Further, the best predictive technique out of the proposed techniques is also identified in this work.

**Chapter 4:**This chapter is a case study on predicting the kinematic viscosity of nano-lubricants employed in heavy earth-moving vehicles. The historical data of three grades of lubricants, namely engine oil, gear oil, and hydraulic oil with the composition of $Al_2O_3$ and$CeO_2$ nano-particles, has been used in this work. Multi-criteria decision-making (MCDM) machine learning techniques have been proposed to predict nano-lubricant kinematic viscosity in each category.

**Chapter 5:** This chapter is a case study on searching the exotic particles in particle physics Big Data generated from the Large Hadron Collider (LHC). The data generated

is enormous in volume. In this work, a scalable multilevel ensemble (MLE) classification technique has been proposed for classifying the exotic particles and non-exotic particles from the massive volume of data. This chapter describes the proposed technique, its implementation details, model evaluation, and comparison with existing techniques.

**Chapter 6:** This chapter concludes the thesis and lists the possible future research directions.

## 1.6 Thesis Contribution

In this thesis, an attempt has been made to solve the Big Data predictive-analytics problems using hybrid and ensemble machine learning methods. The main contribution of the thesis is as follows:

1. A comprehensive study of current status and trends of machine learning techniques and platforms for Big Data is carried which includes cloud computing environment for BDA, the map-reduce paradigm for processing big data, and use of fuzzy logic in BDA.

2. Hybrid machine learning techniques and multilevel ensemble classification technique have been proposed for semi-structured data and high volume structured data, respectively. The hybrid ML techniques include feature extraction, unsupervised learning, feature weighting, optimal feature selection, and supervised learning methods. On the other hand, the multilevel ensemble technique is the combination of the different classifiers in a multi-tier fashion to improve the process's overall efficiency.

3. Different V's of Big Data like volume, variety, veracity, and value are considered in this work. For example, the protein sequences of HTLVs and other human viruses, semi-structured data, are downloaded from bio-informatics benchmark database Uniprot. On the other hand, high volume particle physics data of the Large Hadron Collider (LHC) is downloaded from the UCI machine learning repository.

4. For evaluating the efficiency of the proposed hybrid ML techniques are implemented on a case study from bio-informatics domain, i.e., predicting the type of HTLV from the protein sequences of different human viruses.

5. The efficiency of the multilevel ensemble classification technique has been evaluated on the case study from the particle physics domain. The data set involved in testing the efficiency of this technique is high in volume. A cloud-based platform offered by Amazon web services (AWS) and Databricks Spark data analytics services is used

for implementing the technique. This technique's scalability has been evaluated by increasing the number of Amazon EC2 instances on the AWS cloud platform.

6. Multi-criteria decision-making-based machine learning techniques have been used to predict the kinematic viscosity of different categories of nano-lubricants employed in heavy earthmoving machinery. The historical experimental data of nano-lubricants has been used to build a machine learning-based system that can skip the need for repeated experimentation to predict the lubricant's kinematic viscosity with the particular composition of nano-particles.

7. The experimental results obtained for all three works show the efficiency and usefulness of the hybrid machine learning techniques, multilevel ensemble classifier, and MCDM machine learning techniques for bio-informatics, particle physics, and nano-lubricants domains, respectively. The experimental results for proposed hybrid techniques have shown their usefulness in assisting the traditional confirmatory test for adult T-cell leukemia. On the contrary, the MLE classifier has efficient scalability, greater accuracy, and fast processing in dealing with a high LHC data volume. Simultaneously, MCDM machine learning techniques are the first state of artwork to predict nano-lubricants' kinematic viscosity from historical experimental data generated.

# Chapter 2

# Literature Review

This Section reviews the research work of various researchers in big data analytics and machine learning. Various essential research contributions in the field of machine learning, ensemble and hybrid machine learning are presented in detail.

## 2.1 Big Data Analytics

When learning from data is involved, Big Data has been one of the emerging topics and has attracted tremendous interest from academics in knowledge sciences, policy, and decision-makers in governments and companies[41]. It is one of Gartner's present and potential study frontiers in the top 10 strategic technology developments for 2013 and the top 10 essential technology trends for the next five years [8]. The various researchers have given different definitions for Big Data from 3V's to 7 V's [42]. They presented various problems when discussing Big Data concerns in data collection, storage, security, retrieval, interpretation and visualisation [43].

While some of the researchers identified that the current hardware and software platforms are incapable of handling Big Data due to its huge size and diversified data sources, [13] [41]. They demand some technological change in current hardware and software stacks for efficiently dealing with Big Data problems [44] [45]. On the other side, several scholars say that cloud infrastructures are a powerful option for computing and processing large data applications [44].

In contrast to this, few authors have addressed the variety of challenges for processing Big Data in cloud computing environments, such as suitability of cloud platform for current Big Data analytics tools, transfer of massive data on cloud infrastructure, data security, and legal issues [13] [46]. Then some authors have given the survey on different tools and techniques being developed and implemented on Big Data applications and their advantages, and limitations [1]. They have discussed the popular MapReduce paradigm and its open-source implementation Hadoop.

While some authors have criticized MapReduce as it is an index and schema-free [47]. Others have addressed that Hadoop is suitable for batch analysis applications while it

is not much efficient for stream analysis and interactive analysis applications [1]. Some authors have done optimization on MapReduce and Hadoop to achieve better efficiency, which is purely domain-specific [44]. Some authors have identified that a cloud IoT environment is more suitable for Big Data applications.

However, some authors have identified that classic artificial-intelligence (AI) techniques like machine learning algorithms and artificial neural networks (ANNs) consume large memory and time to perform common statistical operations like classification, clustering, and regression when it comes to large-scale datasets [48]. They identified the need to scale up traditional machine learning algorithms [49] and implement parallel artificial neural networks. With the idea of implementing parallel ANNs, they have also identified the limitations of large memory consumption and training time. Some authors have concluded that Fuzzy and Big Data clustering techniques are more efficient than traditional clustering methods and even better if merged [50] [51]. The following sub-sections address studies in the field of Big Data Analytics.:

### 2.1.1 Cloud based BDA

Assuncao et al. [52] discussed Cloud analysis methods and environments for Large Data systems across four main fields of analysis and Big Data, including data processing and design help, model creation and scoring, simulation and user engagement, and finally market models. They also listed emerging technological challenges and proposed possible directions for cloud-based Big Data storage and analytics technologies. Domenico et al. [53] identified cloud computing infrastructure is efficient for the computational and storage needs of Big Data Analytics. They have also discussed cloud services' suitability, such as IaaS, SaaS, and PaaS, for easy and timely Big Data Analytics and storage. Inukollu et al.[46] discussed security concerns relevant to Big Data in cloud computing. Various security loops linked to the MapReduce paradigm and Hadoop are also described. Possible solutions for handling those security breaches have been proposed.

Ji et al. [44] addressed the critical issues regarding Big Data management and processing in a cloud environment, cloud infrastructure platform, software architecture, cloud storage, and online database schemes, for example. Architecture of Big Data Processing Systems from the Distributed File System, Unstructured and Semi-Structured Data Storage and Open Source Cloud Technologies is explored and given a summary of many optimizations of the MapReduce programming model for large-scale distributed systems.

Agrawal et al. [54], and others have described many obstacles to the effectiveness of data management systems in the Cloud. Design issues for designing and integrating modern

technologies for a seamless transfer of software from conventional business architecture to next-generation cloud infrastructure are addressed. Hashem et al. [55] reviewed the rise of Big Data in cloud computing. The authors have discussed the relationship between Big Data, cloud computing, data storage and Hadoop. The research challenges are based on the size, affordability, consistency, credibility, transformation, and heterogeneity of the data are also investigated. Chang et al. [56] have demonstrated that the Big Data framework implemented for biomedical sciences was quicker on Cloud relative to non-Cloud systems. Previous research determined that network latency, file size, and job failures would affect efficiency, so the tests were carried out to explore their consequences. Organizational Sustainability Modelling (OSM) is used for proper comparisons, with targets in mind. OSM provides the current and predicted results and reveals the variations between the Cloud and non Cloud settings.

### 2.1.2 MapReduce paradigm for BDA

Dean et al. [57] authored an article on the MapReduce programming model's context and how it is integrated into a cluster-based infrastructure. A discussion of MapReduce's output on various tasks is provided. The usage of Google's MapReduce has also been discussed. Lee et al. [47] published a survey on the MapReduce system's specifics, including the framework's respective design. The MapReduce system's advantages and drawbacks are addressed, and practical solutions for its output tuning are given. Ranger et al. [58] created an API and a runtime for MapReduce called Phoenix, which ran on shared memory systems. Its performance potentials and error recovery features were tested on multi-core and symmetric multiprocessor systems. Yang et al. [58], suggested a new model based on MapReduce called Map-Reduce-Merge that introduces a merge step after reducing phases that integrates two reduced outputs from two different MR jobs into one, which can effectively merge data that is already portioned and sorted by the MapReduce modules. As proposed by Nykiel et al. [59], a distribution method with MapReduce named MRshare converts a batch of queries into a new set that can be done more efficiently by grouping workers into groups and analysing each category as a single query.

Dittrich et al. [60] proposed a new variant of Hadoop called Hadoop++ by injecting their new technologies via UDFs in Hadoop without modifying Hadoop. Their findings indicate the dominance of their method on indexing and manipulating joint tasks. Wang et al. [61] suggested a MapReduce-based approach called MapDup-Reduce that can identify close duplicates through broad datasets. Cordeiro et al. [62] suggested the Best of All Worlds (BoB) clustering approach for multidimensional datasets, which is a hard

clustering method that enables the automated and complicated trade-off between disc latency and network delay. Zhou et al. [63] suggest a parallel MapReduce dependent K-mean algorithm. The algorithm runs on Hadoop, and it is more efficient and reliable in large-scale automated text classification. Li et al. [64] has suggested an efficient K-mean algorithm utilising ensemble learning process bagging and distributed computing system MapReduce as an alternate solution to cluster databases.

Collins et al. [65] carried the SWOT analysis of Big Data Analytics in domains like health economics, epidemiology and public health. O'Driscoll et al. [66] provided an overview of the use of Big Data technology, especially Hadoop and cloud computing in biology's Big Data set (genomic data sets), constitutes data warehouses in petabytes exabytes. Gani et al. [67] has provided the taxonomy of indexing techniques for extracting and handling Big Data and categorised indexing techniques based on their approaches, such as non-artificial intelligence, artificial intelligence, and collective synthetic intelligence indexing methods. The study evaluates the importance of various styles and addresses the efficiency drawbacks and advantages of each methodology. Gandomi et al. [11] proposed that an effective and efficient analytics approach is required to process the large amounts of heterogeneous unstructured datasets and pointed out the need to create computationally efficient algorithms for heterogeneous, noisy, and vast structured data to prevent Big Data pitfalls. Mohamed et al. [68] introduced real-time analytics and its opportunities in areas such as relief operations, battlefield strategies, decision making, and financial operations.

## 2.2 Machine Learning trends for Big Data

Machine learning approaches are recorded to outperform most of the physical and computational methods in predictive modelling in terms of precision, robustness, ambiguity analysis, data performance, simplicity, and computation expense [69] [70]. Machine learning techniques have achieved tremendous traction in the last few years with implementations in a large variety of fields. Any of the most popular machine learning approaches include artificial neural networks, decision trees, support vector machines, Bayesians, neuro-fuzzy and wavelet neural networks. Neural network strategies and fuzzy propositional networks have been developed as a single intelligent algorithm [71]. The ML approaches are continually changing to more efficient simulation methods. The hybrid and ensemble approach also outperform single machine learning algorithms. Ensemble and hybrid machine learning techniques are the two most widely employed machine learning strategies. The prospect of ML rests in the development of novel ensemble and hybrid computing strategies. This segment covers the literature of recent developments

in machine learning, including many innovative implementations.

Ensemble approaches and hybrids are outperforming their single machine learning predecessors with their precision and reliability. One method consists of combining the hybridization of machine learning techniques with two or more ML techniques or/and another soft computation, heuristic and meta-heuristic techniques, eventually contributing to a more robust final process. On the other side, where the ensemble aims to build the classifier using bagging, boosting or stacking, this contributes to more than one classifier being trained. This hybrid and ensemble frameworks will result in improved ML models in the future. Some of these high performing ensemble and hybrid techniques are discussed in this literature.

### 2.2.1 Fuzzy based machine learning algorithms for BDA

Fernandez et al. [72] discussed problems relating to disseminating data and parallelizing existing algorithms, and their association with data classification and representation. A detailed review of the critical frameworks on the subject and how they apply to fuzzy sets have been presented. Ayed et al. [50] briefly listed k-means algorithm, agglomerative clustering process, fuzzy clustering process, and Big Data clustering methods, and also explored an idea to create a scalable and noise responsive Big Data clustering framework. Ludwig [73] has built a parallelized variant of the fuzzy C-means algorithm and illustrated the implementation of the map and reduce primitives. A study of validity, scalability, and efficiency is performed to demonstrate the project's consistency and effectiveness in large numbers. Garg et al. [74] evaluated performance of fuzzy k-means clustering algorithms in MapReduce for separate datasets. They implemented fuzzy K-means algorithm on a Hadoop cluster developed on the AWS server (EC2) to evaluate performance for various datasets work. Ananthi et al. [75] has implemented a modern form of clustering algorithm by reducing the time complexity and errors by the use of interval-valued intuitionistic fuzzy sets. The approach's usefulness is shown by contrasting it with other algorithms like intuitionistic fuzzy C-means, fuzzy C-means type-2, and the K-means and fuzzy C-means algorithms. Wang et al. [76] have suggested a clustering approach in which fuzzy data are used to describe linguistic data. Then a fuzzy-compatible relation is constructed to reflect the relationship between two linguistic data. Finally, a fuzzy equivalence relation is extracted from the consistent fuzzy link, as the max-min operation. The linguistic data sequences belonging to the fuzzy equivalence relation were conveniently grouped into clusters.

Lopez et al. [77] suggest a linguistic ambiguous rule-based classification scheme that further utilises the Chi-FRBC-BigDataCS algorithm on top of the MapReduce paradigm

to fix imbalanced Big Data. Using the MapReduce structure, this approach spreads the fuzzy model's data operations and integrates cost-sensitive learning strategies to overcome the variance in the data. After discussing the definition and computational models, Mukkamala et al. [78] outlined a formal model focused on fuzzy set theory, and then provided the organisational semantics of the particular framework with a Facebook illustration of social data in the real world. The Social Data Analytics Tool (SODATO), which is mostly used to collect social data from the Facebook wall of a premium brand, H&M. Briefly examined the Social Data Analytics Tool (SODATO) and conducted a sentiment classification of comments and posts and then evaluated the type by creating crisp and fuzzy artefact sets (posts, comments, likes, and shares). Eventually, they discussed the analytical process and concluded the advantages of set-theoretical methods focused on associational sociology's social, conceptual methodology.

### 2.2.2 Ensemble machine learning algorithms for BDA

The Ensemble methodology is used to improve the precision of the model's predicted values by integrating multiple ML algorithms. Researchers around the world are implementing innovative, adaptive machine learning for prediction and analytics. Dietterich et al. [79] methodologies like ensemble methods like Adaboost and Random Forest were used. It is discovered that ensemble approaches are more efficient than single classifiers in accuracy. Street et al. [80] revealed an algorithm to identify streaming data in real-time. Similar block of data is used with different classifiers to boost accuracy. Feng et al. [81] developed an algorithm based on Online Accuracy Updated Emsemle (OAUE) to process big data sets. The solution tackles the problem of short drift in large data sources. Krawczyk et al. [82] suggested an integration-based nearest neighbour rule-based ensemble approach for implementing analytics on large datasets. Classifiers performance were incorporated via majority voting to speed up the computation. Jia et al. [83] developed a gender classifier utilizing four million face photographs and more than 60,000 facial features. Combining different classifications was used to enhance the precision of facial recognition.

Haque et al. [84] established a multi-tiered ensemble-based approach to manage large amounts when classifying the significant data sources. Three comprehensive Adaboost experiments were conducted with Map Reduce-based processing to achieve scalability and speed up the data analysis technique. Pui et al. [85] proposed a space vector machine algorithm that uses SVMs to distinguish extensive data sets. The findings revealed that greater precision of high-dimensional data could be obtained by integrating several classifiers to make an ensemble. Cuzzocrea et al. [86] have proposed an ensemble

classifier system for addressing big data workload classification and categorization issues in the cloud. The system is used to assign workload and forecast the workload for the next period. Ensemble Extreme Learning machine (ELM) proposed by Wang et al. [87] registered an improvement in training pace of up to 4.6 times and a decrease in test errors by 19%. Diversity among various classifiers is essential to the ensemble building process and has become the focal point for numerous studies. Several different ensemble integration types have been suggested, but no one method is yet defined to be the strongest. Various methods utilized in preserving the biological diversity of the ensemble have been analyzed in this presentation. The ensemble approach allows for several classifiers to be combined tactfully. The homogeneous ensemble combines multiple classifiers on the same training dataset, while heterogeneous ensemble utilizes a single learning system using different sub-samples of training results.

Gorczyca [88] modelled a Trauma Intensity Risk Estimation using Ensemble Machine Learning. The research contrasted the methodology of this analysis with the Harbor View Assessment in terms of precision, F-score ratings, mortality risk, the Bayesian Logistic Injury Severity Score, and the Trauma Mortality Prediction Model. The suggested Ensemble Approach may have yielded stronger outcomes than the currently chosen framework. Results have shown that trauma has a dramatic effect on the likelihood of this undertaking. Wang et al. [89] built a decision tree model set to forecast customer turnover and engagement with search ads in the presence of dynamic and static features. They used the Bing Ads dataset to ensure our established advertisement processes. The findings have been positive and have been able to tackle the challenge of excellent sustainability. Naghibi et al. [90] has created a rotational forest with decision trees as an ensemble approach based on evidence-based belief mechanism and tree-based models (EBFTM) to create possible groundwater maps. None of the methods used in the Ensemble Approach is considered to be considerably greater than each other in terms of the operating characteristics and the region under the curve. The highest result was reached using the EBFTM ensemble and random forest classifiers.

Ali and Prasad [91] built a novel ensemble mode decomposition method with adaptive noise combined with intense learning machines to accurately predict the significant wave height. Several assessments were carried out under the suggested framework, ICEEMDAN-OSELM and ICEEMDANRF, with an adaptive ensemble empiric mode decomposition (Ensemble EM) method and random tree. The suggested ICEEMDAN-ELM approach produced the greatest performance, with the greatest accuracy and the most improved sustainability, according to other approaches. Yamanaka et al. [92] designed a new range of method based on data assimilation-Kalman filter to estimate microstruc-

ture estimation using three-dimensional multi-phase field as device state variables. The proposed approach would improve the accuracy of the model and strengthen its ability to predict the parameters of the system by using all training data. Yadav and Pal [93] established a novel Bagging-Boosting method to estimating women's thyroid gland in the presence of root mean square error mean absolute error relative to decision tree overfitting and neural network. (The DTFNN). The booster bag kit model was around 65% more reliable than the DTFNN model, based on the performance. In brief, the productivity of higher precision and longer lifespans, accompanied by greater commitment to detail and attention to detail, can contribute to the adoption of collaborative protocols by various decision makers.

### 2.2.3 Hybrid machine learning algorithms for BDA

Generally, a hybrid approach integrates the projections from one variable to maximize the second component's coefficient. Hybrid techniques give the benefit of utilizing two or three methods of processing, thus improving performance. These approaches are gaining greater prominence owing to the possibilities they bring. Hu et al. [94] developed the ARIMA-WNN approach for evaluating traffic flow using an application of advanced hybrid machine learning technologies and wavelet neural network with an autoregressive integrated moving average using a fuzzy method. The evolved computing approach has been contrasted with the single type of each contributing mechanism in absolute percentage error, and root means square error. Results showed an improvement of about 60-70% in the hybrid approach's accuracy over single methods. A novel hybrid system for wind power calculation and optimization was proposed by Du et al. [95]. The process required an optimized empirical ensemble of adaptive noise technology to remove the noise, then a wavelet neural network to take the highest precision measurement. The results were compared using a standard deviation percentage error. The use of a hybrid method boosts the accuracy of the prediction and optimization process based on the algorithm's output.

A novel hybrid algorithm focused on attribute and classifier choices was proposed by Zhang et al. [96] solved the credit scoring dilemma, using the classifier ensemble and a sophisticated multi-population niche genetic algorithm. The proposed method increased overall precision of the calculation. The studies were conducted, and quantitative results were drawn. The proposed hybrid solution successfully provided complex estimation and optimization approaches over the single methods base on the data. To test landslide susceptibility, Pham and Prakash [97] developed a novel bagging-based Naive Bayes tree. The proposed hybrid method was correlated with single approaches; such as utilizing

SVMs in terms of the area under the curve and statistical indices. The suggested hybrid BAGNBT strategy contributed to more accurate calculations of landslide susceptibility, which could be implemented as the most robust alternative paradigm for measuring landslide susceptibility across single approaches.

A novel algorithm to boost power load forecasting has been developed by Wu et al. [98]. Sophisticated integration of ensemble empirical mode decomposition was employed in the proposed method. The hybrid approach was contrasted with the other two models in terms of root mean square error, mean absolute error and mean absolute percentage error. The alternative suggested is more effective and more reliable than the criteria based on the findings. A hybrid HybPAS, established by Albalawi et al. [99], includes integrating linear regression-deep neural network models to estimate ply(a) signals in DNA in the presence of sequence-based characteristics and signal processing-based statistics as inputs taken. The hybrid solution was effective at improving precision and performance by 30.29%. Due to their high potential and ability to enhance estimate and optimize performance, hybrid approaches are expanding, and are becoming popular.

The ensemble and hybrid models are the latest and updated versions of machine learning algorithms. They consistently do better than any of the traditional machine learning models. Bagging and boosting strategies are the most common for constructing ensembles. On the other hand, hybrid models are created through combination of machine learning algorithms with heuristic, meta-heuristic, and soft computing techniques.

## 2.3 Research Gaps and Problem Formulation

This section presents the research gaps, problem statement and the research objectives.

### 2.3.1 Research Gaps

The various research gaps identified after exhaustive literature review are as follows:

1. Accessibility of Big Data is on top priority for knowledge discovery process therefore there is a need to break the restraint of CPU-heavy and I/O-poor fully or partially for easy and quick data analysis. The under-developing storage technologies, such as Solid state drives (SSD) and Phase change memories (PCM) can temporarily alleviate the difficulties but they are not the permanent solutions [1].
2. The other important issue is regarding data staging which is related to the heterogeneous nature of data. Data gathered from different sources do not have a structured

format. Transforming and cleaning such unstructured data before loading it into the warehouse for analysis is a challenging task [55]. Efforts have been exerted to simplify the transformation process by adopting technologies such as Hadoop and MapReduce to support the distributed processing of unstructured data formats. However, MapReduce lacks some of the features that have proven paramount to data analysis in DBMS which are as follow [47]:

- MapReduce does not support any high-level language (like SQL in DBMS) and query optimization technique.
- MapReduce is schema-free and index-free and therefore MapReduce job requires parsing each item at reading input and transform it into data objects for data processing, causing performance degradation.
- MapReduce provides the ease of use with a simple abstraction, but in a fixed dataflow. Therefore, many complex algorithms are hard to implement with MapReduce.
- With fault-tolerance and scalability as its primary goals, MapReduce operations are not always optimized for I/O efficiency.

