AN EFFICIENT PREDICTIVE BIG DATA ANALYTICS SYSTEM FOR HIGH DIMENSIONAL DATA

MYAT CHO MON OO

UNIVERSITY OF COMPUTER STUDIES, YANGON

FEBRUARY, 2021

An Efficient Predictive Big Data Analytics System For High Dimensional Data

Myat Cho Mon Oo

University of Computer Studies, Yangon

A thesis submitted to the University of Computer Studies, Yangon in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

February, 2021

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

2.3.2021 Cho Mon

…..…… ……..……

Date Myat Cho Mon Oo

ACKNOWLEDGEMENTS

First of all, I would like to thank the Union Minister, the Ministry of Education for giving the opportunity to attend this course leading to the doctoral degree courses at the University of Computer Studies, Yangon.

Secondly, I would like to express very special thanks to Dr. Mie Mie Thet Thwin, Rector, the University of Computer Studies, Yangon, for allowing me to develop this thesis and giving me general guidance during the period of my study.

I would like to express my deep gratitude to my supervisor, Dr. Thandar Thein, Rector of the University of Computer Studies (Maubin) for providing me with incredibly valuable guidance and advice throughout my Ph.D. candidature. In addition, I would like to express my sincere gratitude to my supervisor for giving much of her time. I was motivated by her excellent guidance and creative ideas to improve my research skills. I admire her courage, her positive outlook, her ability to offer advice. I am very lucky to be working under her supervision.

I would also like to express my respectful gratitude to Dr. Khine Khine Oo, Professor, and Dean of the Ph.D. 10th batch course, the University of Computer Studies, Yangon, for her excellent guidance, caring, and providing me during the Ph.D. study.

I would like to express my respectful gratitude to all my teachers for their encouragement and recommending the thesis. To the reading committee teachers, especially Daw Aye Aye Khine, Associate Professor, Head of English Department, I would like to thank her for valuable supports and editing my thesis from the language point of view.

I also thank my friends from the Ph.D.10th batch for their co-operation and encouragement.

Last but not least, I am very much indebted to my mother and sister for always believing in me, for their endless love and support. They are always supportive of me during my period of studies, especially for this Doctorate Course.

ABSTRACT

In the current big explosion era, data is increasing dramatically every year. Gaining critical business insights by querying and analyzing this vast amount of data is becoming a challenge for conventional data mining techniques. It is not fit for processing big data beyond the capabilities of traditional systems. Massive samples and features of big data create issues such as heavy computational cost and algorithmic instability because it brings the curse of dimensionality. Predictive analytics is the enabler of big data, using machine learning algorithms to extract useful knowledge from large amounts of data and makes more formidable efforts. Effective and reliable results of the predictive analytics system depend on the quality of the predictive model.

This research aims to develop the efficient Predictive Big data Analytics, PBA system, for providing the valuable information and making a better business decision in an efficient and timely manner. To achieve this goal, PBA system with different architectures on big data analytics platforms is implemented by examining the bulk of big data. Firstly, scalability test is carried out by analyzing the performance of machine learning on traditional and big data analytics platform for reducing the generalization error and processing the massive data. The processing performance of analytics engines (MapReduce and Apache Spark) is conducted using a scalable machine learning algorithm and then Spark processing engine is selected to provide computationally efficient and relatively easy to implement the PBA system. For developing a scalable and high-performance PBA system, model selection is performed by evaluation the performance of four different machine learning algorithms (Random Forest, Gradient Boosting, Decision Trees and Linear Regression). The efficient PBA system is established based on the powerful machine learning technique, Scalable Random Forests (SRF). To get the prediction model with high accuracy, Hyperparameters Optimization in SRF is performed. In addition to mitigating data quality challenges, reducing the high dimensions of data improves operational efficiency by minimizing computational and storage costs. Real-time PBA system is developed to achieve high predictive powers in real-time manner. The different U.S stock data from eight companies are captured in real-time and predicts whether the stock prices will rise or fall relative to the price n days ago. In RPBA

system, the features of stock datasets are considered as input feature variables based on the calculation of technical indicators for helping the investors to buy or sell the stocks. Experimental results indicated that the prediction accuracy of the proposed PBA system is better than the RF algorithm from Spark's scalable machine learning library. The important finding of this research is that the combination of SRF's hyperparameters optimization and dimensionality reduction technology can considerably improve the efficiency and effectiveness of the system in terms of accuracy and computational time.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

CHAPTER 1

INTRODUCTION

The massive amount of data generated and shared by multiple disparate sources, including social media, mobile devices, bank records, transactions, and the Internet of Things, has been accelerated. The IBM Big Data Flood Info Graphic reported that 2.5 quintillion bytes of data were created daily around the world, and 90 percent of these generated data was unstructured. Unlike traditional data, Big Data has various structures (relational data, SQL, etc.), unstructured (video, audio, documents, social media comments, clickstream data, etc.), and semi-structured data (XML, JSON, sensors, etc.). Typically, it can be characterized by seven dimensions: Volume, Velocity, Variety, Variability, Veracity, Visualization, and Value of data. Conventional data mining techniques are becoming inadequate to process the large amounts of data, including disparate formats. Data analytics tools and approaches provide organizations with a way to evaluate data sets and obtain new information on a large scale. Organizations can make data-driven decisions using big data analytics systems and predictive analytic techniques, resulting in more effective marketing, new revenue opportunities, fraud detection, marketing, risk assessment, and increased operational efficiency.

This chapter outlines the core concepts of data analytics and machine learning. This highlights the new Big Data trends and the emerging technologies. Then, problems and issues in the research area are described. It follows by motivations, contribution and objectives of the proposed system.

1.1 Data Analytics in the Era of Big Data

With the advent of new technologies like internet of things, smart transport, and wide variety of social media sites, online transaction records produce huge volumes of data in unified formats. The growth of enormous data beyond the limits of human perception has led to the development of technologies for deriving insight. More sophisticated technology, such as the data analytics tools that helps users not only to understand the data, but to evaluate the potential of various actions and decisions. The main important factor is on transforming data into actionable insights.

Big data analytics that discover insights from evidence has a high demand for computing efficiency, knowledge discovery, problem solving, and event prescription. It also poses great challenges in terms of data, process, analytical modeling and management for organizations to turn big data into big insight. Analytics has a spectrum of methodologies, techniques, and approaches from descriptive, diagnostic, predictive and prescriptive analytics.