3. To accelerate the analysis of large-scale datasets various algorithms have been developed to cope up with the increasing volume of data but it is still necessary to develop sampling, on-line and multi-resolution analysis methods to produce timely results [1].
4. For Big Data applications, it is difficult to conduct data visualization because of large and high dimension datasets and state of the art data visualization tools mostly lack performance in functionalities, scalability and response time. So it is necessary to rethink the way to visualize Big Data [65].
5. Big Data mining is more challenging compared with traditional machine learning algorithms. For an instance if clustering techniques are taken into consideration, natural way of clustering Big Data is to extend existing methods (such as hierarchical clustering, K-Mean, and density based clustering) so that they can cope with the huge workloads. But most of these extensions usually rely on analyzing a certain amount of samples of Big Data, and vary in how the sample- based results are used to derive a partition for the overall data, and so on [51].
6. Machine learning algorithms which are used in classification and regression problems, suffers from serious scalability problem in both memory use and computation time. Similarly, there are many scale machine learning algorithms but several important specific sub-fields in large-scale machine learning, such as large-scale recommender systems, natural language processing, association rule learning and ensemble learning face scalability problems [49].

7. The learning process in Artificial Neural Networks (ANNs) over Big Data is severely time and memory consuming. Neural processing of large-scale data sets often leads to very large networks which leads to the degraded performance of conventional training algorithms, high training time and memory limitations [48].

### 2.3.2 Problem Statement

Big Data has become one of the emergent topics when learning from data is involved. The exponential growth of data has directed the attention towards the obtaining of effective models that are able to analyze and extract knowledge from these huge data sources. Based on research gap number five it is seen that the vast amount of data, the variety of the sources and the need for an immediate intelligent response pose a critical challenge to traditional machine learning algorithms. According to research gap number six when Big Data is concerned then there is a need to scale up traditional machine learning algorithms. Hybrid and ensemble machine learning methods can give better results than traditional methods by using them on top of popular Big Data paradigm like MapReduce. They have to be used when classical methods fail to meet appropriate scalability and efficiency because these ensemble and hybrid methods have more complex implementation and use require more computational resources.

### 2.3.3 Research Objectives

The following research objectives are formulated:

1. To study and analyze existing techniques for Big Data Analytics in data mining and optimization on traditional classification and clustering algorithms.
2. To propose an efficient machine learning technique for Big Data Analytics on structured and semi-structured data.
3. To develop the proposed technique and apply it on a suitable application.
4. To evaluate the efficiency of the proposed technique for structured and semi-structured data of high volume.

# Chapter 3

# Hybrid machine learning models for predicting types of Human T-cell Lymphotropic Virus

Life threatening diseases like adult T-cell leukemia, neurodegenerative diseases, demyelinating diseases such as HTLV-1 based myelopathy/tropical spastic paraparesis (HAM/TSP), hypocalcaemia, and bone lesions are caused by group of human retrovirus known as Human T-cell Lymphotropic virus (HTLV). Out of the four different types of HTLVs, HTLV-1 is most prominent in scourging over 20 million people around the world and still not much effort has been made in understanding the epidemiology and controlling the prevalence of this virus. This condition further worsens when most of the infected cases remain asymptomatic throughout their lifetime due to the limited diagnostic methods; that are most of the times unavailable for timely detection of infected individuals. Moreover at present there is no licensed vaccination for HTLV-1 infection. Therefore there is a need to develop the faster and efficient diagnostic method for the detection of HTLV-1.

Influenced from the outcomes of the machine learning techniques in the field of bioinformatics, this is the first study in which 64 hybrid machine learning techniques have been proposed for the prediction of different type of HTLVs (HTLV-1, HTLV-2 and HTLV-3). The hybrid techniques are build by permutation and combination of four classification methods, four feature weighting and four feature selection techniques. The proposed hybrid models when evaluated on the basis of various model evaluation parameters are found to be capable of efficiently predicting the type of HTLVs. The best hybrid model has been identified of having accuracy, AUROC value and F1 score of 99.85%, 0.99 and 0.99 respectively. This kind of the system can assist the current diagnostic system for the detection of HTLV-1 as after the molecular diagnostics of HTLV by various screening tests like enzyme-linked immunoassay or particle agglutination assays there is always a need of confirmatory tests like western blotting, immuno-fluorescence assay or radio-immuno-precipitation assay for distinguishing HTLV-1 from HTLV-2. These confirmatory tests are indeed very complex analytical techniques involving various steps.

The proposed hybrid techniques can be used to support and verify the results of confirmatory test from the protein mixture. Furthermore, better insights about the virus

can be obtained by exploring the physicochemical properties of the protein sequences of HTLVs.

## 3.1 Introduction

Human T-cell lymphotropic virus (HTLV-1) falls in the category of retroviruses and was first identified in 1980 in the T-cell line derived from a patient diagnosed with cutaneous T-cell lymphoma [100]. Soon after the discovery, HTLV-1 was found to be identical with adult T-cell leukemia virus (ATLV) at sequence level [101]. A second category of the human retrovirus identified after the discovery of HTLV-1 was designated as HTLV-2. It was found to have resemblance in genome structure and about 70% similar nucleotide sequence homology to HTLV-1 [102]. Later in the year 2005 the other related viruses with HTLV-1, were reported in central Africa as HTLV-3 and HTLV-4 [103]. However, currently only HTLV-1 is found to be linked with human diseases. Globally approximately 20 million people are estimated to be infected with this oncogenic retrovirus HTLV-1, but still its epidemiology is not properly understood [104].

The three main routes for the transmission of HTLV-1 are mother to child transmission(primarily via breast feeding), parenteral transmission (through contaminated blood transfusion or by sharing infected needles) and sexual transmission (male to female)[105]. However, all the transmission modes of HTLV-1 carries a vast similarity with transmission modes of human immunodeficiency virus (HIV-1), yet a very little effort has been made to prevent or reduce its transmission. Furthermore, the infected person has a high risk for adult T-cell leukemia (ATL), development of rapidly progressive malignancy, myelopathy/tropical spastic paraparesis (HAM/TSP), debilitating and sometimes fatal neurologic condition [106, 107].

In the family of retrovirus HTLV-1 falls in the sub-category of deltaretroviruses, which also includes other viruses like bovine leukemia virus, simian T-cell leukemia virus (STLV) and HTLV-2. Like HTLV-1, bovine leukemia virus and STLV also causes lymphoid malignancies in the host. The studies suggest that HTLV and STLV might have originated from the common ancestors as they carries a similar molecular, virological and epidemiological features and for this reason they are designated as primate T-cell leukemia viruses (PTLVs) [108][109].

As per the geographic distribution; southwestern Japan, sub-Saharan Africa, few regions of Central and South America and Caribbean islands are the areas of highest prevalence of HTLV-1 [105]. Since there are various screening tests for HTLV-1 the estimation of it's prevalence is purely based on the serological screening method used in a particular

region that might underestimate it's prevalence in the population. The commonly used screening methods for HTLV-1 includes enzyme-linked immunoassay (EIA) or particle agglutination (PA) assays. EIA's are helpful in joint testing of HTLV-1 and HTLV-2. After that the confirmatory test is required for differentiating HTLV-1 and HTLV-2. On the other hand PA assays test is only for the screening of HTLV-1. The most commonly used confirmatory tets are immuno-fluorescence assay (IFA), western blotting (WB), or radio-immuno-precipitation assay (RIPA) [110]. Further the studies have also shown the uncertainty in the results of the confirmatory tests which may be due to various reasons like the window period, an unspecific antigen (viral)- serum (patient's) reaction or presence of a viral variant. So the screening and treatment of HTLV-1 induced diseases is still unsatisfactory [108, 111]. To overcome this situation, in the year 2014 the Global Virus Network (GVN) launched a taskforce for promoting basic research on HTLV-1, to develop novel methods for HTLV-1 prevention and treatment and to recommend new public health measures [104].

Moreover, the recent technologies like artificial intelligence and machine learning have shown quite a good efficiency in several areas of molecular biology and bio-informatics like drug toxicity prediction, antibody classification, fusion peptide prediction etc. Wu et al. [112] have developed a sequence-based fusion peptide (FP) model. In this hidden markov method is combined with similarity comparison to predict new putative FPs. The developed method have attained the classification accuracies of 91.97% and 92.31% corresponding to 10-fold and leave-one-out cross-validation, respectively.

Moreover, the model has discovered 53,946 np-FPs after scanning sequences without FP annotations. Mei et al. [113] proposed the negative data sampling method based on one-class SVM to predict proteome-wide protein interactions between HTLV retrovirus and Homo sapiens. The computational results of the proposed shows its suitability for negative data sampling over two-class protein-protein interaction (PPI) predictor. Furthermore, valuable insight regarding pathogenesis of HTLV retrovirus has also been obtained after conducting the gene ontology based clustering of the predicted PPI networks. Huang et al. [114] have combined least square regression (LSR) technique with several feature vectorization techniques namely auto covariance (AC), conjoint triad (CT), local descriptor (LD), moran autocorrelation (MA) and normalized moreaubroto autocorrelation (NMB).

The combination of least square regression with these feature vectorization techniques resulted in five different techniques which are collectively abbreviated as LSR+ (i.e. LSRAC, LSRCT, LSRLD, LSRMA and LSRNMB) in the work. Further, in this work authors have combined LSR+ techniques support vector machine (SVM) to predict protein-

protein interactions (PPI) and PPI networks. The proposed technique when applied on four datasets namely Saaccharomyces cerevisiae, Escherichia coli, Homo sapiens and Caenorhabditis elegans showed the better efficiency with respect to existing algorithms. Wang et al. [115] have developed a web service for KAT-specific acetylation site prediction. This work is the extension of their previous work in which the authors have provided the online tool and R package for the method of their previous study. Apart from this other useful services such as the integration of protein–protein interaction information to enhance prediction accuracy are also included in the web server. Zheng et al. [116] have used the penalized matrix decomposition (PMD) for extracting metasamples for clustering for gene expression data. The proposed method outperforms the conventional methods such as hierarchical clustering (HC), self-organizing maps (SOM), affinity propagation (AP) and nonnegative matrix factorization (NMF) to identify the samples with complex classes.

Mota et al. [117] assessed the molecular diversity of gp21 and HBZ proteins in TSP/HAM and healthy carriers. The molecular analysis of DNA samples of the individuals infected with HTLV-1 were performed using bioinformatics tools after its polymerase chain reaction (PCR) and sequencing. Yu et al. [118] have proposed normalized feature vectors - adjacent amino acids (NFV-AAA) approach based on singular value decomposition (SVD) method of the matrix for analyzing the similarity between different protein sequences. The authors claim that the proposed technique efficiently analyzes the similarity between the protein sequences of nine ND5 proteins with correlation of upto 99.70%.

Khanna et al. [119] developed a multilevel ensemble model for the prediction of IgA and IgG from fixed and variable length epitopes. Hooda et al. [120] developed an ensemble framework named as Better Balanced Feature Selection Ensemble ($B^2FSE$) for the classification of drug toxicity molecules. The developed framework is capable of handling imbalanced and high dimensional complex data. Motivated from the initiative of GVN and effectiveness of the machine learning techniques in the filed of molecular biology and bioinformatics an effort has been made in this direction to detect HTLV-1 using machine learning techniques. Here the multiple hybrid machine learning approaches have been proposed. The proposed techniques are capable of efficiently predicting and classifying HTLV efficiently. They can be helpful in the development of faster and efficient screening methods, identification of better antigens for detecting the type of HTLV antibodies. Furthermore in the development of novel therapeutic vaccines and drugs for the treatment of the infected individuals.

Further this paper also include the hasty overview of the dataset and its features, feature extraction, feature weighting and feature selection techniques, that are presented in

Table 3.1: Illustration of physicochemical features used for peptides

| Feature Category | Information | Category Count |
|---|---|---|
| $F_1$ | Aliphatic Index [123] | 1 |
| $F_2$ | Boman Index [124] | 1 |
| $F_3$ | Insta Index [125] | 1 |
| $F_4$ | Homent Index [126] | 2 |
| $F_5$ | Molecular Weight [127] | 2 |
| $F_6$ | Peptide Charge [128] | 45 |
| $F_7$ | Hydrophobicity for 44 scales [129] [130] | 44 |
| $F_8$ | Iso Electric Point at 9 pKscale [128] | 9 |
| $F_9$ | Kidera Factors [131] | 10 |
| $F_{10}$ | aaComp [132] | 18 |
| $F_{11}$ | aaDescriptors: Mean, SD, Var [133] [131] | 159 |

Section 5.2. In Section 3.3 methodology used for the proposed technique is explained. Model evaluation parameters are discussed in Section 3.4. Section 3.5 describes the result analysis and comparison of hybrid techniques proposed in this work. Section 3.6 includes the discussion about the efficiency of the proposed (best) technique and its relevant use in the field of medical sciences. Finally the outcome of the work is concluded in Section 5.5.

## 3.2 Methods and Materials

In this section the brief description of the dataset used and its features; feature extraction technique; labelling of the data using clustering technique and finally the feature importance techniques is provided.

### 3.2.1 Data set and its features

The dataset consisting of protein sequences of HTLVs (HTLV-1, HTLV-2 and HTLV-3); their similar proteins and protein sequences of random viruses other than HTLV (which includes Hepatitis B, Hepatitis C, human Rotavirus A and Banna virus) are downloaded in the fasta format from freely accessible protein sequence database UniProt [121] [122]. This resulted in the generation of four different fasta files as per for HTLV-1, HTLV-2, HTLV-3 with their respective similar proteins and protein sequences of viruses other than HTLV. Table 3.1 shows the physiochemical properties used in the study.

However, the protein sequences of HTLV-1, HTLV-2 and HTLV-3 are 60-80% homologous at amino acid level but they differ in their N-terminal and C-terminal parts of the sequence. Also the length of protein sequences varies in different HTLV sequences. As

Table 3.2: Glimpse of data of extracted features

| $\mathbf{F_a}$ | $\mathbf{F_b}$ | $\mathbf{F_c}$ | $\mathbf{F_d}$ | —— | $\mathbf{F_{kd}}$ | $\mathbf{F_{ke}}$ | $\mathbf{F_{kf}}$ | $\mathbf{F_{kg}}$ |
|---|---|---|---|---|---|---|---|---|
| 17.09 | 3.77 | 3.12 | 0.51 | —— | 0.17 | 0.31 | 0.21 | 0.32 |
| 16.25 | 3.61 | 3.57 | 0.71 | —— | 0.21 | 0.31 | 0.25 | 0.35 |
| 18.61 | 3.35 | 2.12 | 0.48 | —— | 0.12 | 0.23 | 0.16 | 0.43 |
| 28.41 | 4.36 | 4.31 | 0.69 | —— | 0.25 | 0.21 | 0.22 | 0.61 |
| 18.57 | 4.39 | 3.18 | 0.62 | —— | 0.19 | 0.37 | 0.25 | 0.29 |

the different protein sequences code for different genes they perform specific functions. Some of them help in attachment of virus to host are known as envelope proteins. Few are responsible for maintenance of structure of virus and are called structural proteins and rest are non structural proteins which helps in replication of virus.

### 3.2.2 Feature extraction

The features (physicochemical properties) are extracted from four different fasta files containing the protein sequences of HTLVs (HTLV-1, HTLV-2, HTLV-3) and other viruses using an open source software R, licensed under GNU GPL. The feature extraction resulted in a high-dimensional data with 292 features for each protein sequence present in fasta file. Finally the data for all the protein sequences is merged in a single csv file for further evaluation. Table 3.2 shows the glimpse dataset generated after feature extraction from the protein sequences.

### 3.2.3 Clustering of dataset

After the generation of the common csv file the next step is to label the data and for that partition based clustering has been used in this work. The challenges involved in this phase of work are discussed below:

#### 3.2.3.1 Finding optimal clustering algorithm

For finding the optimal clustering algorithm for our dataset the multiple clustering algorithms (K-means, Clara, Pam and Fuzzy C-means) have been tested for the dataset, which provided the K-means as the best clustering algorithm on the basis of cluster validity parameters like dunn index [134] and silhouette coefficient [135]. Table 3.3 shows the score for optimal clustering algorithm.

Table 3.3: Optimal clustering algorithm score

| Clustering Methods | Parameter | Score |
|---|---|---|
| **K-means** | Dunn Index | **0.2476** |
| | Silhouette Coefficient | **0.4854** |
| Clara | Dunn Index | 0.1151 |
| | Silhouette Coefficient | 0.1882 |
| Pam | Dunn Index | 0.1151 |
| | Silhouette Coefficient | 0.1882 |
| Fuzzy C-means | Dunn Index | 0.1151 |
| | Silhouette Coefficient | 0.1847 |

#### 3.2.3.2 K-means for labelling the dataset

K-means is a simple and most popular partitioning clustering algorithm that separates the data into pre-defined number of groups (i.e. k clusters) [136] [137]. It segregates the data points or objects in different clusters in such a way that there must be high intra-cluster similarity and low inter-cluster similarity i.e. data points in the same cluster should be as homogenous as they can be with respect to the data points in the other cluster. The most commonly used distance measure to compute similarity is Euclidean distance. The squared Euclidean distance of every data point from each cluster center is computed to allocate the data point to the closest clusters. Equation 3.1 shows the evaluation of Euclidean distance between point $o_i$ and cluster $c_j$ with N data objects in space and K clusters respectively:

$$d(o, c) = \sqrt{\sum_{i=1}^{N} \sum_{j=1}^{K} (o_i - c_j)^2} \tag{3.1}$$

The next challenge in the work was to find the optimal number of clusters (value of K) for the dataset. This problem is resolved using data visualization technique called elbow plot. This technique involves the method of plotting with in sum of sqaured errors (WSSE) of clusters (Equation 3.2) formed by iteratively executing K-means algorithm for a particular cluster range. The cluster range used for finding optimal number of clusters for this work is 2 to 10. Figure 3.1 depicts optimal number of clusters used for labelling the dataset in this work.

$$WSSE = \sum_{k=1}^{K} \sum_{o_i \epsilon k} (o_i - \mu_k)^2 \tag{3.2}$$

where $o_i$ is a data point that belongs to cluster $C_k$ and $\mu_k$ is the average (mean) of all

the points assigned to cluster $C_k$.

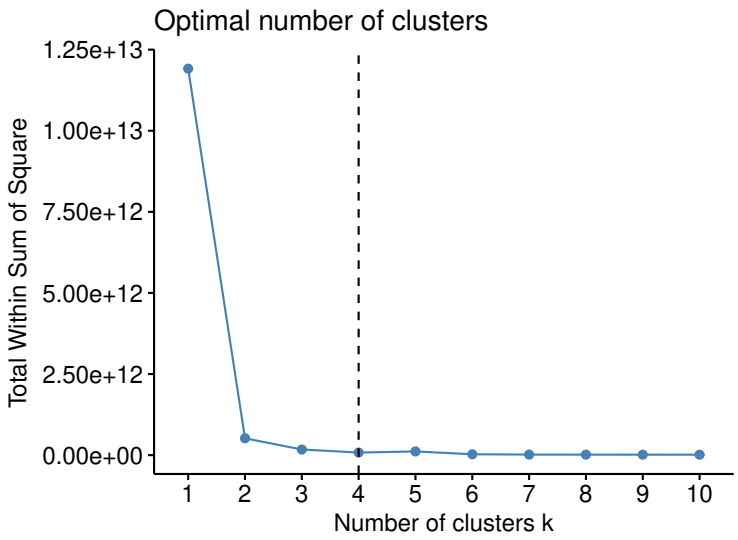



Figure 3.1: Elbow plot showing optimal number of clusters

### 3.2.4 Feature importance

In this section the overview of the feature weighting and optimal feature subset finding techniques used in this work is given. Since dataset is high dimensional consisting of 292 features and to increase the efficiency of the machine learning models used here, it is necessary to assign weights to the features and to find optimal feature subset for the work.

#### 3.2.4.1 Feature weighting

For assigning weights to the features four different types of methods available in FSelector package of R has been used. Table 3.4 illustrates the feature weighting techniques used in this work.

#### 3.2.4.2 Optimal feature subset

For optimal features finding two heuristic and two greedy search techniques namely hill climbing search, best first search, forward search and backward search are used. The results from these four techniques are separately provided to four types of classification techniques namely decision tree (DT), random-forest(RF), support vector machine(SVM) and neural networks(NN) . Table 3.5 provides the overview of the optimal feature searching techniques used.

### 3.2.5 Machine learning techniques

Four widely used classification techniques (DT, RF, SVM and NN) are used for development of the final hybrid techniques proposed in this work. The purpose of considering these different type of classification techniques is to use four different approaches at final (classification) step of the proposed techniques for comparative study. As all the four techniques used in final step uses their own inherent function to classify the data. In general, decision tree uses the greedy approach for classification task. Random forest is the ensemble of the decision tree technique in which multiple decision trees are generated and the final classification is done on the basis of the majority voting. On the other hand, SVM classifies data by constructing the hyperplanes between the data points. While in case of artificial neural networks the classification output depends upon the activation function used for building the neural network. The brief description of all the classification techniques used in this work is as follow:

- $DecisionTree(DT)$: The decision tree algorithm follows a greedy approach to partition the feature space using a recursive binary partitioner [22]. In decision tree algorithm same label is predicted for the bottommost (leaf) partition and for that each partition is greedily chosen from a set of possible splits by selecting the best split, so as to maximize the information gain at a tree node.

- $RandomForest(RF)$: Random Forest algorithm was proposed by Breiman in 2001 as an ensemble of decision tree classification approach with a layer of randomness in the bagging method [138]. A node in random forest is split by selecting the best predictor from the set of predictors that are chosen randomly at that node.

- $SupportVectorMachine(SVM)$: In multiclass problem SVM creates a set of hyperplanes in a high dimensional space, for the purpose of classification or regression task [139] [140]. The hyperpalne with greater functional margin (i.e. the largest distance to the nearest training-data points of a particular class) is selected as the final hyperpalne for classifier, as greater the margin lower will be the generalization error of the classifier.

- $NeuralNetwork(NN)$: Neural Network model used in this work is feed-forward, back-propagation single layer network [141] [142]. In neural networks there are interconnected information processing units that make use of some activation function for transforming input to output. In back-propagation networks training process usually involves some delta rule that calculates the difference between actual outputs and the desired outputs (i.e. error for the output produced). The error is then back-propagated to all the units and the weights are optimized at each connection

Table 3.4: Illustration of feature weighting techniques

| Technique | R package | Description |
|---|---|---|
| chi-squared (chs) | Fselector | Discrete attributes are assigned weights based on a chi-squared test |
| gain-ratio (gr) | Fselector | Discrete attributes are assigned weights based on their correlation with continuous class attribute. formula : (H(Class) + H(Attribute) - H(Class, Attribute)) / H(Attribute) |
| information-gain (ig) | Fselector | Discrete attributes are assigned weights based on their correlation with continuous class attribute. formula : H(Class) + H(Attribute) - H(Class, Attribute) |
| random-forest-importance (rfi) | Fselector | Weights are assigned to the attributes using RandomForest algorithm |

for better results.

Table 3.6 illustrates the machine learning techniques and their tuning parameters used in this work.

## 3.3 Methodology used for the proposed techniques

This section carries the brief description of the methodology used for building the proposed hybrid machine learning techniques by permutation and combination of four feature weighting, four optimal feature selection techniques and four classification methods. The proposed hybrid techniques include three phases for model building and prediction which are discussed as follow:

- In the phase 1 of the technique the protein sequences of three different type of HTLVs and other non HTLV viruses are downloaded in the fasta format from [121] [122] as already quoted in Section 5.2.1. After that the features were extracted from the protein sequences available in fasta files using feature extraction techniques available in bioconductor package of R. This resulted in high dimensional data with 292 features for each protein sequence. Finally the data generated during the feature extraction process for the different fasta files is saved in common csv file. After the generation of the common csv file the next step involved in this phase is labelling of the dataset. For labelling the dataset clustering technique has been used and the challenges involved for this process is to find the optimal clustering algorithm and then optimal number of clusters for our dataset. Finding of optimal

Table 3.5: Optimal feature finding techniques

| **Technique** | **Description** |
|---|---|
| Hill climbing search (HCS) [143] | The HCS algorithm begins with the selection of random attribute set and choosing the best one out of it after evaluating all its neighbours. |
| Forward search (FS) [143] | The FS algorithm uses the greedy search approach for finding the optimal results for the problem. The algorithm begins with expanding the starting node and evaluating its children then finally choosing the best one which in turn becomes a new starting node. This process is unidirectional and moves from an empty set of attributes. |
| Backward search (BS) [143] | Similar to FS, BS algorithm also uses the greedy search approach for finding the optimal results for the problem. The algorithm begins with expanding the starting node and evaluating its children then finally choosing the best one which in turn becomes a new starting node. But in BS, the process moves from an empty set of attributes and is unidirectional. |
| Best first search (BFS) [143] | The BFS algorithm is again somewhat similar to forward search with a difference that it chooses the best node from all already evaluated nodes and again evaluates it and this This best node selection process is repeated approximately maximum backtracks times (default value is 5) in case no better node found. |

cluster algorithm has been discussed in Section 3.2.3.1. The selection of optimal number of clusters for the dataset has been discussed in Section 3.2.3.2. This phase is common for all the hybrid techniques proposed in this work.

- The phase 2 of the work involves some series of steps which includes the assigning weights to the features and finding optimal feature subset. After the labelling of dataset via clustering technique in phase 1 the next step is to assign weights to the features for the purpose of feature importance. For assigning weights to the features 4 different feature weighting techniques are used that are discussed in Section 3.2.4.1. Once the feature weighting is done the next step is to find the optimal feature subset and for this 4 different optimal features finding techniques are used that are discussed in Section 3.2.4.2. The output of each feature weighting technique is separately provided to each feature selection technique and the resulting output of each combination is utilized in the pahse 3 i.e used for training 4 different classifaton models.

Table 3.6: Machine learning techniques used

| **Model** | **Method** | **R Package** | **Tuning parameters** |
|---|---|---|---|
| Decision Tree [22] | rpart | rpart | (usesurrogate=0, maxsurrogate=0) |
| Random Forest [144] | randomForest | randomForest | ntree=1500,mtry=10 |
| Support Vector Machine [139] | ksvm | kernlab | kernel="rbfdot", prob.model=TRUE |
| Neural Network [142] | nnet | nnet | size=10 |

- The phase 3 of the technique starts with splitting the data with optimal features (generated by the combination of each feature weighting and optimal feature selection technique) in training set and testing set for the purpose of training and evaluation of the machine learning models. The ratio of split is 70:30 i.e. 70% of dataset with optimal features is used for training of the models and remaining 30% for making predictions. Once the models are trained using training datasets then they are validated using evaluation parameters discussed further in Section 3.4. Finally to verify the consistency of trained models the K-fold cross validation is preformed.

Figure 3.2 shows the methodology of the proposed techniques used in this work.

## 3.4 Model Evaluation

Model evaluation is the process of measuring the performance of the trained model using various parameters. The various model evaluation parameters used in this work includes accuracy, recall, specificity, precision, AUROC,F1 score and negative predicted value. Furthermore the performance consistency and robustness of the models has also been analysed using K-fold cross validation. This section carries the brief description of the evaluation parameters used; where,
True Positive(TP) : when actual label or class is positive and is predicted as positive
True Negative(TN) : when actual label or class is negative and is predicted as negative
False Positive(FP) : when actual label or class is negative but predicted as positive
False Negative(FN) : when actual label or class is positive but predicted as negative
L is the set of M classes or labels,

$$L = l_1, l_2, l_3, ....., l_M \tag{3.3}$$

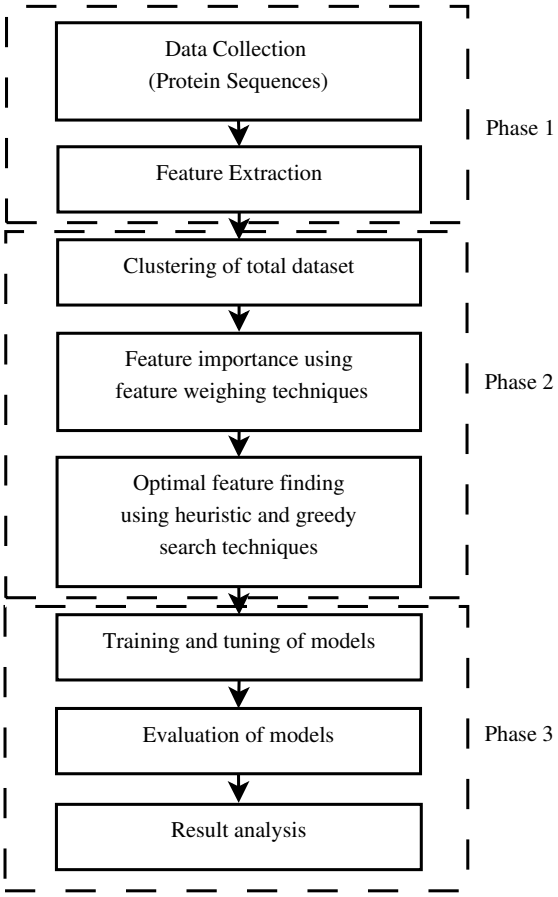



Figure 3.2: Methodology used for proposed techniques

The actual label vector y consists of N elements

$$y_1, y_2, y_3, ....., y_N \epsilon L \tag{3.4}$$

and a prediction vector $\hat{y}$ of N elements is generated by a multiclass prediction algorithm

$$\hat{y_1}, \hat{y_2}, \hat{y_3}....., \hat{y_N} \epsilon L \tag{3.5}$$

and,

$$\hat{\delta}(x) = \begin{cases} 1 & if\ x = 0 \\ 0 & otherwise \end{cases} \tag{3.6}$$

### 3.4.1 Accuracy

Accuracy is the measure of correctness of the model [119]. It tells how precisely the model is able to predict the target value or class for the new data. Higher the accuracy more precise are the predictions made by the model. It can be computed as follow:

$$Accuracy = \frac{TP}{TP + FN} * 100 \tag{3.7}$$

### 3.4.2 Recall or True Positive Rate(TPR)

Recall also known as sensitivity [145] is true positive rate of the model. It tells how many correct predictions for a particular class are made by the classifier with proportion to actual count for that class. It can be computed as follow:

$$\begin{gathered} Recall(TPR) = \frac{TP}{P} = \frac{TP}{TP + FN} \\ or \\ TPR = \frac{\sum_{i=1}^{N} \hat{\delta}(\hat{y} - l) * \hat{\delta}(y_i - l)}{\sum_{i=1}^{N} \hat{\delta}(y_i - l)} \end{gathered} \tag{3.8}$$

### 3.4.3 Specificity or True Negative Rate(TNR)

Specificity or TNR [145] is the measure of ability of the predictive method to correctly identify the labels that actually do not belong to the particular class and it can be computed using equation 3.9.

$$Specificity(TNR) = \frac{TN}{TN + FP} \tag{3.9}$$

### 3.4.4 Precision or Positive Predicted Value(PPV)

Precision or PPV [145] is defined as the ratio of total number of true positives with the sum of number of true positives and number of false positives predicted by classifier. Equation 3.10 is used for computing PPV for the classifiers.

$$\begin{aligned} Precision(PPV) &= \frac{\sum TruePositive}{\sum PredictedConditionPositive} \\ \Rightarrow PPV &= \frac{TP}{TP + FP} \end{aligned} \tag{3.10}$$

### 3.4.5 Negative Predicted Value(NPV)

NPV [145] is defined as the ratio of total number of true negatives with the sum of number of true negatives and number of false negatives predicted by classifier. Equation 3.10 is used for computing NPV for the classifiers.

$$\begin{aligned} NPV &= \frac{\sum TrueNegative}{\sum PredictedConditionNegative} \\ \Rightarrow NPV &= \frac{TN}{TN + FN} \end{aligned} \tag{3.11}$$

### 3.4.6 F1 score

F1 score [145] is the harmonic average of precision (PPV) and true positive rate of the predictive method. It's values lies between 0 and 1, value 0 signifies the worst score and the perfection of the classifier increases as this value approaches to 1. F1 score can be computed using equation 3.12.

$$F1score = 2 * \frac{PPV * TPR}{PPV + TPR} \tag{3.12}$$

### 3.4.7 Area under ROC Curve (AUROC)

Area under ROC curve (AUROC) is another important parameter for measuring the efficiency of the classifier. ROC curve is generated by plotting the true positive rate of the classifier against its false positive rate. AUROC is just the area under the ROC curve. Its value ranges between 0 and 1. Higher the value of AUROC (i.e. closer to 1) better is the classifier.

### 3.4.8 K-fold cross validation

K-fold cross validation [146] is the one of the most popular technique used for analysing the consistency and robustness of the predictive method. In K-fold cross validation, the training dataset is split into k sub-samples of equal size. Then out of the k sub samples the k-1 sub samples are used as the training data for training the model and the remaining one sub sample is used as a testing data in each of the k iteration in such a manner that each of the k sub-sample is used exactly once as the validation or testing data. Finally the results generated in k iterations can be averaged for taking the estimate of the mean accuracy of the predictive method or the k results produced can be plotted on graph (scatter plot or box plot) for visualizing the fluctuation or variation in accuracy values of

the predictive method or classifier. The method with less variation is considered as more consistent method.

## 3.5 Result Analysis

The results obtained by implementing the different hybrid models on the high dimensional dataset comprising of 292 features extracted from proteins of HTLV-1, HTLV-2, HTLV-3 and non HTLV have been discussed in this section. The comparison of different hybrid approaches used in this work is also provided for the purpose of finding the overall best performing hybrid approach and for that the evaluation parameters discussed in Section 3.4 are taken into consideration.

### 3.5.1 Analysis of accuracy

The accuracies of different hybrid approaches used are shown in Table 3.7. It can be clearly seen from Table 3.7 that the decision tree performs well with all the feature weighting and optimal feature selection techniques giving the accuracy of more than 95% in all cases. The highest accuracy of 98.54% is achieved with the hybrid approach of chi-squared(chs) and best-first search(bfs) in case of decision tree models. While on the other hand random-forest turns to be the second best model in terms of accuracy with all the feature weighting and optimal feature finding techniques with accuracy as lowest as 73.4%, but it gives the highest accuracy of 99.85% with random-forest-importance (rfi) and forward search (fs) technique from all other hybrid models used in this work. Lastly, the hybrid SVM and hybrid neural network models provide the accuracy range from 64.06% to 97.58% and 54.96% to 98.68% respectively. Figure 3.3 shows the variation in accuracies achieved by all hybrid approaches of four machine learning techniques (DT, RF, SVM and NN) used in this work. From this box plot it can be concluded that the hybrid decision tree models have least variation as compared to other hybrid models.

### 3.5.2 Analysis of other parameters

For finding the best hybrid model accuracy is not the only parameter when it is a multi-class problem; the other parameters like true positive rate(recall), true negative rate(specificity), positive predicted value (precision), negative predicted value of a particular class, area under ROC curve (AUROC) and F1 score of the predictive method also play a vital role. Table 3.8 shows the comparative results of all the other parameters for the best hybrid models, achieved in this work. From table 3.8 it is clearly seen that random-forest model in combination with random-forest-importance and forward search

Table 3.7: Accuracy Comparison

| **Model** | **Feature Weighting Techniques** | **Optimal Feature Finding Techniques** | | | |
|---|---|---|---|---|---|
| | | **Hill Climbing Search** | **Backward Search** | **Forward Search** | **Best-First Search** |
| Decision Tree | chi-squared | 98.44 | 96.88 | 98.44 | **98.44** |
| | random-forest importance | 95.31 | 95.31 | 98.44 | 96.94 |
| | information gain | 95.31 | 95.31 | 95.31 | 96.88 |
| | gain ratio | 96.88 | 96.88 | 98.44 | 95.31 |
| Random Forest | chi-squared | 95.96 | 85.94 | 95.31 | 99.45 |
| | random-forest importance | 94.12 | 87.5 | **99.85** | 98.44 |
| | information gain | 96.06 | 76.56 | 98.44 | 99.56 |
| | gain ratio | 94.08 | 73.4 | 96.88 | 96.88 |
| SVM | chi-squared | 84.30 | 71.88 | 96.94 | 95.49 |
| | random-forest importance | 80.50 | 71.88 | 97.14 | 95.99 |
| | information gain | 82.04 | 64.06 | 95.21 | 96.02 |
| | gain ratio | 85.80 | 75 | 95.56 | **97.58** |
| Neural Networks | chi-squared | 80.47 | 54.69 | **98.68** | 60.97 |
| | random-forest importance | 79.25 | 65.62 | 97.78 | 80.83 |
| | information gain | 84.08 | 56.25 | 96.65 | 62 |
| | gain ratio | 86.21 | 57.81 | 98.42 | 55.93 |

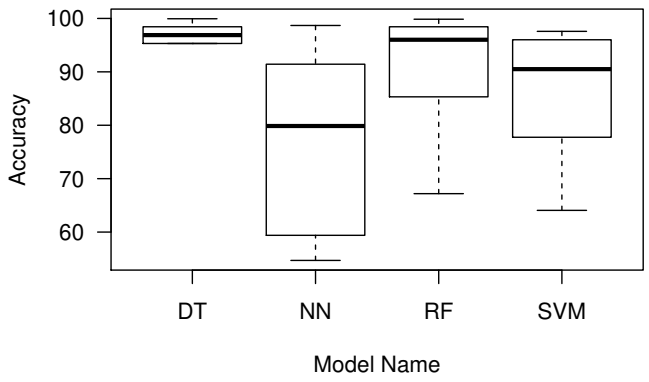



Figure 3.3: Accuracy box-plot of models

Table 3.8: Parameters comparison of best hybrid models

| Model | Avg TPR | Avg TNR | Avg PPV | Avg NPV | F1 Score | AUROC |
|---|---|---|---|---|---|---|
| $chs - bfs - DT^1$ | 0.71 | 0.99 | 0.75 | 0.74 | 0.73 | 0.91 |
| $rfi - fs - RF^2$ | **0.99** | **0.99** | **0.99** | **0.99** | **0.99** | **0.99** |
| $gr - bfs - SVM^3$ | 0.87 | 0.97 | 0.79 | 0.97 | 0.74 | 0.89 |
| $chs - fs - NN^4$ | 0.89 | 0.98 | 0.95 | 0.99 | 0.91 | 0.98 |

[1]Decisison Tree in combination with chi-squared and best-first search, [2]Random Forest in combination with random-forest-importance and forward search, [3]SVM in combination with gain-ratio and best-first search, [4]Neural Network in combination with chi-squared and forward search.

Table 3.9: Illustration of optimal features found by feature selection techniques

| Feature Selection Technique | Optimal Feature Subset | Count |
|---|---|---|
| Hill climbing search | F1-aliphaticIndex, F6-homentIndex1, F6-homentIndex2, F7-molecularWeight2, ... ,F13.3-aaDescriptorsVar53 | 155 |
| Forward Search | F7-molecularWeight2, F9-hydro28 | 2 |
| Backward Search | F1-aliphaticIndex, F6-bomanIndex, F6-instaIndex, F7-molecularWeight1, ... , F13.3-aaDescriptorsVar53 | 189 |
| Best first search | F7-molecularWeight2, F8-pCharge15 | 2 |

technique gives the best results in terms of all the parameters with F1 score and AUROC value of 0.99. In contrast to hybrid decision tree models which are found better in terms of accuracy, this hybrid random-forest approach shows quite better results in terms of all other parameters, as all the decision tree hybrid models shows the poor prediction of HTLV-2.

### 3.5.3 Analysis of K-fold cross validation

Another parameter which need to be analysed is reliability of the technique i.e. whether the model is free from over-fitting and under-fitting issues. Over-fitting of the classifier means that the classifier is giving the good accuracy with the training data but performing poorly with testing data. On the other hand under-fitting signifies that the classifier is performing poorly for both the training and testing data. To analyse the reliability and consistency of the used hybrid techniques, 10 fold cross validation of the best hybrid models is performed. Figure 3.4 shows the variation of accuracy results of 10 fold cross validation for best hybrid models of decision tree, random-forest, SVM and neural networks. It is clearly visible from the box-plot (Figure 3.4) that hybrid random forest model shows the most consistent accuracy results with respect to other hybrid approaches.

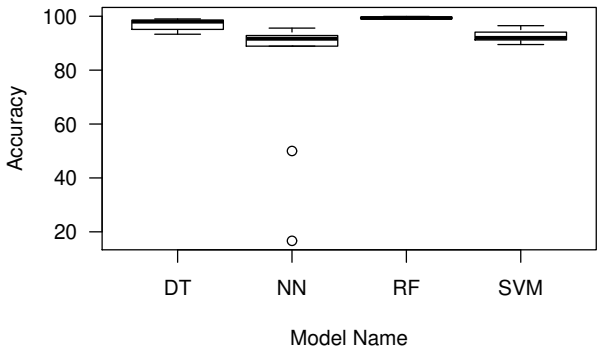



Figure 3.4: Accuracy box-Plot of K-fold cross validation for best hybrid models

## 3.6 Discussions

The four feature selection techniques are used for providing the optimal feature subset for training the classification models. The feature selection techniques find the optimal features according to their inherent objective function. Table 3.9 shows optimal features found by different feature selection techniques. Based on these the optimal features the models gets trained and predicts the type of HTLV from the testing (new) data. As discussed in Section 3.5 the random forest in combination with random forest importance and forward search (rfi-fs- RF) shows the most accurate and consistent results. This model is able to predict all type of HTLVs from the data of HTLVs and non HTLV protein sequences.

Also, by analyzing all the evaluation parameters it is found that rfi-fs- RF proves to be the best predictive method in terms accuracy, TPR, TNR, PPV, NPV, F1 score, AUROC and 10-fold cross validation. One of the reason rfi-fs-RF classifier outperforms all other hybrid approaches is the use of random forest algorithm in the very first step that is feature weighting and in the final step while predicting classes of the new (test) data.

As random forest algorithm is the ensemble technique of decision tree in which several decision trees are generated using random sample of features and data subset and the final class is predict on the behalf of majority voting. This reduces the chances of over-fitting and improves the overall efficiency of the classifier. Further the information gain, gain ratio and chi-square algorithms which are being used in other hybrid techniques have their own drawbacks. Information gain algorithm can get biased by giving more weightage to feature with larger values which can result in over-fitting and selection of

non-optimal features for training the classifier.

Similarly chi square feature weighting technique is at par with information gain technique. However, gain ratio technique is somewhat better than information gain as it reduces bias created in information gain technique by taking the intrinsic information of the split into account. Due to this property it may provide higher weightage to the features just on the basis of low intrinsic information and consider the features greater than average information gain resulting in over compensation. So the performance of the final classifier gets effected by feature weighting technique and optimal selection of the feature subset to train the classifier.

The other reason for rfi-fs-RF to outperform the other hybrid techniques is the optimal feature finding technique. If we compare the optimal feature subset finding techniques of best hybrid technique in each category in turns out be forward search and best first search. Both of these techniques follow the greedy approach for finding the final solution but forward search algorithm does the more exhaustive search starting from the empty node to complete feature set for finding the optimal feature subset. Another reason for rfi-fs-RF to be the best predictive technique is the use of random forest algorithm in the final classification step. This works includes the multi-classification problem to predict the type of HTLV and random forest outperforms the decision tree hybrid approaches as in random forest classification several decision trees are constructed and final class is predicted on the behalf of majority voting which enhances the overall performance of the classifier.

In case of SVM hybrid techniques for multi-classification problem SVM reduce it to multiple binary classification problem. This results in maximizing the margin and have to rely on the concept of distance between different points which needs to be converted to the probability. For artificial neural networks hybrid technique smaller training dataset can be one reason that it lacks in performance as compared to rfi-fs-RF.

In a nutshell it can be said that this kind of the system is capable of assisting the current diagnostic system for the detection of HTLV-1. Since, after the molecular diagnostics of HTLV by various screening tests like enzyme-linked immunoassay or particle agglutination assays there is always a need of confirmatory tests like western blotting, immunofluorescence assay or radio-immuno-precipitation assay for distinguishing HTLV-1 from HTLV-2. These confirmatory tests are indeed very complex analytical techniques involving various steps. The proposed hybrid techniques can be used to support and verify the results of confirmatory test from the protein mixture. It can be widely used in medical studies and for exploring the physicochemical properties of HTLVs for further research and gaining better insights about the virus.

## 3.7 Conclusion

Considering the need to understand the epidemiology of HTLV-1 and to control its prevalence efforts have been made using the machine learning techniques for gaining better insights about the virus. In this work the 64 hybrid machine learning methods are developed and tested for the prediction of different type of human t-cell lymphotropic virus (HTLV) in high dimensional dataset of 292 features extracted from the protein sequences of HTLVs (HTLV-1, HTLV-2 and HTLV-3), non HTLV and their similar proteins. The dataset is firstly labelled using the K-means clustering algorithm, then the feature weighting is done so as to identify the important features for training the machine learning models. Finding of the optimal features to serve as an input to train the models is done using two heuristic search and two greedy search techniques.

Finally the models are trained using the optimal features and are evaluated on the basis of model accuracy, recall (TPR), specificity (TNR), precision (PPV), negative predicted value (NPV), AUROC value and F1 score. Furthermore, the robustness of the best models in each category is explored using 10-fold cross validation. Finally based on the analysis of all the evaluation parameters it is found that random-forest in combination random-forest importance and forward search is the most accurate and reliable predictive method among other methods developed in this work.

# Chapter 4

# Kinematic viscosity prediction of nanolubricants employed in heavy earth moving machinery

Recent researchers widely used nanoparticle additives for improving thermal and rheological properties of machine lubricant. In present study the effect of $Al_2O_3$ and $CeO_2$ nanoparticles on transmission oil (SAE30), hydraulic oil (HYDREX100) and gear oil (EP90) of Heavy Earth Moving Machinery is investigated. Nano-lubricant samples are prepared in 0.01% - 4% nanoparticle volume fraction range. Four machine learning techniques namely decision tree (DT), random forest (RF), generalized linear models (GLM) and neural network (NN) have been used to predict the kinematic viscosity for $Al_2O_3$ and $CeO_2$ nanolubricants. Further, multi-criteria decision-making (MCDM) technique named Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) have been used to find the best predictive method in each category of the nanolubricants. Decision tree (DT), random forest (RF) and neural network (NN) methods are found to be most accurate in kinematic viscosity prediction of transmission oil ($R^2 = 0.861$), hydraulic oil ($R^2 = 0.971$) and gear oil ($R^2 = 0.973$), respectively. Eventually, this study provides a new theoretical basis in nanolubricants for creating software programs that allow the user to know the lubrication oil efficiency to suppress the operating costs for heavy earth moving machinery.

## 4.1 Introduction

Rheological behaviour of lubricants significantly affect the performance of a machine. Heavy Earth Moving Machinery (HEMM) like rope shovels, draglines, bucket-wheel excavator and hydraulic shovels, needed robust lubrication to perform in elevated off road environmental condition. Engine, transmission, hydraulic and gear oil are majorally employed lubricants in HEMM. The properties of HEMM lubrication oil is indicative of machine life and its performance [147]. The recent researches perceive nanoparticles additives as a new approach for improvement in lubricant properties [148, 149, 150, 151].

The suspension obtained with the addition of nanoparticles is termed as nanolubricant [152].

The dispersion of nanoparticles in base oil, significantly modify the rheological behaviour of base oil. Viscosity, which is a prime rheological property, changes with nanoparticles additives [153]. It infers the resisting force in between fluid layers in relative motion. Temperature and nanoparticles concentration significantly changes base fluid viscosity [154]. Researchers attempted numerical, experimental and intelligence methods to accurately determine the viscosity of nanolubricants [152] [155] [156] [157]. The machine learning methods provides solution for complicated engineering challenges within least computational time. This provide the possible solution for the challenge of accurately determination of thermophysical properties of nanolubricants.

Shahsavar et al [158] used ANN to evaluate the thermal conductivity of $Fe_3O_4$/paraffin nanofluid. The accuracy of the model was assessed based on four known statistical indices which includes root mean square (RMS), root mean square error (RMSE), mean absolute deviation (MAE) and coefficient of determination ($R^2$).They reported that the proposed model of thermal conductivity could estimate the outputs with RMS, RMSE, MAE and $R^2$ values of 0.0678, 0.0179, 0.0041 and 0.96 respectively. Esfe et al [157] tested MWCNT (50%) - Al2O3 (50%)/10W40 hybrid nanofluid at temperatures and volume fraction range of 5°C - 55°C and 0.05% - 1% respectively. They observed co-efficient of determination for the proposed correlation and ANN are 0.9973 and 0.9944 respectively. Furether, they used Genetic Algorithm-Radial Basis Function (GA-RBF) neural networks, Least Square Support Vector Machine (LS-SVM) and Gene Expression Programming using artificial intelligence methods to predict the viscosity of $TiO_2$/SAE50 nanolubricant. They reported RMSE values of 0.58, 1.28, 6.59 and $R^2$ values of 0.99998, 0.99991 and 099777 for GA-RBF, LS-VM and GEP respectively [159]. Vakili et al [160] used genetic algorithm in ANN to improve learing process for rheological property prediction graphene/deionized water nanofluid. They observed model predict value with 0.985 $R^2$.

[161] used temperature, shear rate, nanoparticle size, nanoparticle density and particle concentration as input variables to develop ANN model and reported correlation coefficient as 0.9954. [162] analyzed the dynamic viscosity of MWCNT (40%) - $SiO_2$(60%)/5W50 under temperature, volume fraction and shear rate range of 5°C - 55°C , 0% - 1% and 50-800 rpm respectively. Multi-layer perceptron algorithm is used in ANN to evaluate relative viscosity and observed $R^2$ value of the proposed model as 0.9914. [163] used ANN approach for irreversibility performance analysis of domestic refrigerator by utilizing LPG with $TiO_2$ –lubricant as replacement of R134a. They found ANN model predictions concurred well with experimental results and brougth out an absolute fraction of variance of

(0.989 - 0.990), RMSE of (0.831 - 1.061) and mean absolute percentage error of (1.734 - 2.056 %) respectively with experimental results.

In the present study, kinematic viscosity of nanolubricants, formed with $Al_2O_3$ and $CeO_2$ nanoparticles in transmission oil, hydraulic oil and gear oil, is predicted using machine learning technique. Previous studies shows hard $Al_2O_3$ nanospheres contributed in ball bearing effect [153] and $CeO_2$ nanoparticles have extreme pressure characteristics [164]. These anti-friction properties of $Al_2O_3$ and $CeO_2$ nanoparticles, motivates to use with heavy earth moving machinery lubricants. Viscosity and density of samples where measured with varying temperature and particle volume fraction. The four machine leaning methods, Decision tree, random forest, neural network and linear model are used for prediction of viscosity. Models are trained on the 70% experimental data and remaining 30% is used for testing. Further, the multi-criteria decision making technique, TOPSIS, is used to determine the most suited machine learning method for each case. In this paper the brief overview of material and experimentation, machine learning methods, model evaluation parameters and MCDM technique TOPSIS is presented in Section 4.2. Section 4.3 carries the thorough analysis of the results obtained by experimentation and machine learning techniques in the prediction of kinematic viscosity of the nano-lubricants used in this work. Finally, the outcome of this work is concluded in Section 6.1.

## 4.2 Materials and Methods

### 4.2.1 Material and measurement

Commercial grade transmission oil (SAE30), hydraulic oil (HYDREX100) and gear (EP90) has been used as a base oil in the present study. Base oils are widely employed as lubricants in HEMM. $Al_2O_3$ and $CeO_2$ are widely used environment friendly nanoparticle additives [164] [165]. The spherical morphology of $Al_2O_3$ improves anti-friction property due to ball bearing effect [153] and $CeO_2$ nanoparticles improves load carrying capacity by third body effect. In present study, $Al_2O_3$ and $CeO_2$ has been used as nanoparticles additives with nominal grain size of 60 nm. Figure 4.1 and 4.2 show the FESEM (Make-Carl Zeiss AG, Model- Supra 55) images of $Al_2O_3$ and $CeO_2$ nanoparticles respectively. It can be noted that alpha $Al_2O_3$ nanoparticles have spherical morphology. Figure 4.2 clearly depicts plate shaped morphology of $CeO_2$ nanoparticles. Nanolubricant samples have been synthesised using two step method. Dry nanoparticles dispersed in base oil in the particle volume fraction range of 0.01% to 4% and weight of the appropriate volume fraction measured from equation 4.1, where w and $\rho$ indicates weight and density [153]. The wide range of nanoparticle volume fraction is taken to incorporate all possible particle

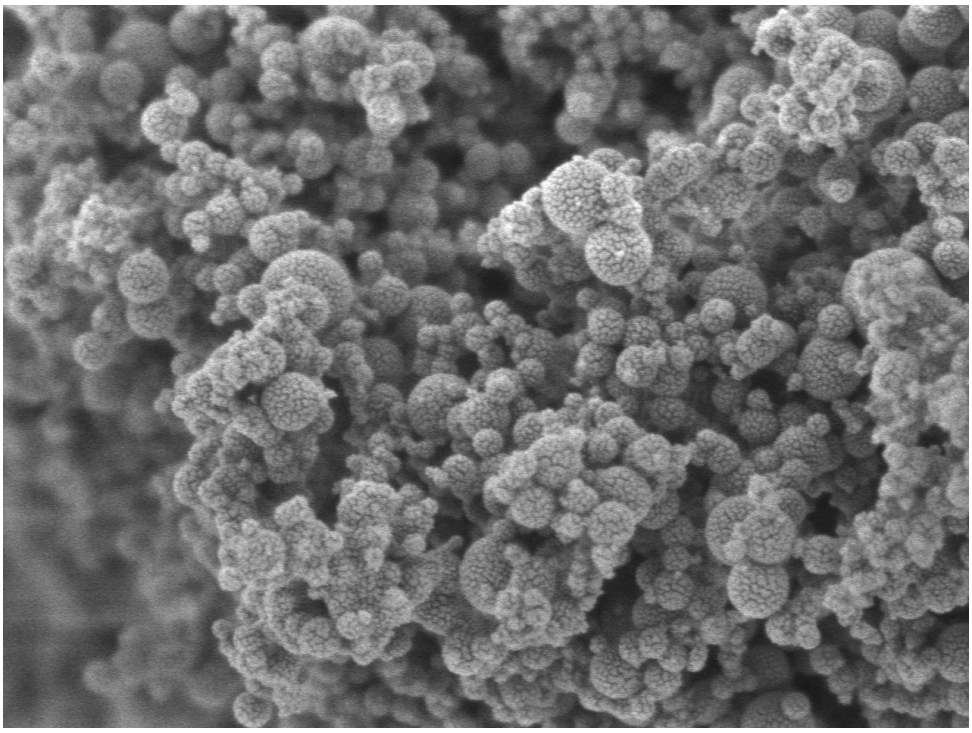


Figure 4.1: FESEM microgrph for $Al_2O_3$ nanoparticles

volume fraction used in existing studies. The suffix $p$ and $l$ indicates particle and lubricants, respectively. The uniform dispersion of nanoparticles is performed by magnetic stirrer at 500 rpm for 45 minutes magnetic stirrer followed by intensive ultra-sonication (20 kHz). The existing studies show that the used duration of magnetic stirrer with ultra-sonication produced homogeneous dispersion of nanoparticles [166] [167]. Aging test indicates that samples are fairly stable for test duration. The viscosity and density of nanolubricant samples has been measured by using Stabinger Viscometer (Make - Anton Paar, Model - SVM3000) in temperature range of 20°C - 50°C. The temperature from 20°C to 50°C was chosen in the current study to simulate the warm-up phase, which was a critical operating condition in automobile engines [168]. In SVM 3000 a lightweight magnetic rotor floats in a liquid filled tube, which rotates at constant speed. The rotor is centred by the centrifugal forces. The relative speed of rotor is calibrated in terms of viscosity. Stabinger $Viscometer^{TM}$ measures the dynamic viscosity and density of oils according to ASTM D7042. From this result, the viscometer automatically calculates the kinematic viscosity and delivers measurement results which are equivalent to ISO 3104 or ASTM D445. All the measurements are performed under steady state conditions. Repetitive measurement indicates that the test results of viscosity and density have uncertainty of $\pm 2\%$ and $\pm 0.5\%$ respectively.

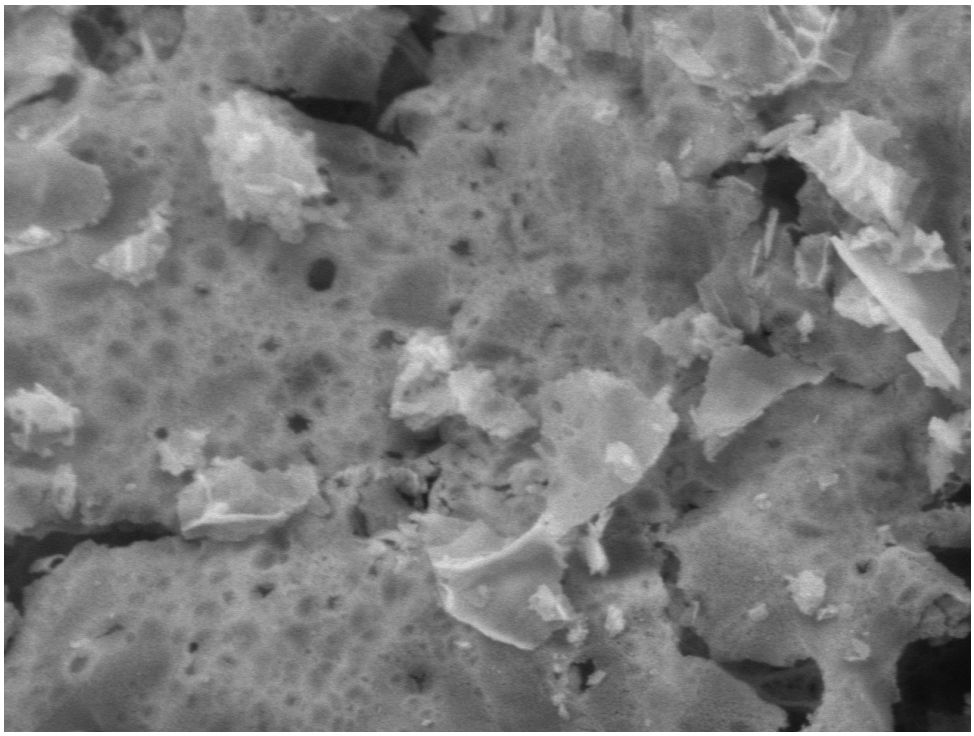


Figure 4.2: FESEM micrograph for $CeO_2$ nanoparticles

$$\phi = \frac{W_p/\rho_p}{W_p/\rho_p + W_l/\rho_l} \tag{4.1}$$

### 4.2.2 Machine Learning Techniques

In this work the machine learning methods available in an open source software R licensed under GNU GPL are used. The brief description of four machine learning techniques used is as follows:

1. Decision Tree (DT): The decision tree algorithm follows a greedy approach to partition the feature space using a recursive binary partitioner [22]. In the decision tree algorithm same label is predicted for the bottommost (leaf) partition and for that each partition is greedily chosen from a set of possible splits by selecting the best split, so as to maximize the information gain at a tree node. For constructing the decision tree for regression problems standard deviation reduction is used instead of information gain for partitioning the data.

2. Random Forest (RF): Random Forest algorithm was proposed by Breiman in 2001 as an ensemble of decision tree approach with a layer of randomness in the bagging method [144]. A node in random forest is split by selecting the best predictor from the set of predictors that are chosen randomly at that node.

3. Neural Networks (NN): Neural Network model used in this work is feed-forward, back-propagation single layer network. In neural networks there are interconnected information processing units that make use of some activation function for transforming input to output [142]. In back-propagation networks training process usually involves some delta rule that calculates the difference between actual outputs and the desired outputs (i.e. error for the output produced). The error is then back-propagated to all the units and the weights are optimized at each connection for better results.

4. Generalised Linear Models (GLM): It uses linear models to carry out regression, single stratum analysis of variance, and analysis of covariance [169].

### 4.2.3 Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) Method

The TOPSIS method [170] is the most commonly used MCDM technique in wide range of applications. Generally in MCDM problems, there exist the cases in which alternative, with minimum Euclidean distance from the positive ideal solution (PIS), has the shorter distance from the negative-ideal solution (NIS), than the other alternative(s). For such situations, the TOPSIS method provides complete ranking with closeness scores by calculating closeness to the ideal solution and distance from the negative ideal solution. In this work we have used the TOPSIS method available in an open source software R licensed under GNU GPL for ranking and finding the best predictive model for each category of nanolubricant [171].

### 4.2.4 Model Evaluation Parameters

1. Correlation (r): Correlation is used to describe the statistical relationship between actual and predicted values. It is defined as in equation 4.2.

$$r = \frac{\sum_{i=1}^{n}(r_i - \bar{r_i})(s_i - \bar{s_i})}{\sqrt{\sum_{i=1}^{n}(r_i - \bar{r_i})^2 \sum_{i=1}^{n}(s_i - \bar{s_i})^2}} \tag{4.2}$$

   where, $r$ is the actual value, $s$ is the predicted value, $\bar{r}$ is the mean of the all actual values, $\bar{s}$ is the mean of the all predicted values, and n is the number of instances. The range of correlation value lie between [-1,1]. The correlation value tending towards 1 or -1 is considered to be good.

2. Coefficient of determination ($R^2$): It ($R^2$) is used to recapitualte the explanatory

power of the regression model. $R^2$ , describes the proportion of variance of the dependent variable explained by the regression model. If the $R^2$ value tends towards 1 it signifies perfection of the regression model and if $R^2$ is 0 then it signifies that the regression model is a total failure i.e. no variance is explained by regression. To compute coefficient of Determination the square of $r$ (i.e. correlation) value is taken. It is defined as in equation 4.3.

$$R^2 = r \times r \tag{4.3}$$

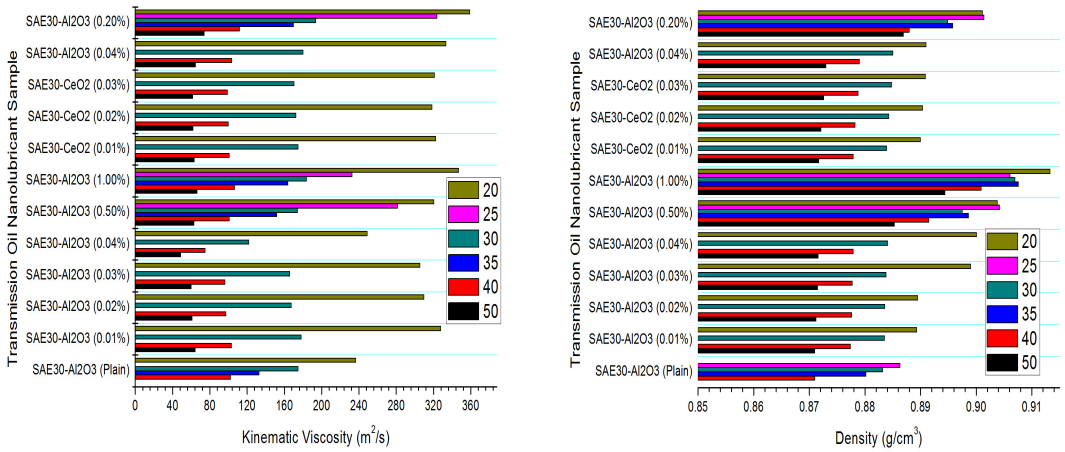
3. Mean Absolute Error (MAE): For a regression model MAE is used to measure the average absolute difference between the predicted and actual values. Equation 4.4 is used to calculate the MAE value, where, $r$ is actual value, $s$ is predicted value, and $n$ is the total number of instances..

$$MAE = \frac{\sum_{i=1}^{n} \mid s_i - r_i \mid}{n} \tag{4.4}$$

4. Root Mean Square Error (RMSE): The average error rate of a regression model is measured by RMSE. However, the models whose errors are measured in the same units can only be compared using this parameter. Equation 4.5 is used to calculate RMSE, where, $r$ is actual target, $s$ is predicted target, and $n$ is the total number of instances..

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(s_i - r_i)^2}{n}} \tag{4.5}$$

5. K-fold cross validation: It is the one of the most popular technique used for analysing the consistency and robustness of the predictive method. In K-fold cross validation [146], the training dataset is split into $k$ sub-samples of equal size. Then out of the $k$ subsamples the *k-1* sub samples are used as the training data for training the model and the remaining one sub sample is used as a testing data in each of the $k$ iterations in such a manner that each of the $k$ subsample is used exactly once as the validation or testing data. Finally the results generated in $k$ iterations can be averaged for taking the estimate of the mean values of model evaluation parameters of the predictive method. The method with less variation is considered as more consistent method.




(a) Viscosity

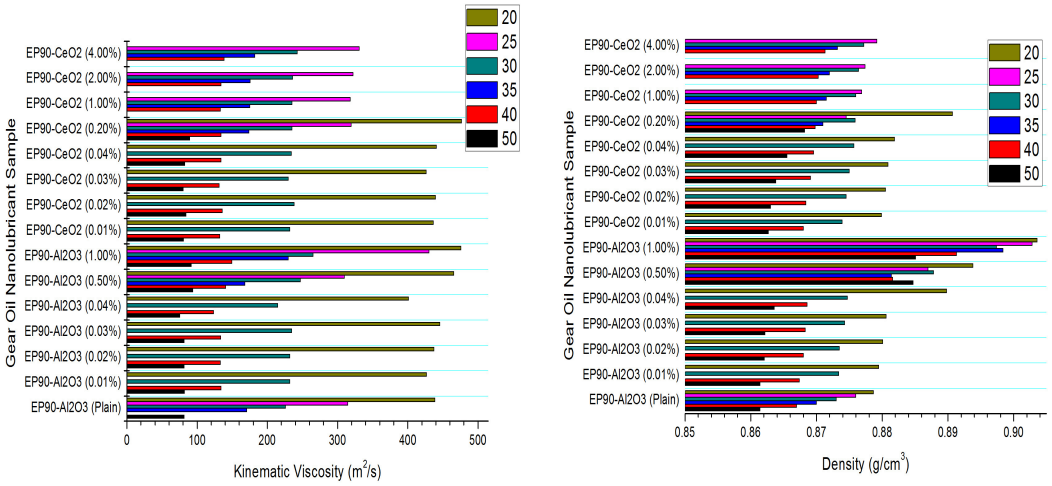
(b) Density

Figure 4.3: Viscosity and Density of transmission oil nanolubricant

## 4.3 Results and Discussion

Figure 4.3a shows the viscosity variation for transmission oil nanolubricants with $Al_2O3$ and $CeO_2$ nanoparticles. It can be clearly observed from Figure 4.3a that viscosity of nanolubricants decreases with increase in temperature which is due to weakening of intermolecular force of attraction in between base fluid molecules. Also, there is increase in relative movement in nanoparticle and base fluid molecules with rise in temperature. Viscous friction reduces with decrease in the viscosity, but it also limit load carrying ability of the fluid. The dispersion of $Al_2O_3$ cause initial increment (1.3% at 40°C & 0.01% $Al_2O_3$) in viscosity, but with further minor increment in particle volume fraction, viscosity decreases (4.5% at 40°C & 0.02% $Al_2O_3$). The reason for this viscosity is nanoparticles coming between the lube oil layers leading to the ease of relative movement between nanolubricant layers [172]. However further increasing nanoparticles beyond a certain limit, there is increment (1.5% at 40°C & 0.5% $Al_2O_3$) in viscosity with particle volume fraction. It can be observed that dispersion of $CeO_2$ nanoparticles causes continuous increment in viscosity, which is due to its plate shaped morphology. The variation in density for transmission oil nanolubricants with $Al_2O_3$ and $CeO_2$ nanoparticles is shown in Figure 4.3b. It can be noted that density of nanolubricants decreases the increase in temperature.

Figure 4.4a shows the viscosity variation for gear oil nanolubricants with $Al_2O_3$ and $CeO_2$ nanoparticles. Here also, it can be clearly observed from Figure 4.4a that viscosity of nanolubricants decreases with increase in temperature. Gear oil is beaing a dense oil, it



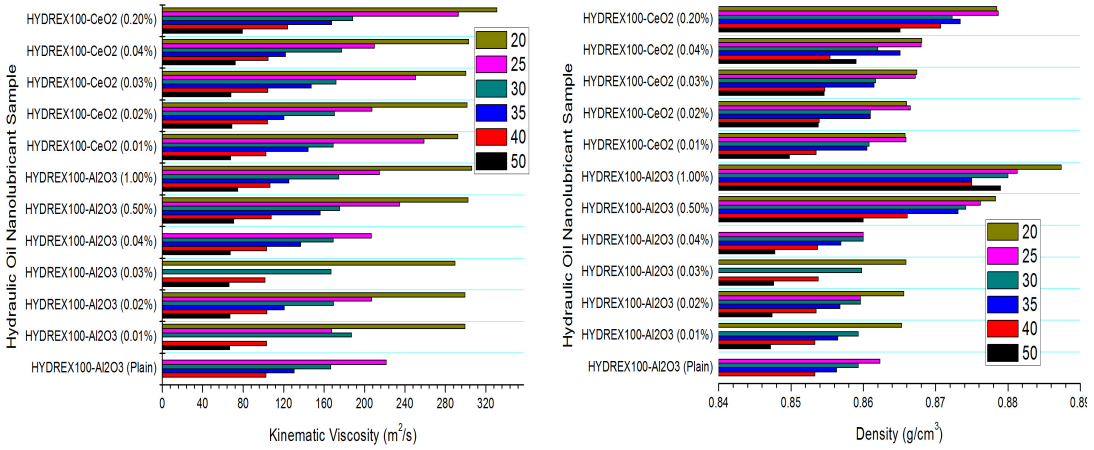

(a) Viscosity (b) Density

Figure 4.4: Viscosity and Density of gear oil nanolubricant

shows only minor variation in viscosity at lower particle volume fraction (2.6% at 30°C & 0.01% $Al_2O_3$; 2.7% at 30°C & 0.02% $Al_2O_3$; 2.67% at 30°C & 0.01% $CeO_2$; 3.7% at 30°C & 0.02% $CeO_2$). The variation in density for gear oil nanolubricants with $Al_2O_3$ and $CeO_2$ nanoparticles is shown in Figure 4.4b. It can be noted that there is 2.8% increment in density with 1% volume fraction of $Al_2O_3$ nanopartices, where as 0.48% increment in density reported for 4% volume fraction $CeO_2$ nanoparticles. The lesser variation in viscosity with the dispersion of $CeO_2$ nanoparticles is due to its higher density 7.132 $g/cm^3$ as compared to $Al_2O_3$ nanopartices 3.965 $g/cm^3$.

Figure 4.5a shows the viscosity variation for hydraulic oil nanolubricants with $Al_2O_3$ and $CeO_2$ nanoparticles. Again, it can be observed from Figure 4.5a that viscosity of nanolubricants decreases with increasing temperature. The viscosity of hydraulic oil nanolubricants increase with dispersion nanoparticles at higher particle volume fraction. The variation in density for hydraulic oil nanolubricants with $Al_2O_3$ and $CeO_2$ nanoparticles is shown in Figure 4.5b. It can be observed that density of nanolubricants decreases with increase in temperature.

Further, the result analysis of the four machine learning techniques used for the prediction of kinematic viscosity in each category has also been discussed in this section. For the purpose of finding the effectiveness of machine learning techniques in the prediction of kinematic viscosity, the experimental data collected in each category (gear oil, hydraulic oil and transmission oil) are distributed into training and testing dataset. The ratio of split of training and testing dataset for each category is 70:30 i.e. training the machine




(a) Viscosity (b) Density

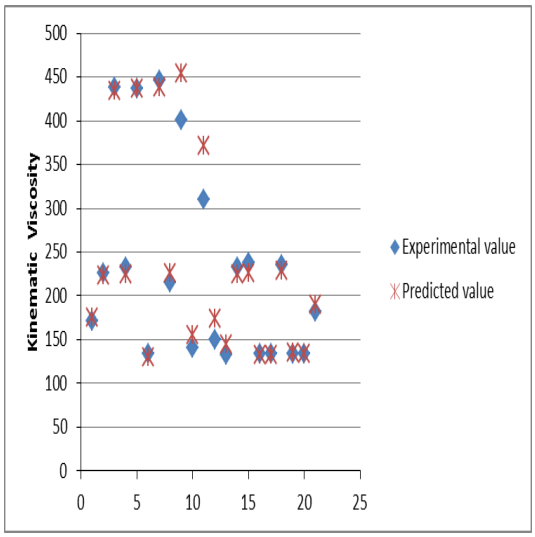
Figure 4.5: Viscosity and Density of hydraulic oil nanolubricant

Table 4.1: Performance comparison of machine leaning methods in prediction of kinematic viscosity of gear oil, hydraulic oil and transmission oil

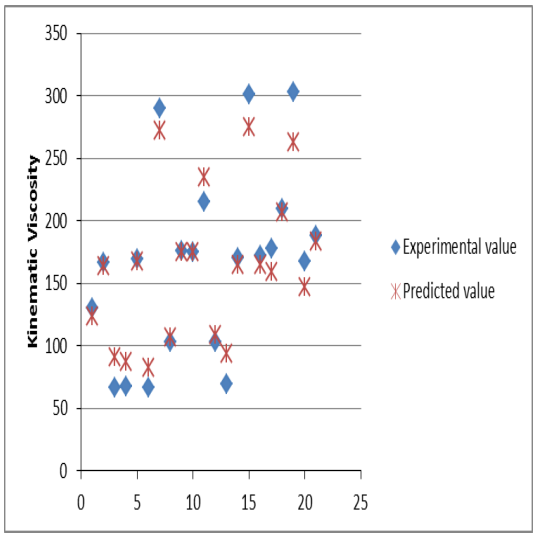
| Model | Gear oil (EP90) | | | | Hydraulic oil (Hydrex100) | | | | Transmission oil (SAE30) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | r | $R^2$ | MAE | RMSE | r | $R^2$ | MAE | RMSE | r | $R^2$ | MAE | RMSE |
| **Decision Tree** | 0.955 | 0.912 | 31.211 | 36.879 | 0.915 | 0.837 | 26.632 | 29.059 | **0.928** | **0.861** | **16.865** | **19.508** |
| **Random Forest** | 0.972 | 0.944 | 19.737 | 27.613 | **0.985** | **0.971** | **12.842** | **16.664** | 0.713 | 0.508 | 27.377 | 38.056 |
| **Generalised Linear** | 0.963 | 0.927 | 38.056 | 41.145 | 0.946 | 0.894 | 23.531 | 26.554 | 0.984 | 0.968 | 27.824 | 29.920 |
| **Neural Network** | **0.986** | **0.973** | **11.802** | **19.746** | 0.983 | 0.966 | 13.398 | 17.430 | 0.917 | 0.842 | 16.457 | 24.311 |

learning models on the 70% of the experimental data and then predicting the kinematic viscosity of the remaining 30% data, then comparing the predicted values with the actual values in order to evaluate the efficiency of the trained models. Finally, the results of model evaluation parameters have shown that the best predictive model selected for each category of nanolubricants supports the fact that machine learning models are ready for production purpose to predict the kinematic viscosity of the untested lubricants by providing the data with same parameters on which they have been trained. The efficiency of the trained models is evaluated on the basis of evaluation parameters discussed in Section 5.4.1. Table 4.1 show the average of model evaluation parameters after 5 fold cross validation of all four machine learning techniques in prediction of kinematic viscosity of gear oil, hydraulic oil and transmission oil. Figure 4.6 shows the comparison between the experimental (observed) values and predictive values of the best machine learning technique in each category.

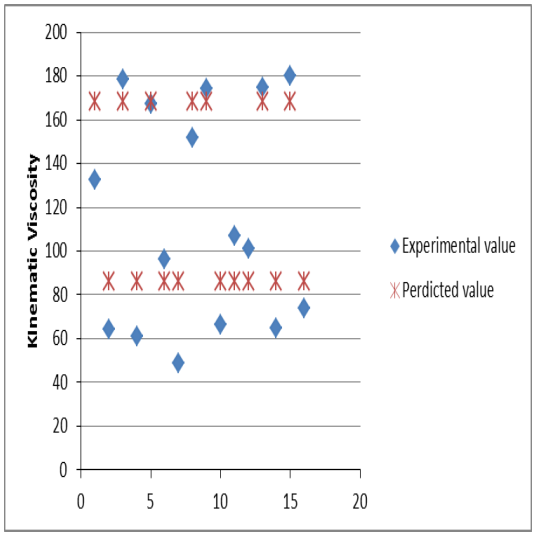
The correlation value shows the statistical relationship between the actual (experimental values) and predicted values by machine learning techniques. In case of gear oil the correlation values are 0.95, 0.97, 0.96 and 0.98 for decision tree, random forest, linear




(a) Neural network for gear oil nanolubricant




(b) Random forest for hydraulic oil nanolubricant




(c) Decision tree for transmission oil nanolubricant

Figure 4.6: Predicted VS Actual results for gear, hydraulic and transmission oil nanolubricant

Table 4.2: TOPSIS score and rank of the machine learning methods in each category

| Model | Gear oil | | Hydraulic oil | | Transmission oil | |
|---|---|---|---|---|---|---|
| | **Score** | **Rank** | **Score** | **Rank** | **Score** | **Rank** |
| **Decision Tree** | 0.242 | 3 | 0.000 | 4 | **0.865** | **1** |
| **Random Forest** | 0.676 | 2 | **1.000** | **1** | 0.019 | 4 |
| **Generalised Linear Model** | 0.015 | 4 | 0.226 | 3 | 0.532 | 3 |
| **Neural Network** | **1.000** | **1** | 0.951 | 2 | 0.777 | 2 |

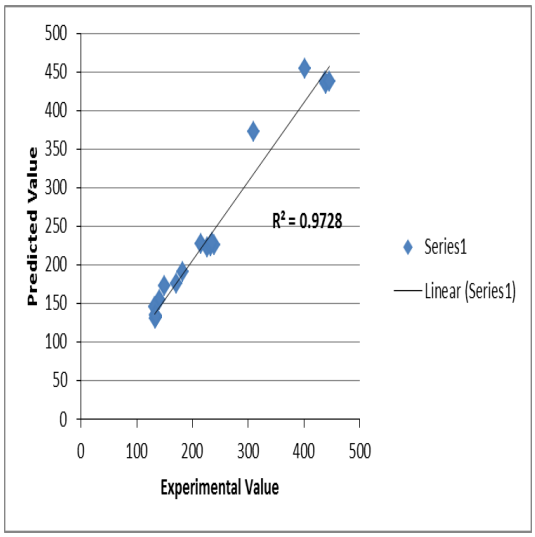
model and artificial neural networks respectively. So it is clearly seen that artificial neural network model gives the highest correlation value in case of gear oil which is reasonably acceptable. On the other hand for hydraulic oil and transmission oil the highest correlation value is provided by random forest model and decision tree model (i.e. 0.98 and 0.92) respectively.

The coefficient of determination ($R^2$) is another important parameter for evaluating the performance of the regression models. It determines how close the predicted data values are to the fit regression line. Figure 4.7 shows the $R^2$ value and fit regression line of best predictive model in each category. The artificial neural network, random forest and decision tree model gives the best $R^2$ value of 0.97, 0.97 and 0.86 for gear oil, hydraulic oil and transmission oil respectively.

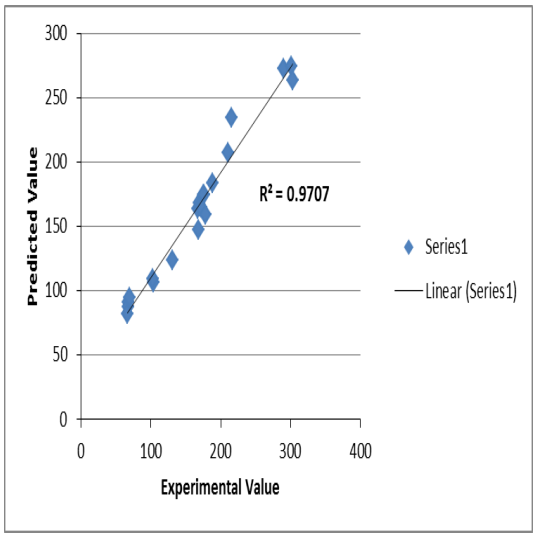
Further, for finding the best predictive model in each category ranking of the machine learning models is done on the basis correlation, $R^2$, MAE and RMSE values, by the use of MCDM technique TOPSIS. Table 4.2 shows the TOPSIS score and rank of machine learning models used. The neural network model, which is found to be the best predictive model in case of gear oil provides the least MAE and RMSE values of 11.80 and 19.74 respectively. For hydraulic oil random forest is ranked as a best predictive model, provides the MAE and RMSE values of 12.84 and 16.64 respectively. Similarly, for transmission oil decision tree model is ranked as a best predictive model, provides the MAE and RMSE value of 16.84 and 19.50 respectively.

## 4.4 Conclusion

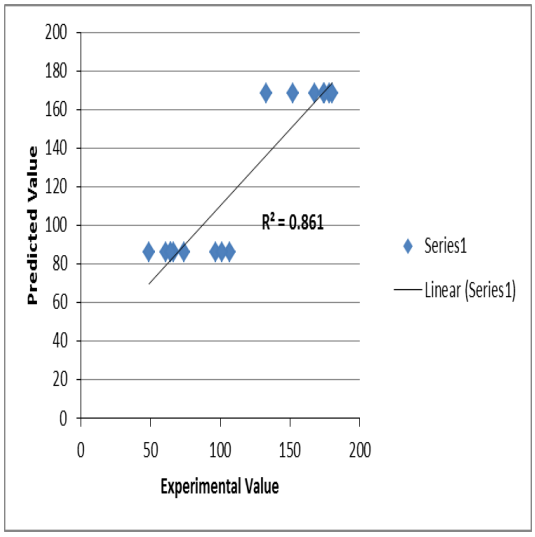
In the present study, the effectiveness of four machine learning techniques for viscosity prediction of transmission oil, hydraulic oil and gear oil nanolubricant has been visualized. Base fluids are widely used lubricants in HEMM. The morphology of the dispersed nanoparticles is analyzed by FESEM micrographs and it has been observed that $Al_2O_3$ nanoparticles are spherical in shape, whereas $CeO_2$ is of plate shaped. Test samples are prepared by the dispersion of nanoparticles are base oil and thermophysical properties




(a) Neural network for gear oil nanolubricant




(b) Random forest for hydraulic oil nanolubricant




(c) Decision tree for transmission oil nanolubricant

Figure 4.7: Correlation graph of best predictive method for gear, hydraulic and transmission oil nanolubricant

are tested at varying temperatures. Further, the four machine learning techniques are employed to predict viscosity of transmission oil, hydraulic oil and gear oil based nanolubricant. All the machine learning models are trained on 70% of the experimental data is used. The remaining 30% is used as the test data for evaluating the efficiency of the four machine learning technique in each category of nanolubricant. Finally, the MCDM based technique TOPSIS is used on the model evaluation results to find the best predictive method in each category. Following are the key findings of present work:

- Density of all samples increases with particle volume fraction and decreases with temperature.
- Variation in the viscosity of the samples is a function of particle volume fraction and size. Spherical morphology of $Al_2O_3$ nanoparticles makes decrement in viscosity at lower particle volume fraction.
- NN, RF and DT come out to be best predictive method for gear oil, hydraulic oil and transmission oil nanolubricant respectively.

There is still plenty of work for further development. Researchers can use similar machine learning models for prediction of tribological properties of nanolubricants. Further, the work can also be extended in developing the ensemble or hybrid machine learning models for various other nanolubricants.

# Chapter 5

# Multilevel ensemble classifier for particle physics Big Data

In high-energy physics the modern particle colliders like Large Hadron Collider and Tevatron produces a huge amount of experimental data that exceeds to petabytes in a period of a year. High-energy physicists need to explore this huge amount of data as it can lead to new scientific breakthrough related to origin of the universe. The data generated by particle colliders is quite fruitful source for the discovery of exotic particles. To track down these exotic particles from the huge volume of generated data is a trivial problem. High-energy physicists are trying hard through different machine learning approaches to single out these particles from the huge decay products. However, they are finding difficulty in efficiently training the machine learning models on the available data due to the complexity involved in consequential variation in decaying process. Recently the deep learning techniques have also been implemented in this area that showed some improved results than traditional machine learning techniques with accuracy upto 88%. In this work a novel multilevel ensemble machine learning technique has been developed for differentiating the signal and background in such huge complex data. The proposed technique is capable of dealing with this huge volume and complexity of data more efficiently than the existing techniques with the accuracy as high as 98.57%. Moreover for timely and faster classification of this huge volume of experimental data the most popular Big Data platform Apache Spark has been used for the deployment of the proposed technique.

## 5.1 Introduction

The tremendous growth of data in the past decade exacerbated data intensive computing problems commonly known as Big Data problem [1]. It is observed that 90% of the total data globally accumulated in the past five years. The rapid data growth is making trouble to people in variety of sectors and one such field is the high-energy or particle physics. It aims to study the elementary constituents of matter [173]. The data generated by the high throughput scientific experiments primarily from modern accelerators is huge

in volume and have high velocity. This poses the significant challenge to traditional statistical techniques and machine-learning tools for gaining the insights from this data [173]. The primary goal of these high-throughput experiments is to allow the scientists to explore the fundamental nature of matter and the laws regulating its interactions in order to gain a better understanding of the universe [174]. The modern accelerators like Large Hadron Collider (LHC) [175] [176] and Tevatron [177] [178] can cause head to head collision of two beams of protons and/or antiprotons. This collision results in the creation of exotic particles occurring at high-energy densities of 13 TeV or more. The particles generated at such high-energy densities are the detected by the particle detectors present in the particle colliders. The properties of these particles are then measured and observed by high-energy physicists so as to gain deeper insights about the very nature of matter. For the purpose of gaining better and fruitful insights physicists make use of sophisticated statistical and machine-learning techniques [179] [180].

Furthermore, the discovery of a new particle involves a trivial signal-versus-background classification problem. It has been seen that wide majority of particle collisions does not result in production of exotic particles. For an instance, in LHC there are approximately $10^{11}$ collisions per hour and out of those Higgs boson is produced in approximately 300 collisions on an average [173]. Therefore, to differentiate the collisions that generates particles of interest (signal) from collisions generating other particles (background) requires a powerful data analysis techniques [181, 182].

Moreover, data generated from the collisions ranges to petabytes per year so there is a need for faster and efficient analytical tools and techniques [33] that can directly boost particle discovery potential of the particle colliders. The work has been done in [173] using a deep neural networks to search the exotic particles in two benchmark datasets HIGGS and SUSY. The authors have used the deep neural network consisting of five layers with 300 hidden units in each layer to classify the signal and background in two benchmark datasets and have achieved better results in comparison to shallow neural networks and boosted decision tress. In this work an efficient multilevel ensemble classifier to classify the huge collision data in signal and background is proposed and developed on a distributed computing platform. For the purpose of developing the multilevel ensemble classifier in a distributed environment cloud based cluster platform has been used.

Apache Spark, an efficient Big Data framework is used for developing ensemble classifier. The efficiency of the ensemble classifier has been verified using two benchmark datasets, HIGGS and SUSY. Further this paper includes brief description of machine learning techniques, processing platform and datasets used in this work in Section 5.2. The description of the proposed technique has been given in Section 5.3. Section 5.4

carries the details of the model evaluation parameters used for evaluating the efficacy of the proposed technique, discussion about the results obtained and the comparison with the existing techniques. Finally the outcome of this work has been concluded in Section 5.5.

## 5.2 Methods and Materials

This section includes the brief description of the particle physics benchmark datasets, the Big Data platform and implementation architecture used in this work.

### 5.2.1 Dataset and its features

HIGGS Dataset: The HIGGS dataset is downloaded from [183]. The dataset consists of 28 features in which the first 21 features are kinematic properties of the Higgs Bosons measured by the particle detectors in the accelerator and are termed as low-level features. The next 7 features are termed as high-level features and are the function of the first 21 features. It has been considered as the benchmark classification to classify the particles generated by simulated collision into theoretical Higgs Bosons (signal) and the decay products (background) with different kinematic features. The dataset consist of 11 million instances [173] and is of capacity 7.48 GB.

SUSY Dataset: The SUSY dataset is downloaded from [184]. The dataset consists of 18 features in which the first 8 features are kinematic features measured by the particle detectors in the accelerator and are termed as low-level features. The last 10 features are termed as high-level features and are the function of the first 8 features. The dataset consist of 5 million instances [173] and is of capacity 2.22 GB.

### 5.2.2 Processing Platform and Implementation Architecture

Apache Spark: Apache Spark is the one of the popular in-memory cluster computing platform developed by the researchers at AMP labs of University of California, Berkeley campus. It is used for faster and interactive data analytics. Spark with its approach of Resilient Distributed Dataset (RDD) and in-memory computation has been actively adopted by the people from industry and research community [7]. It abstracts APIs in Python, Java, Scala, SQL and R. It integrates well with other tools falling under Big Data umbrella for example Spark can run on the top of Hadoop clusters and access data from any of the Big Data storage systems like HDFS, Cassandra, MangoDB and HBASE. Spark has its own cluster manager known as standalone scheduler and it can also work

in conjunction with other resource managers like YARN and MESOS [2]. Apache Spark has a super-active community of its contributors and in 2015 RDD API was extended to include dataframes to allow its users to group the distributed data into columns as in relational databases [185]. The best example is that an RDD of key-value pairs when converted to a dataframe is represented as a table with different columns for key and value.

Spark ecosystem compromises of several components that are tightly integrated. Spark core is the fundamental component of Spark ecosystem. It monitors, distributes and schedules the tasks across the cluster for the applications running on Spark cluster. With its inherent speed Spark's core engine empowers its other components designed for variety of workloads, like machine learning, SQL and graph based computation. Figure 5.1 shows the various components of Spark's ecosystem [185]. All components in Spark's ecosystem are tightly integrated to provide various benefits like building of applications that can combine several processing models. For example in Spark it is possible to develop an application that can use machine learning technique to classify data available from streaming sources [185, 186].

In this work Apache Spark machine learning library MLlib is used for developing a scalable multilevel ensemble classifier for searching the exotic particles in huge particle physics datasets.

A typical Spark cluster has two major components, Master node (Driver program) and Worker nodes (Executors). Node refers to the single machine in the cluster. The node on which the Spark context is created in the cluster becomes the driver program (Master node) which in consultation with cluster manager (that could be the Spark's own standalone cluster manger or external cluster managers like YARN or MESOS) assigns jobs to the other nodes in the cluster that serve as worker nodes [186] [185]. However, for this work a unified Apache Spark data analytics platform Databricks [187] in integration AWS cloud infrastructure is used. For the purpose of data storage Amazon Simple Scalable Storage(S3) has been used [188]. Spark cluster architecture used for the deployment of the ensemble classifier is shown in Figure 5.2. An auto scalable cluster of nodes ranging from two to eight nodes is used. For the driver node amazon EC2 instance r4.2xlarge having 61 GB of main memory, 8 cores and 1 Databricks Unit (DBU) and r4xlarge having 30.5 GB of memory, 4 cores and 1 DBU for worker nodes are used.

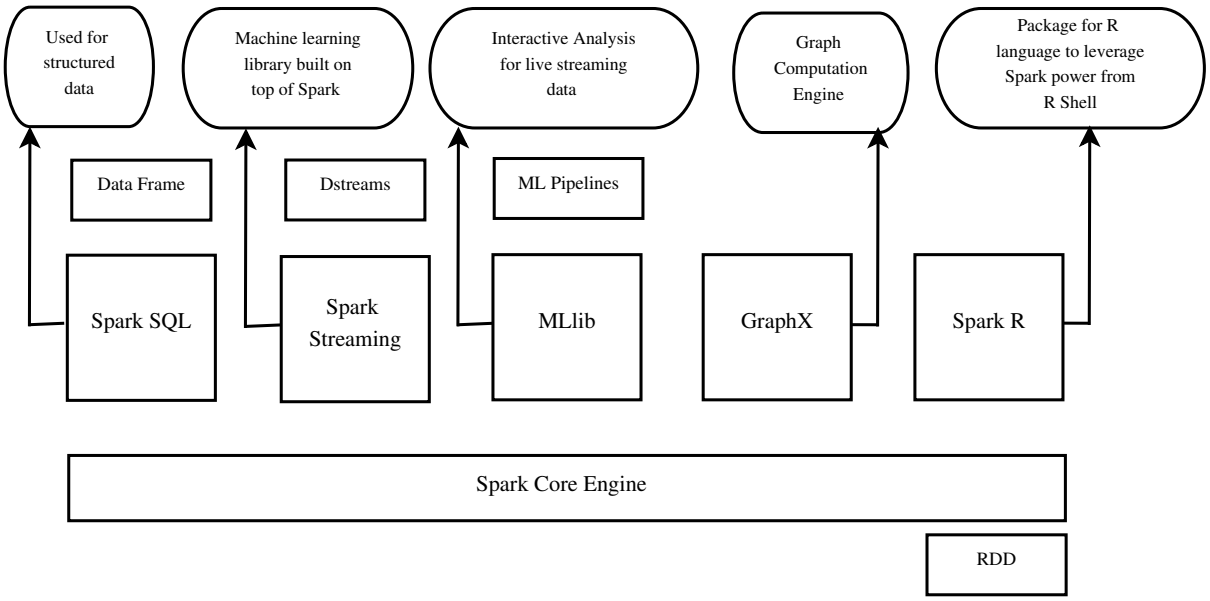



Figure 5.1: Spark Ecosystem

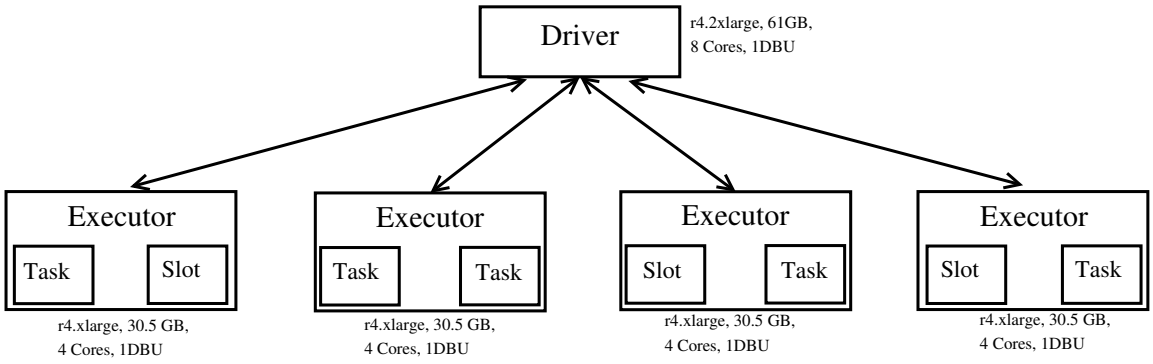



Figure 5.2: Spark Architecture

Table 5.1: Machine Learning techniques and their tuning parameters used

| **Technique** | **Method** | **Tuning Parameters** |
|---|---|---|
| Support Vector Machine | LinearSVC | maxIter=10, regParam=0.1 |
| Decision Tree | dt | maxDepth=5, minInstancesPerNode=20, maxMemoryInMB=256, impurity= "gini" |
| Gradient Boosted Tree | gbt | maxDepth=5, maxBins=32, minInstancesPerNode=1, maxMemoryInMB=256, lossType='logistic', maxIter=20 |
| Multilayer Perceptron | MultilayerPerceptron | maxIter=100, layers=4, blockSize=128, seed=1234, solver='l-bfgs' |

### 5.2.3 Machine Learning Techniques

This section carries the brief overview of the various machine learning techniques used to build the proposed ensemble classifier. Table 5.1 shows various tuning parameters used for each machine learning technique. The four machine learning techniques used in this work are as follows:

**i.** Support Vector Machine (SVM): In SVM hyperplane or set of hyperplanes is created in a high dimensional space, for the purpose of classification or regression task. The hyperplane with greater functional margin (i.e. the largest distance to the nearest training-data points of a particular class) is selected as the final hyperplane for classifier, as greater the margin lower will be the generalization error of the classifier [24]. In Spark MLlib linear SVM classifier(LSVC) supports binary classification and uses the Orthant-Wise Limited-memory Quasi-Newton (OWLQN) optimizer [189] for optimizing the hinge loss function.

**ii.** Decision Tree (DT): The decision tree algorithm follows a greedy approach to partition the feature space using a recursive binary partitioner. In decision tree algorithm same label is predicted for the bottommost (leaf) partition and for that each partition is greedily chosen from a set of possible splits by selecting the best split, so as to maximize the information gain at a tree node [22]. For an instance at each tree node the split is chosen from the set $argmax_s$IG(D,s); where IG(D,s) is the information gain when a split s is applied to a dataset D.

**iii.** Gradient Boosted Tree (GBT): GBTs [190] [191] are the ensemble of decision tree that iteratively trains a sequence of decision trees in order to minimize the loss function. GBTs can efficiently deal with the categorical features even without feature scaling. GBTs are also capable of capturing non-linearities and feature

interactions. GBTs predicts the label of each training instance in a stage-wise fashion similar to other boosting methods, and then it generalizes them by allowing optimization of an arbitrary differentiable loss function. For the classification the loss function currently supported by GBTs in Spark MLlib is log loss (Equation 5.1) function [185].

$$LogLoss = 2\sum_{i=1}^{N} log(1 + exp(-2y_i F(x_i))) \tag{5.1}$$

where $N$ is number of instances, $y_i$ is label of instance i, $x_i$ is features of instance and $F(x-i)$ is predicted label for instance i.

**iv.** Multilayer Perceptron (MLP): MLP in Spark MLlib is the multilayer feed-forward back-propagation artificial neural network [142]. It consists of multiple layers of interconnected nodes and each layer is connected to its next layer in the network. The interconnected nodes in each layer are called processing units or neurons that make use of some activation function for transforming input to output. The first layer of MLP is the input layer and represents the input data. The nodes in the other layers or intermediate layers maps inputs to outputs by the use of linear combination of inputs with weight $w$, bias $b$ and applying some activation function. In Spark MLlib the nodes of the intermediate layer of MLP use sigmoid function (Equation 5.2) as an activation function while the output layer uses the softmax function (Equation 5.3)[185] .

$$f(z_i) = \frac{1}{1 + e^{-z_i}} \tag{5.2}$$

$$f(z_i) = \frac{e^{z_i}}{\sum_{k=1}^{N} e^{z_k}} \tag{5.3}$$

## 5.3 Proposed Multilevel Ensemble Technique

The ensemble methods are in general used to increase the predictive power of the classifier so as to deal with the worst case scenario. The proposed the multilevel ensemble technique is discussed in this section, which is later developed for efficient classification of signal and background for the benchmark datasets described in Section 5.2.1. In the proposed Multi-Level Ensemble (MLE) classifier four machine learning techniques have been used namely, Support Vector Machine (SVM), Decision Tree (DT), Gradient-Boosted Tree

(GBT) and Multi-Layer Perceptron (MLP). In this technique all the models are trained on 70% of data and the remaining 30% is used for testing. The technique is divided into three phases which are discussed as follows:

**i.** Phase 1: In this phase of the technique SVM classifier is being trained on 70% of data and the predictions are generated on the remaining 30%. After that the true predictions generated by the classifier are separated from the false predictions generated along with their actual labels.

**ii.** Phase 2: Here two classifiers, GBT and DT classifiers are used. GBT classifier is trained on the data with false prediction and DT classifier is trained on the true predictions generated in Phase 1. The split used for training and test data is 70:30 in both the cases. Thereafter, the process of separating the true predictions and false predictions made by two classifiers (i.e. is GBT and DT) is repeated again. Then we move to the Phase 3 of the technique.

**iii.** Phase 3: The Phase 3 of the technique starts with generating the dataframe by concatenating the true predictions generated in Phase 1 and false predictions generated by the two classifiers in Phase 2. Then the multilayer perceptron with 4 layers is trained on the combined dataframe and the final predictions are generated.

Using this approach, both the true predictions as well as false predictions are refined to increase the overall accuracy of the proposed classifier. In this technique false positive results (background considered as signal) at each level are dealt by inputting the true predictions of the previous level classifier to the next level classifiers. Since the data travels through all the four models the classifier learns the data in a perfect manner and hereby contributing to the robust and more accurate predictions. All the three phases discussed in this section are combined in a single machine learning pipeline using Apache Spark. Figure 5.3 shows the proposed machine learning pipeline .

## 5.4 Results

In this section the brief discussion about the results obtained by the ensemble classifier in terms of different model evaluation parameters and its comparison with the existing system is provided.

### 5.4.1 Model Evaluation Parameters

The brief description of the model evaluation parameters used for evaluating the proposed ensemble classifier are as follows:

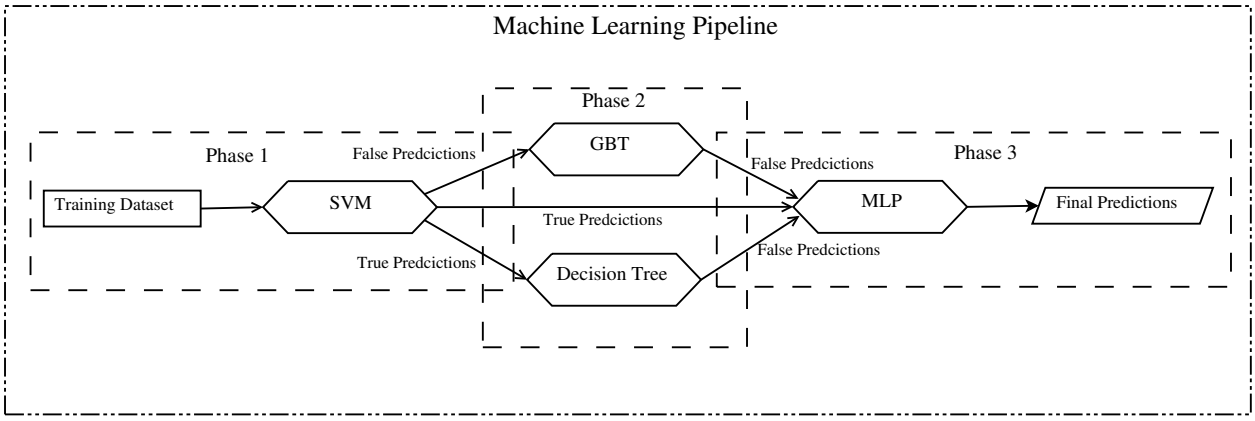



Figure 5.3: Proposed multilevel ensemble classifier

**i.** Accuracy: Accuracy is the measure of correctness of the model. It tells how precisely the model is able to predict the target value or class for the new data. Higher the accuracy more precise are the predictions made by the model. It can be computed as follows:

$$Accuracy = \frac{TP}{TP + FN} * 100 \tag{5.4}$$

**ii.** AUROC: Area Under ROC Curve (AUROC) [192] is another important parameter for measuring the efficiency of the classifier. ROC curve is generated by plotting the true positive rate of the classifier against its false positive rate. AUROC is just the area under the ROC curve. Its value ranges between 0 and 1. Higher the value of AUROC (i.e. closer to 1) better is the classifier. Equation 5.5 is used for calculating AUROC.

$$AUROC = \int_0^1 \frac{TP}{TP + FN} d(\frac{FP}{TN + FP}) \tag{5.5}$$

where; True Positive(TP) : when actual label or class is positive and is predicted as positive, True Negative(TN) : when actual label or class is negative and is predicted as negative, False Positive(FP) : when actual label or class is negative but predicted as positive and False Negative(FN) : when actual label or class is positive but predicted as negative.

**iii.** AUPRC: Another parameter which cannot be ignored while evaluating the efficiency of the classifier is area under precision recall curve (AUPRC). Precision-Recall curve [193] is generated by plotting precision against recall and AUPRC is the area under the PR curve. Similar to AUROC its value ranges between 0 and 1, higher it's value more efficient is the classifier. Generally this parameter is considered more reliable than AUROC in the class imbalance classification problems. Since in this work we are refining the false positives as well as true positives in three phases of the

ensemble classifier it is necessary to consider AUPRC for evaluating the efficiency of the ensemble classifier. Equation 5.6 is used to calculate the AUPRC.

$$AUPRC = \int_0^1 \frac{TP}{TP+FP} d(\frac{TP}{TP+FN}) \tag{5.6}$$

**iv.** K-fold cross validation: K-fold cross validation [146] is one of the most popular technique used for analysing the consistency and robustness of the predictive method. In K-fold cross validation, the training dataset is split into k sub-samples of equal size. Then out of the k sub samples the k-1 sub samples are used as the training data for training the model and the remaining one sub sample is used as a testing data in each of the k iteration in such a manner that each of the k sub-sample is used exactly once as the validation or testing data. Finally the results generated in k iterations can be averaged for taking the estimate of the mean accuracy of the predictive method or the k results produced can be plotted on graph (line plot or box plot) for visualizing the fluctuation or variation in parameter values of the predictive method or classifier. The method with less variation is considered as more consistent method.

### 5.4.2 Discussion and Comparison

This section carries the discussion about the results obtained by the proposed MLE classifier and its comparison with the existing techniques on the basis of model evaluation parameters discussed in Section 5.4.1. In terms of accuracy the MLE classifier has achieved the mean accuracy of 97.82% and 98.57% for HIGGS and SUSY benchmark datasets respectively. The mean AUROC values for the MLE classifier are 0.963 and 0.984 respectively which is higher than the mean AUROC values achieved in [173]. Table 5.2 shows the comparative analysis of the AUROC values of the MLE classifier with existing techniques used in [173]. In terms of AUPRC values the MLE classifier scores the value of 0.985 and 0.967 for HIGGS and SUSY benchmark datasets respectively and this parameter is not considered in [173]. Table 5.4 shows the mean accuracy, AUC and AUPRC values achieved by the MLE classifier. Since the Big Data platform Apache Spark is used for the implementation of the MLE classifier the another important parameter that cannot be ignored for evaluating the efficiency is the runtime for training the model and generating the predictions. For evaluating the runtime efficiency of the MLE classifier both datasets HIGGS and SUSY are split into number of small files which are the subset of the complete dataset. The process of splitting the datasets is repeated from 50000 instances split file to complete instances for both HIGGS and SUSY datasets.

Table 5.2: Comparative results of AUROC values for MLE classifier with other techniques

| **Technique** | **AUROC** | |
|---|---|---|
| | HIGGS | SUSY |
| Boosted Decision Tree | 0.81 | 0.863 |
| Neural Network | 0.816 | 0.875 |
| Deep Neural Network | 0.885 | 0.876 |
| **MLE Classifier** | **0.963** | **0.984** |

Table 5.3: Runtime evaluation of different splits of datasets

| **No. of rows in Dataset** | **File size in KB** | | **Runtime in minutes** | | **Cluster Size** |
|---|---|---|---|---|---|
| | HIGGS | SUSY | HIGGS | SUSY | |
| 50 K | 35670 | 23343 | 1.1 | 1.1 | 3 Nodes |
| 100 K | 71339 | 46686 | 1.43 | 1.3 | |
| 1000 K | 713380 | 170954 | 8.59 | 3.7 | |
| 2500 K | 1783447 | 1167127 | 9.28 | 6.58 | 8 Nodes |
| Complete dataset | 7847166 | 2334256 | 14.3 | 10.68 | |

Table 5.3 lists the number of rows, size of the file in bytes and the corresponding time taken by the MLE classifier to generate predictions. For the split files with number of rows upto 1000k three nodes cluster and for the split files more than 1000k rows eight nodes cluster have been used in this work.

To verify the consistency and robustness of the MLE classifier 5 fold cross validation is carried out on complete datasets. The MLE classifier shows the consistent performance for both the datasets in terms all model evaluation parameters. Figure 5.4 shows the 5 fold cross validation results in terms of accuracy for HIGGS and SUSY datasets. For HIGGS dataset MLE classifier's accuracy ranges from 96.82% to 98.5% and for SUSY it ranges from 98.2% to 98.89%. Figure 5.5 shows the AUROC cross valiadtion results for both the datasets. In terms of AUROC the values ranges from 0.949 to 0.974 and 0.98 to 0.987 respectively for HIGGS and SUSY datasets. Figure 5.6 shows the AUPRC cross validation results for the datasets. The AUPRC value ranges from 0.977 to 0.992 and 0.965 to 0.969 respectively for HIGGS and SUSY datasets. From Figures 5.4, 5.5 and 5.6 it is clearly seen that MLE classifier shows consistent performance for both the datasets.

Table 5.4: Mean values of model evaluation parameters for ensemble classifier

| **Evaluation Parameter** | **HIGGS** | **SUSY** |
|---|---|---|
| Accuracy (%) | 97.82 | 98.57 |
| AUROC | 0.963 | 0.984 |
| AUPRC | 0.985 | 0.967 |

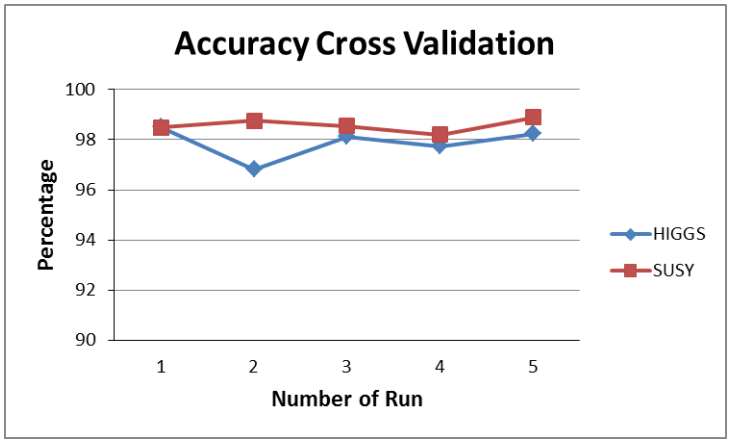



Figure 5.4: K-fold cross validation of accuracy

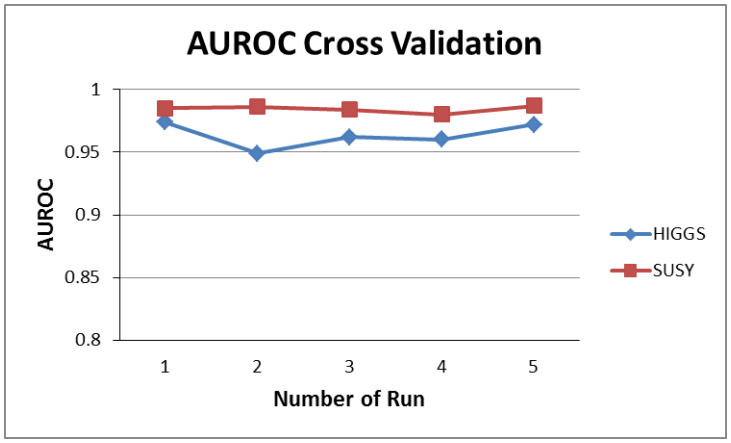



Figure 5.5: K-fold cross validation of AUROC

Moreover it is also seen that MLE classifier shows better results in terms of accuracy and AUROC for SUSY dataset, while for HIGGS datasets AUPRC values are higher than SUSY.

## 5.5 Conclusion

Considering the need of efficient machine learning techniques in the field of high-energy physics that can enhance the search of exotic particles in huge experimental data generated by the particle colliders an effort has been made in this direction using four machine learning techniques. A multilevel ensemble classifier has been developed that is capable of searching the exotic particles in two huge particle physics benchmark datasets generated from the Monte Carlo simulations in LHC. The MLE classifier developed has shown better and efficient results in terms evaluation parameters than the existing techniques implemented so far. The proposed MLE classifier classifies the data into signal and back-

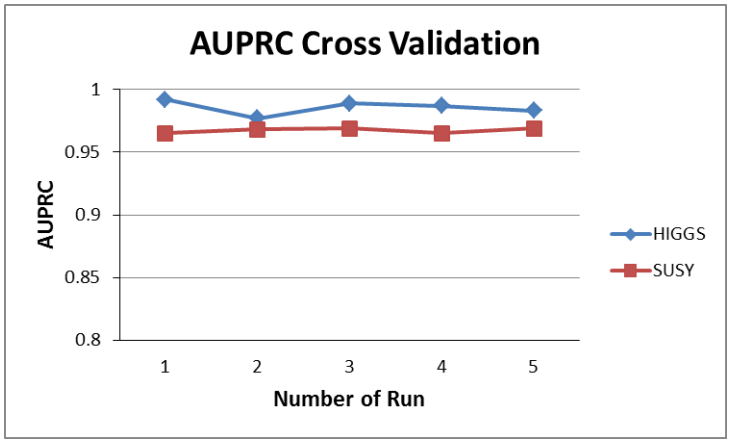



Figure 5.6: K-fold cross validation of AUPRC

ground in three phases. In each phase the true predictions and false predictions are generated which are further refined in the next phase of the classifier and thereby contributing in achieving the overall accuracy of 97.82% and 98.57% for HIGGS and SUSY datasets respectively. Furthermore the ensemble classifier has also achieved the excellent result in terms of parameters like AUROC and AUPRC.

# Chapter 6

# Conclusions and Future Works

This chapter is the concluding part of the thesis and proposes some suggestions towards which the present work can be further extended. Section 6.1 brings out the overall conclusions of the research work carried out in this thesis, and in section 6.2 suggestions regarding the future research directions and possible extensions of the work presented in the thesis are made.

## 6.1 Conclusion

In this research work, hybrid and ensemble machine learning approaches are proposed and developed to solve Big Data predictive analytics problems in bioinformatics, material science, and particle physics. This research work is motivated by the rise of a new scientific paradigm known as data-intensive scientific discovery in many sectors like healthcare, natural sciences, astronomy, bioinformatics, e-commerce, banking, social networking, and many more. One of the major reasons for this paradigm shift is the advent of digital systems in almost every field of our daily life, leading to a huge volume of data in some sectors. More interestingly, most of the generated data in the present day is semi-structured or unstructured data. In several sectors like bio-informatics, material sciences, and particle physics, the use of these data-intensive approaches is still challenging due to the nature and volume of data. Sometimes the traditional machine learning techniques seem to behave in a limited fashion in the knowledge discovery process due to the complexity of Big Data.

In this work, some research problems were identified in bioinformatics, material sciences, and particle physics; they require some innovative machine learning solutions. The hybrid and ensemble solutions developed in this work make use of unsupervised, supervised, heuristic, and greedy approaches to provide an accurate and efficient solution for the problem at hand. Moreover, we have also taken the leverage of the most popular Big Data platform, Spark, for dealing with the scalability of the ensemble technique developed in this work. One of the works' multi-criteria decision-making technique has also been used to find the optimal predictive analytics technique for the problem. The developed

techniques' efficiency was evaluated on various model evaluation parameters like precision, recall, AUROC, AUPRC, RMSE, etc. Comparative analysis of the model evaluation results is also done with the benchmark techniques available for that domain.

HTLVs type prediction (Case Study 1): In this work, the 64 hybrid machine learning methods are developed and tested for the prediction of different type of human t-cell lymphotropic virus (HTLV) in a high dimensional dataset of 292 features extracted from the protein sequences of HTLVs (HTLV-1, HTLV-2, and HTLV-3), non-HTLV and their similar proteins. The dataset is firstly labeled using the K-means clustering algorithm; then, the feature weighting is done to identify the important features for training the machine learning models. Finding the optimal features to serve as an input to train the models is done using two heuristic search and two greedy search techniques. Finally, the models are trained using the optimal features and are evaluated based on model accuracy, recall (TPR), specificity (TNR), precision (PPV), negative predicted value (NPV), AUROC value, and F1 score. Furthermore, the robustness of the best models in each category is explored using 10-fold cross-validation.

Finally, based on the analysis of all the evaluation parameters, it is found that random-forest in combination random-forest importance and forward search is the most accurate and reliable predictive method among other methods developed in this work.The best hybrid model has been described as having outstanding clinical accuracy, AUROC value and F1 score of 99.85% from 0.99, and 0.99. This kind of method will assist the existing diagnostic system for the identification of HTLV-1 such that after a molecular diagnosis of HTLV by immunoassays like enzyme-linked immunoassay or particle agglutination assays, there is still a need for confirmatory tests like western blotting, immuno-fluorescence assay, or radio-immuno-precipitation assay to differentiate HTLV-1 from HTLV-2. This confirmatory procedures are very elaborate and require a variety of complicated steps. The proposed hybrid techniques enable the identification of the protein mixture in the actual solution.

Kinematic viscosity prediction of nanolubricants (Case Study 2): In this work, the effectiveness of four machine learning techniques for viscosity prediction of transmission oil, hydraulic oil, and gear oil nanolubricant has been visualized. Base fluids are widely used lubricants in heavy earthmoving machinery. The dispersion of nanoparticles prepares test samples are base oil, and thermophysical properties are tested at varying temperatures. Further, the four machine learning techniques are employed to predict transmission oil, hydraulic oil, and gear oil-based nanolubricant viscosity. All the machine learning models are trained on 70% of the experimental data is used. The remaining 30% is used as the test data for evaluating the efficiency of the four machine learning techniques in each

nanolubricant category. Finally, the MCDM based technique TOPSIS is used on the model evaluation results to find the best predictive method in each category. A neural network, random forest, and decision tree come out to be the best predictive method for gear oil, hydraulic oil, and transmission oil nanolubricant, respectively. Eventually, this study provides a new theoretical basis in nanolubricants for creating software programs that allow the user to know the lubrication oil efficiency to suppress the operating costs for heavy earthmoving machinery.

Multilevel ensemble classifier (Case Study3): Considering the need for efficient machine learning techniques in the field of high-energy physics that can enhance the search of exotic particles in huge experimental data generated by the particle colliders, an effort has been made in this direction using four machine learning techniques. A multilevel ensemble classifier has been developed to search the exotic particles in two huge particle physics benchmark datasets generated from the Monte Carlo simulations in LHC. The MLE classifier developed has shown better and efficient results in terms of evaluation parameters than the existing techniques. The proposed MLE classifier classifies the data into signal and background in three phases. In each phase, the true predictions and false predictions are generated, further refined in the next phase of the classifier, thereby contributing to achieving the overall accuracy of 97.82% and 98.57% for HIGGS and SUSY datasets, respectively. Furthermore, the ensemble classifier has also achieved excellent results in terms of parameters like AUROC and AUPRC.

## 6.2 Future Work

This work primarily focuses on developing hybrid and ensemble machine learning techniques for Big Data problems, but plenty of work can be taken into consideration for future development in this direction. Some of the suggestions for future work are as follow:

1. The web-based hybrid machine learning platform can be commercially developed to predict HTLV 1 to support existing diagnostic systems.

2. MCDM based machine learning techniques developed in this work can be used for kinematic viscosity prediction of several other nanolubricants by developing computer software that can work with a variety of nanolubricants data. This work can also be extended by replacing the machine learning techniques with suitable deep learning algorithms.

3. In recent years, several deep learning algorithms came into the limelight for solv-

ing Big Data problems, but they still face scalability and efficiency issues. The approaches adopted in this research work to build efficient machine learning techniques to improve the efficiency of deep learning techniques and address scalability issues.

# List of Publications

1. G Sharma, PS Rana and S Bawa, *"Hybrid machine learning models for predicting types of Human T-cell Lymphotropic Virus"*, IEEE/ACM Transactions on Computational Biology and Bioinformatics, IEEE, 18(4):1524-1534, 2021. [SCI, IF 3.71]
2. G Sharma, A Kotia, SK Ghosh, PS Rana, S Bawa and MKH Ali, *"Kinematic Viscosity Prediction of Nanolubricants Employed in Heavy Earth Moving Machinery using Machine Learning Techniques"*, International Journal of Precision Engineering and Manufacturing, Springer, 21(10):1921–1932, 2020. [SCI, IF 2.11]
3. G Sharma, PS Rana and S Bawa, *"Multilevel ensemble classifier for particle physics Big Data: Implementation using Apache Spark"*, Neurocomputing, Elsevier, [Under Review, IF 4.438]

# References


[1] CL Philip Chen and Chun-Yang Zhang. Data-intensive applications, challenges, techniques and technologies: A survey on big data. *Information sciences*, 275:314–347, 2014.

[2] Sara Landset, Taghi M Khoshgoftaar, Aaron N Richter, and Tawfiq Hasanin. A survey of open source tools for machine learning with big data in the hadoop ecosystem. *Journal of Big Data*, 2(1):1–36, 2015.

[3] Clifford Lynch. Big data: How do your data grow? *Nature*, 455(7209):28, 2008.

[4] Alex Szalay et al. Science in an exponential world. 2008.

[5] Apache Hadoop. https://hadoop.apache.org/.

[6] Apache Mahout. https://mahout.apache.org/.

[7] Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Murphy Mccauley, M Franklin, Scott Shenker, and Ion Stoica. Fast and interactive analytics over hadoop data with spark. *Usenix Login*, 37(4):45–51, 2012.

[8] Eric Savitz. Gartner: Top 10 strategic technology trends for 2013. *URL http://www. forbes. com/sites/ericsavitz/2012/10/22/gartner-10-critical-tech-trends-for-the-next-five-years*, 2012.

[9] Gordon Bell, Tony Hey, and Alex Szalay. Beyond the data deluge. *Science*, 323(5919):1297–1298, 2009.

[10] Doug Laney. 3d data management: Controlling data volume, velocity and variety. *META group research note*, 6(70):1, 2001.

[11] Amir Gandomi and Murtaza Haider. Beyond the hype: Big data concepts, methods, and analytics. *International journal of information management*, 35(2):137–144, 2015.

[12] Phillip Russom et al. Big data analytics, tdwi best practices report. *Fourth quarter*, pages 1–35, 2011.

[13] Karthik Kambatla, Giorgos Kollias, Vipin Kumar, and Ananth Grama. Trends in big data analytics. *Journal of parallel and distributed computing*, 74(7):2561–2573, 2014.

[14] James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, and Angela H Byers. Big data: The next frontier for innovation, competition, and productivity. 2011.

[15] Alexander Szalay and Jim Gray. 2020 computing: Science in an exponential world. *Nature*, 440(7083):413, 2006.

[16] Ewaryst Tkacz and Adrian Kapczynski. *Internet-technical development and applications*, volume 64. Springer Science & Business Media, 2009.

[17] Randal E Bryant. Data-intensive supercomputing: The case for disc. 2007.

[18] Michael I Jordan and Tom M Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, 2015.

[19] Eric WT Ngai, Yong Hu, Yiu Hing Wong, Yijun Chen, and Xin Sun. The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. *Decision support systems*, 50(3):559–569, 2011.

[20] Anand Rajaraman and Jeffrey David Ullman. *Mining of massive datasets*. Cambridge University Press, 2011.

[21] Adil Fahad, Najlaa Alshatri, Zahir Tari, Abdullah Alamri, Ibrahim Khalil, Albert Y Zomaya, Sebti Foufou, and Abdelaziz Bouras. A survey of clustering algorithms for big data: Taxonomy and empirical analysis. *IEEE transactions on emerging topics in computing*, 2(3):267–279, 2014.

[22] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[23] John Mingers. An empirical comparison of pruning methods for decision tree induction. *Machine learning*, 4(2):227–243, 1989.

[24] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[25] Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian network classifiers. *Machine learning*, 29(2):131–163, 1997.

[26] Jacek M Zurada. *Introduction to artificial neural systems*, volume 8. West publishing company St. Paul, 1992.

[27] Pavel Berkhin. A survey of clustering data mining techniques. In *Grouping multidimensional data*, pages 25–71. Springer, 2006.

[28] Anil K Jain, Richard C Dubes, et al. *Algorithms for clustering data*, volume 6. Prentice hall Englewood Cliffs, 1988.

[29] Leonard Kaufman and Peter J Rousseeuw. *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley & Sons, 2009.

[30] Jiawei Han, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.

[31] Jiawei Han, Micheline Kamber, and Anthony KH Tung. Spatial clustering methods in data mining. *Geographic data mining and knowledge discovery*, pages 188–217, 2001.

[32] Andrew McAfee, Erik Brynjolfsson, Thomas H Davenport, DJ Patil, and Dominic Barton. Big data: the management revolution. *Harvard business review*, 90(10):60–

68, 2012.

[33] Omar Y Al-Jarrah, Paul D Yoo, Sami Muhaidat, George K Karagiannidis, and Kamal Taha. Efficient machine learning for big data: A review. *Big Data Research*, 2(3):87–93, 2015.

[34] Dinkar Mylaraswamy, Brian Xu, Paul Dietrich, and Anandavel Murugan. Case studies: Big data analytics for system health monitoring. In *Proceedings on the International Conference on Artificial Intelligence (ICAI)*, page 1. The Steering Committee of The World Congress in Computer Science, Computer . . . , 2014.

[35] Roshan Sumbaly, Jay Kreps, and Sam Shah. The big data ecosystem at linkedin. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, pages 1125–1134. ACM, 2013.

[36] Jiang Zheng and Aldo Dagnino. 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)*, pages 952–959. IEEE, 2014.

[37] Gianmarco De Francisci Morales and Albert Bifet. Samoa: scalable advanced massive online analysis. *Journal of Machine Learning Research*, 16(1):149–153, 2015.

[38] Albert Bifet and Gianmarco De Francisci Morales. Big data stream learning with samoa. In *2014 IEEE International Conference on Data Mining Workshop*, pages 1199–1202. IEEE, 2014.

[39] Matteo Riondato, Justin A DeBrabant, Rodrigo Fonseca, and Eli Upfal. Parma: a parallel randomized algorithm for approximate association rules mining in mapreduce. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 85–94. ACM, 2012.

[40] SAMOA-MOA. https://github.com/samoa-moa/samoa-moa.

[41] Alex Szalay. Extreme data-intensive scientific computing. *Computing in Science & Engineering*, 13(6):34–41, 2011.

[42] Laney Douglas. 3d data management: Controlling data volume, velocity and variety. *Gartner. Retrieved*, 6(2001):6, 2001.

[43] Xiaolong Jin, Benjamin W Wah, Xueqi Cheng, and Yuanzhuo Wang. Significance and challenges of big data research. *Big Data Research*, 2(2):59–64, 2015.

[44] Changqing Ji, Yu Li, Wenming Qiu, Uchechukwu Awada, and Keqiu Li. Big data processing in cloud computing environments. In *2012 12th international symposium on pervasive systems, algorithms and networks*, pages 17–23. IEEE, 2012.

[45] David Leong. A new revolution in enterprise storage architecture. *IEEE Potentials*, 28(6):32–33, 2009.

[46] Venkata Narasimha Inukollu, Sailaja Arsi, and Srinivasa Rao Ravuri. Security issues

associated with big data in cloud computing. *International Journal of Network Security & Its Applications*, 6(3):45, 2014.

[47] Kyong-Ha Lee, Yoon-Joon Lee, Hyunsik Choi, Yon Dohn Chung, and Bongki Moon. Parallel data processing with mapreduce: a survey. *AcM sIGMoD Record*, 40(4):11–20, 2012.

[48] Katsunari Shibata and Yusuke Ikeda. Effect of number of hidden neurons on learning in large-scale layered neural networks. In *2009 ICCAS-SICE*, pages 5008–5013. IEEE, 2009.

[49] Jian-xiong Dong, Adam Krzyzak, and Ching Y Suen. Fast svm training algorithm with decomposition on very large data sets. *IEEE transactions on pattern analysis and machine intelligence*, 27(4):603–618, 2005.

[50] Abdelkarim Ben Ayed, Mohamed Ben Halima, and Adel M Alimi. Survey on clustering methods: Towards fuzzy clustering for big data. In *2014 6th International conference of soft computing and pattern recognition (SoCPaR)*, pages 331–336. IEEE, 2014.

[51] Sanjay Ranka and Sartaj Sahni. Clustering on a hypercube multicomputer. *IEEE Transactions on Parallel and Distributed Systems*, 2(2):129–137, 1991.

[52] Marcos D Assunção, Rodrigo N Calheiros, Silvia Bianchi, Marco AS Netto, and Rajkumar Buyya. Big data computing and clouds: Trends and future directions. *Journal of Parallel and Distributed Computing*, 79:3–15, 2015.

[53] Domenico Talia. Clouds for scalable big data analytics. *Computer*, 46(5):98–101, 2013.

[54] Divyakant Agrawal, Sudipto Das, and Amr El Abbadi. Big data and cloud computing: current state and future opportunities. In *Proceedings of the 14th international conference on extending database technology*, pages 530–533, 2011.

[55] Ibrahim Abaker Targio Hashem, Ibrar Yaqoob, Nor Badrul Anuar, Salimah Mokhtar, Abdullah Gani, and Samee Ullah Khan. The rise of big data on cloud computing: Review and open research issues. *Information systems*, 47:98–115, 2015.

[56] Victor Chang and Gary Wills. A model to compare cloud and non-cloud storage of big data. *Future Generation Computer Systems*, 57:56–76, 2016.

[57] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: Simplified data processing on large clusters. 2004.

[58] Colby Ranger, Ramanan Raghuraman, Arun Penmetsa, Gary Bradski, and Christos Kozyrakis. Evaluating mapreduce for multi-core and multiprocessor systems. In *2007 IEEE 13th International Symposium on High Performance Computer Architecture*, pages 13–24. Ieee, 2007.

[59] Tomasz Nykiel, Michalis Potamias, Chaitanya Mishra, George Kollios, and Nick Koudas. Mrshare: sharing across multiple queries in mapreduce. *Proceedings of the VLDB Endowment*, 3(1-2):494–505, 2010.

[60] Jens Dittrich, Jorge-Arnulfo Quiané-Ruiz, Alekh Jindal, Yagiz Kargin, Vinay Setty, and Jörg Schad. Hadoop++ making a yellow elephant run like a cheetah (without it even noticing). *Proceedings of the VLDB Endowment*, 3(1-2):515–529, 2010.

[61] Chaokun Wang, Jianmin Wang, Xuemin Lin, Wei Wang, Haixun Wang, Hongsong Li, Wanpeng Tian, Jun Xu, and Rui Li. Mapdupreducer: detecting near duplicates over massive datasets. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 1119–1122, 2010.

[62] Robson Leonardo Ferreira Cordeiro, Caetano Traina, Agma Juci Machado Traina, Julio López, U Kang, and Christos Faloutsos. Clustering very large multi-dimensional datasets with mapreduce. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 690–698, 2011.

[63] Ping Zhou, Jingsheng Lei, and Wenjun Ye. Large-scale data sets clustering based on mapreduce and hadoop. *Journal of Computational Information Systems*, 7(16):5956–5963, 2011.

[64] Hai-Guang Li, Gong-Qing Wu, Xue-Gang Hu, Jing Zhang, Lian Li, and Xindong Wu. K-means clustering with bagging and mapreduce. In *2011 44th Hawaii International Conference on System Sciences*, pages 1–8. IEEE, 2011.

[65] Brendan Collins. Big data and health economics: strengths, weaknesses, opportunities and threats. *Pharmacoeconomics*, 34(2):101–106, 2016.

[66] Aisling ODriscoll, Jurate Daugelaite, and Roy D Sleator. Big data, hadoop and cloud computing in genomics. *Journal of biomedical informatics*, 46(5):774–781, 2013.

[67] Abdullah Gani, Aisha Siddiqa, Shahaboddin Shamshirband, and Fariza Hanum. A survey on indexing techniques for big data: taxonomy and performance evaluation. *Knowledge and information systems*, 46(2):241–284, 2016.

[68] Nader Mohamed and Jameela Al-Jaroodi. Real-time big data analytics: Applications and challenges. In *2014 international conference on high performance computing & simulation (HPCS)*, pages 305–310. IEEE, 2014.

[69] Amir Mosavi, Timon Rabczuk, and Annamária R Varkonyi-Koczy. Reviewing the novel machine learning tools for materials design. In *International Conference on Global Research and Education*, pages 50–58. Springer, 2017.

[70] Aaron E Maxwell, Timothy A Warner, and Fang Fang. Implementation of machine-learning classification in remote sensing: An applied review. *International Journal*

*of Remote Sensing*, 39(9):2784–2817, 2018.

[71] Servet Soyguder. Intelligent system based on wavelet decomposition and neural network for predicting of fan speed for energy saving in hvac system. *Energy and Buildings*, 43(4):814–822, 2011.

[72] Alberto Fernandez, Cristobal Jose Carmona, Maria Jose del Jesus, and Francisco Herrera. A view on fuzzy systems for big data: progress and opportunities. *International Journal of Computational Intelligence Systems*, 9(sup1):69–80, 2016.

[73] Simone A Ludwig. Mapreduce-based fuzzy c-means clustering algorithm: implementation and scalability. *International journal of machine learning and cybernetics*, 6(6):923–934, 2015.

[74] Dweepna Garg and Khushboo Trivedi. Fuzzy k-mean clustering in mapreduce on cloud based hadoop. In *2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies*, pages 1607–1610. IEEE, 2014.

[75] VP Ananthi, Pagavathigounder Balasubramaniam, and T Kalaiselvi. A new fuzzy clustering algorithm for the segmentation of brain tumor. *Soft Computing*, 20(12):4859–4879, 2016.

[76] Yu-Jie Wang. A clustering method based on fuzzy equivalence relation for customer relationship management. *Expert Systems with Applications*, 37(9):6421–6428, 2010.

[77] Victoria López, Sara Del Río, José Manuel Benítez, and Francisco Herrera. Cost-sensitive linguistic fuzzy rule based classification systems under the mapreduce framework for imbalanced big data. *Fuzzy Sets and Systems*, 258:5–38, 2015.

[78] Raghava Rao Mukkamala, Abid Hussain, and Ravi Vatrapu. Fuzzy-set based sentiment analysis of big social data. In *2014 IEEE 18th International Enterprise Distributed Object Computing Conference*, pages 71–80. IEEE, 2014.

[79] Thomas G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000.

[80] W Nick Street and YongSeog Kim. A streaming ensemble algorithm (sea) for large-scale classification. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 377–382, 2001.

[81] Xiao-Feng Gu, Jia-Wen Xu, Shi-Jing Huang, and Liao-Ming Wang. An improving online accuracy updated ensemble method in learning from evolving data streams. In *2014 11th International Computer Conference on Wavelet Actiev Media Technology and Information Processing (ICCWAMTIP)*, pages 430–433. IEEE, 2014.

[82] Bartosz Krawczyk and Michał Woźniak. Combining nearest neighbour classifiers based on small subsamples for big data analytics. In *2015 IEEE 2nd International Conference on Cybernetics (CYBCONF)*, pages 311–316. IEEE, 2015.

[83] Sen Jia and Nello Cristianini. Learning to classify gender from four million images. *Pattern recognition letters*, 58:35–41, 2015.

[84] Ahsanul Haque, Brandon Parker, Latifur Khan, and Bhavani Thuraisingham. Evolving big data stream classification with mapreduce. In *2014 IEEE 7th International Conference on Cloud Computing*, pages 570–577. IEEE, 2014.

[85] Yongjun Piao, Hyun Woo Park, Cheng Hao Jin, and Keun Ho Ryu. Ensemble method for classification of high-dimensional data. In *2014 International Conference on Big Data and Smart Computing (BIGCOMP)*, pages 245–249. IEEE, 2014.

[86] Alfredo Cuzzocrea, Enzo Mumolo, and Pietro Corona. Cloud-based machine learning tools for enhanced big data applications. In *2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, pages 908–914. IEEE, 2015.

[87] Xiao-Lin Wang, Yang-Yang Chen, Hai Zhao, and Bao-Liang Lu. Parallelized extreme learning machine ensemble based on min–max modular network. *Neurocomputing*, 128:31–41, 2014.

[88] Michael T Gorczyca, Nicole C Toscano, and Julius D Cheng. The trauma severity model: An ensemble machine learning approach to risk prediction. *Computers in biology and medicine*, 108:9–19, 2019.

[89] Qiu-Feng Wang, Mirror Xu, and Amir Hussain. Large-scale ensemble model for customer churn prediction in search ads. *Cognitive Computation*, 11(2):262–270, 2019.

[90] Seyed Amir Naghibi, Mojtaba Dolatkordestani, Ashkan Rezaei, Payam Amouzegari, Mostafa Taheri Heravi, Bahareh Kalantar, and Biswajeet Pradhan. Application of rotation forest with decision trees as base classifier and a novel ensemble model in spatial modeling of groundwater potential. *Environmental monitoring and assessment*, 191(4):1–20, 2019.

[91] Mumtaz Ali and Ramendra Prasad. Significant wave height forecasting via an extreme learning machine model integrated with improved complete ensemble empirical mode decomposition. *Renewable and Sustainable Energy Reviews*, 104:281–295, 2019.

[92] Akinori Yamanaka, Yuri Maeda, and Kengo Sasaki. Ensemble kalman filter-based data assimilation for three-dimensional multi-phase-field model: Estimation of anisotropic grain boundary properties. *Materials & Design*, 165:107577, 2019.

[93] Dhyan Chandra Yadav and Saurabh Pal. To generate an ensemble model for women thyroid prediction using data mining techniques. *Asian Pacific journal of cancer prevention: APJCP*, 20(4):1275, 2019.

[94] Qinzhong Hou, Junqiang Leng, Guosheng Ma, Weiyi Liu, and Yuxing Cheng. An adaptive hybrid model for short-term urban traffic flow prediction. *Physica A:*

*Statistical Mechanics and its Applications*, 527:121065, 2019.

[95] Pei Du, Jianzhou Wang, Wendong Yang, and Tong Niu. A novel hybrid model for short-term wind power forecasting. *Applied Soft Computing*, 80:93–106, 2019.

[96] Wenyu Zhang, Hongliang He, and Shuai Zhang. A novel multi-stage hybrid model with enhanced multi-population niche genetic algorithm: An application in credit scoring. *Expert Systems with Applications*, 121:221–232, 2019.

[97] Binh Thai Pham and Indra Prakash. A novel hybrid model of bagging-based naïve bayes trees for landslide susceptibility assessment. *Bulletin of Engineering Geology and the Environment*, 78(3):1911–1925, 2019.

[98] Jinran Wu, Zhesen Cui, Yanyan Chen, Demeng Kong, and You-Gan Wang. A new hybrid model to predict the electrical load in five states of australia. *Energy*, 166:598–609, 2019.

[99] Fahad Albalawi, Abderrazak Chahid, Xingang Guo, Somayah Albaradei, Arturo Magana-Mora, Boris R Jankovic, Mahmut Uludag, Christophe Van Neste, Magbubah Essack, Taous-Meriem Laleg-Kirati, et al. Hybrid model for efficient prediction of poly (a) signals in human genomic dna. *Methods*, 166:31–39, 2019.

[100] Bernard J Poiesz, Francis W Ruscetti, Adi F Gazdar, Paul A Bunn, John D Minna, and Robert C Gallo. Detection and isolation of type c retrovirus particles from fresh and cultured lymphocytes of a patient with cutaneous t-cell lymphoma. *Proceedings of the National Academy of Sciences*, 77(12):7415–7419, 1980.

[101] Yorio Hinuma, Kinya Nagata, Masao Hanaoka, Masuyo Nakai, Tadashi Matsumoto, Ken-Ichiro Kinoshita, Shigeru Shirakawa, and Isao Miyoshi. Adult t-cell leukemia: antigen in an atl cell line and detection of antibodies to the antigen in human sera. *Proceedings of the National Academy of Sciences*, 78(10):6476–6480, 1981.

[102] Angela Manns, Rainford J Wilks, Edward L Murphy, Grace Haynes, J Peter Figueroa, Marge Barnett, Barrie Hanchard, and William A Blattner. A prospective study of transmission by transfusion of htlv-i and risk factors associated with seroconversion. *International journal of cancer*, 51(6):886–891, 1992.

[103] Renaud Mahieux and Antoine Gessain. The human htlv-3 and htlv-4 retroviruses: new members of the htlv family. *Pathologie Biologie*, 57(2):161–166, 2009.

[104] Luc Willems, Hideki Hasegawa, Roberto Accolla, Charles Bangham, Ali Bazarbachi, Umberto Bertazzoni, Anna Barbara de Freitas Carneiro-Proietti, Hua Cheng, Luigi Chieco-Bianchi, Vincenzo Ciminale, et al. Reducing the global burden of htlv-1 infection: An agenda for research and action. *Antiviral research*, 137:41–48, 2017.

[105] Fernando A Proietti, Anna Bárbara F Carneiro-Proietti, Bernadette C Catalan-Soares, and Edward L Murphy. Global epidemiology of htlv-i infection and associ-

ated diseases. *Oncogene*, 24(39):6058, 2005.

[106] Antoine Gessain, Alain Jouannelle, Patrick Escarmant, Alain Calender, Laurence Schaffar-Deshayes, et al. Htlv antibodies in patients with non-hodgkin lymphomas in martinique. *The Lancet*, 323(8387):1183–1184, 1984.

[107] Mitsuhiro Osame, Koichiro Usuku, Shuji Izumo, Naomi Ijichi, Hiroyoko Amitani, Akihiro Igata, Makoto Matsumoto, and Mitsutoshi Tara. Htlv-i associated myelopathy, a new clinical entity. *The Lancet*, 327(8488):1031–1032, 1986.

[108] Masao Matsuoka. Human t-cell leukemia virus type i (htlv-i) infection and the onset of adult t-cell leukemia (atl). *Retrovirology*, 2(1):27, 2005.

[109] Sonia Van Dooren, Marco Salemi, and A-M Vandamme. Dating the origin of the african human t-cell lymphotropic virus type-i (htlv-i) subtypes. *Molecular Biology and Evolution*, 18(4):661–671, 2001.

[110] Denise Utsch Gonçalves, Fernando Augusto Proietti, João Gabriel Ramos Ribas, Marcelo Grossi Araújo, Sônia Regina Pinheiro, Antônio Carlos Guedes, and Anna Bárbara F Carneiro-Proietti. Epidemiology, treatment, and prevention of human t-cell leukemia virus type 1-associated diseases. *Clinical microbiology reviews*, 23(3):577–589, 2010.

[111] Mitsuhiro Osame, Robert Janssen, Hiroaki Kubota, Hiroshi Nishitani, Akihiro Igata, Shigenobu Nagataki, Masataka Mori, Ikuo Goto, Hiromi Shimabukuro, Rima Khabbaz, et al. Nationwide survey of htlv-i–associated myelopathy in japan: Association with blood transfusion. *Annals of neurology*, 28(1):50–56, 1990.

[112] Sijia Wu, Jiuqiang Han, Ruiling Liu, Jun Liu, and Hongqiang Lv. A computational model for predicting fusion peptide of retroviruses. *Computational biology and chemistry*, 61:245–250, 2016.

[113] Suyu Mei and Hao Zhu. A novel one-class svm based negative data sampling method for reconstructing proteome-wide htlv-human protein interaction networks. *Scientific reports*, 5:8034, 2015.

[114] De-Shuang Huang, Lei Zhang, Kyungsook Han, Suping Deng, Kai Yang, and Hongbo Zhang. Prediction of protein-protein interactions based on protein-protein correlation using least squares regression. *Current Protein and Peptide Science*, 15(6):553–560, 2014.

[115] Likun Wang, Yipeng Du, Ming Lu, and Tingting Li. Aseb: a web server for kat-specific acetylation site prediction. *Nucleic acids research*, 40(W1):W376–W379, 2012.

[116] Chun-Hou Zheng, Lei Zhang, Vincent To-Yee Ng, Chi Keung Shiu, and D-S Huang. Molecular pattern discovery based on penalized matrix decomposition. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 8(6):1592–

1603, 2011.

[117] Aline Cristina A Mota-Miranda, Fernanda K Barreto, Everton Baptista, Lourdes Farre-Vale, Joana P Monteiro-Cunha, Bernardo Galvao-Castro, and Luiz Carlos J Alcantara. Molecular study of hbz and gp21 human t cell leukemia virus type 1 proteins isolated from different clinical profile infected individuals. *AIDS research and human retroviruses*, 29(10):1370–1372, 2013.

[118] Hong-Jie Yu and De-Shuang Huang. Normalized feature vectors: a novel alignment-free sequence comparison method based on the numbers of adjacent amino acids. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 10(2):457–467, 2013.

[119] Divya Khanna and Prashant Singh Rana. Multilevel ensemble model for prediction of iga and igg antibodies. *Immunology letters*, 184:51–60, 2017.

[120] Nishtha Hooda, Seema Bawa, and Prashant Singh Rana. B2fse framework for high dimensional imbalanced data: A case study for drug toxicity prediction. *Neurocomputing*, 2017.

[121] Uniprot. http://www.uniprot.org/proteomes/?query= human+t+cell+leukemia+virus &sort=score. Accessed on 25, January 2018.

[122] UniProt. http://www.uniprot.org. Accessed on 25, January 2018.

[123] Atsushi Ikai. Thermostability and aliphatic index of globular proteins. *The Journal of Biochemistry*, 88(6):1895–1898, 1980.

[124] HG Boman. Antibacterial peptides: basic facts and emerging concepts. *Journal of internal medicine*, 254(3):197–215, 2003.

[125] Kunchur Guruprasad, BV Bhasker Reddy, and Madhusudan W Pandit. Correlation between stability of a protein and its dipeptide composition: a novel approach for predicting in vivo stability of a protein from its primary sequence. *Protein Engineering, Design and Selection*, 4(2):155–161, 1990.

[126] David Eisenberg, Robert M Weiss, and Thomas C Terwilliger. The hydrophobic moment detects periodicity in protein hydrophobicity. *Proceedings of the National Academy of Sciences*, 81(1):140–144, 1984.

[127] Elisabeth Gasteiger, Christine Hoogland, Alexandre Gattiker, Marc R Wilkins, Ron D Appel, Amos Bairoch, et al. Protein identification and analysis tools on the expasy server. In *The proteomics protocols handbook*, pages 571–607. Springer, 2005.

[128] Bengt Bjellqvist, Graham J Hughes, Christian Pasquali, Nicole Paquet, Florence Ravier, Jean-Charles Sanchez, Severine Frutiger, and Denis Hochstrasser. The focusing positions of polypeptides in immobilized ph gradients can be predicted from their amino acid sequences. *Electrophoresis*, 14(1):1023–1031, 1993.

[129] JM Zimmerman, Naomi Eliezer, and R Simha. The characterization of amino acid sequences in proteins by statistical methods. *Journal of theoretical biology*, 21(2):170–201, 1968.

[130] Akintola A Aboderin. An empirical hydrophobicity scale for $\alpha$-amino-acids and some of its applications. *International Journal of Biochemistry*, 2(11):537–544, 1971.

[131] Akinori Kidera, Yasuo Konishi, Masahito Oka, Tatsuo Ooi, and Harold A Scheraga. Statistical analysis of the physical properties of the 20 naturally occurring amino acids. *Journal of Protein Chemistry*, 4(1):23–55, 1985.

[132] Peter Rice, Ian Longden, and Alan Bleasby. Emboss: the european molecular biology open software suite, 2000.

[133] Gerard JP Van Westen, Remco F Swier, Jörg K Wegner, Adriaan P IJzerman, Herman WT van Vlijmen, and Andreas Bender. Benchmarking of protein descriptor sets in proteochemometric modeling (part 1): comparative study of 13 amino acid descriptor sets. *Journal of cheminformatics*, 5(1):41, 2013.

[134] Joseph C Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. pages 32–57, 1973.

[135] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[136] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

[137] Wong Hartigan. Algorithm as 136:a k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C, (Applied Statistics)*, 28:100–108, 1979.

[138] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[139] Jason Weston and Chris Watkins. Multi-class support vector machines. Technical report, Citeseer, 1998.

[140] S. Sathiya Keerthi and Elmer G Gilbert. Convergence of a generalized smo algorithm for svm classifier design. *Machine Learning*, 46(1-3):351–360, 2002.

[141] Brian D Ripley. *Pattern recognition and neural networks*. Cambridge university press, 2007.

[142] Martin Riedmiller and Heinrich Braun. A direct adaptive method for faster backpropagation learning: The rprop algorithm. In *Proceedings of the IEEE international conference on neural networks*, volume 1993, pages 586–591. San Francisco, 1993.

[143] Norvig Russell. *Artificial Intelligence: A Modern Approach (2nd ed.)*. Prentice Hall

press, 2003.

[144] Andy Liaw, Matthew Wiener, et al. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.

[145] David Martin Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. 2011.

[146] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995.

[147] Ashwani Kumar and Subrata Kumar Ghosh. Size distribution analysis of wear debris generated in hemm engine oil for reliability assessment: A statistical approach. *Measurement*, 131:412–418, 2019.

[148] Mohamed Kamal Ahmed Ali, Hou Xianjun, Liqiang Mai, Cai Qingping, Richard Fifi Turkson, and Chen Bicheng. Improving the tribological characteristics of piston ring assembly in automotive engines using al2o3 and tio2 nanomaterials as nanolubricant additives. *Tribology International*, 103:540–554, 2016.

[149] Mohamed Kamal Ahmed Ali, Peng Fuming, Hussein A Younus, Mohamed AA Abdelkareem, FA Essa, Ahmed Elagouz, and Hou Xianjun. Fuel economy in gasoline engines using al2o3/tio2 nanomaterials as nanolubricant additives. *Applied energy*, 211:461–478, 2018.

[150] Mohamed Kamal Ahmed Ali, Hou Xianjun, Mohamed AA Abdelkareem, M Gulzar, and AH Elsheikh. Novel approach of the graphene nanolubricant for energy saving via anti-friction/wear in automobile engines. *Tribology International*, 124:209–229, 2018.

[151] Mohamed Kamal Ahmed Ali and Hou Xianjun. Improving the tribological behavior of internal combustion engines via the addition of nanoparticles to engine oils. *Nanotechnology Reviews*, 4(4):347–358, 2015.

[152] Ankit Kotia, Pranami Rajkhowa, Gogineni Satyanarayana Rao, and Subrata Kumar Ghosh. Thermophysical and tribological properties of nanolubricants: A review. *Heat and Mass Transfer*, 54(11):3493–3508, 2018.

[153] Ankit Kotia, Gaurab Kumar Ghosh, Isha Srivastava, Piyush Deval, and Subrata Kumar Ghosh. Mechanism for improvement of friction/wear by using al2o3 and sio2/gear oil nanolubricants. *Journal of Alloys and Compounds*, 782:592–599, 2019.

[154] Kianoosh Shababi, Masoumeh Firouzi, and Ahmad Fakhar. An experimental study on rheological behavior of sae50 engine oil. *Journal of Thermal Analysis and Calorimetry*, 131(3):2311–2320, 2018.

[155] Mark A Kedzierski. Viscosity and density of aluminum oxide nanolubricant. *international journal of refrigeration*, 36(4):1333–1340, 2013.

[156] Omer A Alawi, Nor Azwadi Che Sidik, Hong Wei Xian, Tung Hao Kean, and SN Kazi. Thermal conductivity and viscosity models of metallic oxides nanofluids. *International Journal of Heat and Mass Transfer*, 116:1314–1325, 2018.

[157] Mohammad Hemmat Esfe, Mohammad Hassan Kamyab, Masoud Afrand, and Mahmoud Kiannejad Amiri. Using artificial neural network for investigating of concurrent effects of multi-walled carbon nanotubes and alumina nanoparticles on the viscosity of 10w-40 engine oil. *Physica A: Statistical Mechanics and its Applications*, 510:610–624, 2018.

[158] Amin Shahsavar, Shoaib Khanmohammadi, Arash Karimipour, and Marjan Goodarzi. A novel comprehensive experimental study concerned synthesizes and prepare liquid paraffin-fe3o4 mixture to develop models for both thermal conductivity & viscosity: A new approach of gmdh type of neural network. *International Journal of Heat and Mass Transfer*, 131:432–441, 2019.

[159] Mohammad Hemmat Esfe, Afshin Tatar, Mohammad Reza Hassani Ahangar, and Hossein Rostamian. A comparison of performance of several artificial intelligence methods for predicting the dynamic viscosity of tio2/sae 50 nano-lubricant. *Physica E: Low-dimensional Systems and Nanostructures*, 96:85–93, 2018.

[160] M Vakili, S Khosrojerdi, P Aghajannezhad, and M Yahyaei. A hybrid artificial neural network-genetic algorithm modeling approach for viscosity estimation of graphene nanoplatelets nanofluid using experimental data. *International Communications in Heat and Mass Transfer*, 82:40–48, 2017.

[161] HR Ansari, MJ Zarei, S Sabbaghi, and P Keshavarz. A new comprehensive model for relative viscosity of various nanofluids using feed-forward back-propagation mlp neural networks. *International Communications in Heat and Mass Transfer*, 91:158–164, 2018.

[162] Mohammad Hemmat Esfe and Ali Akbar Abbasian Arani. An experimental determination and accurate prediction of dynamic viscosity of mwcnt (% 40)-sio2 (% 60)/5w50 nano-lubricant. *Journal of Molecular Liquids*, 259:227–237, 2018.

[163] Jatinder Gill, Jagdev Singh, Olayinka S Ohunakin, and Damola S Adelekan. Artificial neural network approach for irreversibility performance analysis of domestic refrigerator by utilizing lpg with tio2–lubricant as replacement of r134a. *International Journal of Refrigeration*, 89:159–176, 2018.

[164] L Pena-Paras, D Maldonado-Cortes, J Taha-Tijerina, M Irigoyen, and J Guerra. Experimental evaluation of the tribological behaviour of ceo2 nanolubricants under extreme pressures. In *IOP Conference Series: Materials Science and Engineering*, volume 400, page 072003. IOP Publishing, 2018.

[165] Ankit Kotia and Subrata Kumar Ghosh. Experimental analysis for rheological

properties of aluminium oxide (al2o3)/gear oil (sae ep-90) nanolubricant used in hemm. *Industrial Lubrication and Tribology*, 2015.

[166] MZ Sharif, WH Azmi, AAM Redhwan, and NMM Zawawi. Preparation and stability of silicone dioxide dispersed in polyalkylene glycol based nanolubricants. In *MATEC web of conferences*, volume 90, page 01049. EDP Sciences, 2017.

[167] Ankit Kotia, Krishna Chowdary, Isha Srivastava, Subrata Kumar Ghosh, and Mohamed Kamal Ahmed Ali. Carbon nanomaterials as friction modifiers in automotive engines: Recent progress and perspectives. *Journal of Molecular Liquids*, page 113200, 2020.

[168] Mohamed Kamal Ahmed Ali and Hou Xianjun. Improving the heat transfer capability and thermal stability of vehicle engine oils using al2o3/tio2 nanomaterials. *Powder Technology*, 363:48–58, 2020.

[169] John M Chambers. Computational methods for data analysis. Technical report, 1977.

[170] Ching-Lai Hwang and Kwangsun Yoon. *Multiple attribute decision making: methods and applications a state-of-the-art survey*, volume 186. Springer Science & Business Media, 2012.

[171] K Paul Yoon and Ching-Lai Hwang. *Multiple attribute decision making: an introduction*, volume 104. Sage publications, 1995.

[172] Mohamed Kamal Ahmed Ali, Hou Xianjun, Richard Fiifi Turkson, Zhan Peng, and Xiandong Chen. Enhancing the thermophysical properties and tribological behaviour of engine oils using nano-lubricant additives. *RSC Advances*, 6(81):77913–77924, 2016.

[173] Pierre Baldi, Peter Sadowski, and Daniel Whiteson. Searching for exotic particles in high-energy physics with deep learning. *Nature communications*, 5:4308, 2014.

[174] S Dawson, A Gritsan, H Logan, J Qian, C Tully, R Van Kooten, A Ajaib, A Anastassov, I Anderson, D Asner, et al. Higgs working group report of the snowmass 2013 community planning study. *arXiv preprint arXiv:1310.8361*, 2013.

[175] Georges Aad, T Abajyan, B Abbott, J Abdallah, S Abdel Khalek, O Abdinov, R Aben, B Abi, M Abolins, OS AbouZeid, et al. Search for a multi-higgs-boson cascade in w+ w- b b$^-$ events with the atlas detector in p p collisions at s= 8 tev. *Physical Review D*, 89(3):032002, 2014.

[176] Georges Aad, T Abajyan, B Abbott, J Abdallah, S Abdel Khalek, AA Abdelalim, O Abdinov, R Aben, B Abi, M Abolins, et al. Observation of a new particle in the search for the standard model higgs boson with the atlas detector at the lhc. *Physics Letters B*, 716(1):1–29, 2012.

[177] T Aaltonen, J Adelman, B Álvarez González, S Amerio, D Amidei, A Anastassov,

A Annovi, J Antos, G Apollinari, JA Appel, et al. Search for a two-higgs-boson doublet using a simplified model in p p$^-$ collisions at s= 1.96 tev. *Physical review letters*, 110(12):121801, 2013.

[178] Serguei Chatrchyan, Vardan Khachatryan, Albert M Sirunyan, Armen Tumasyan, Wolfgang Adam, Ernest Aguilo, T Bergauer, M Dragicevic, J Erö, C Fabjan, et al. Observation of a new boson at a mass of 125 gev with the cms experiment at the lhc. *Physics Letters B*, 716(1):30–61, 2012.

[179] Sepp Hochreiter. Recurrent neural net learning and vanishing gradient. *International Journal Of Uncertainity, Fuzziness and Knowledge-Based Systems*, 6(2):107–116, 1998.

[180] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[181] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[182] Pierre Baldi and Peter Sadowski. The dropout learning algorithm. *Artificial intelligence*, 210:78–122, 2014.

[183] https://archive.ics.uci.edu/ml/datasets/higgs. Accessed on 15, June 2018.

[184] https://archive.ics.uci.edu/ml/datasets/susy. Accessed on 15, June 2018.

[185] http://spark.apache.org/. Accessed on 30, July 2018.

[186] Matei Zaharia, Holden Karau, Andy Konwinski, and Patrick Wendell. *Learning Spark Lightning-Fast Big Data Analysis*. O'Reilly Media, 2015.

[187] https://dbc-aa32d8b5-8db5.cloud.databricks.com/login.html. Accessed on 28, July 2018.

[188] https://s3.console.aws.amazon.com/s3/home?region=us-east-1#. Accessed on 28, July 2018.

[189] Galen Andrew and Jianfeng Gao. Scalable training of l 1-regularized log-linear models. In *Proceedings of the 24th international conference on Machine learning*, pages 33–40. ACM, 2007.

[190] Leo Breiman. Arcing the edge. Technical report, Technical Report 486, Statistics Department, University of California at Berkeley, 1997.

[191] Jerome H Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378, 2002.

[192] Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[193] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc

curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240. ACM, 2006.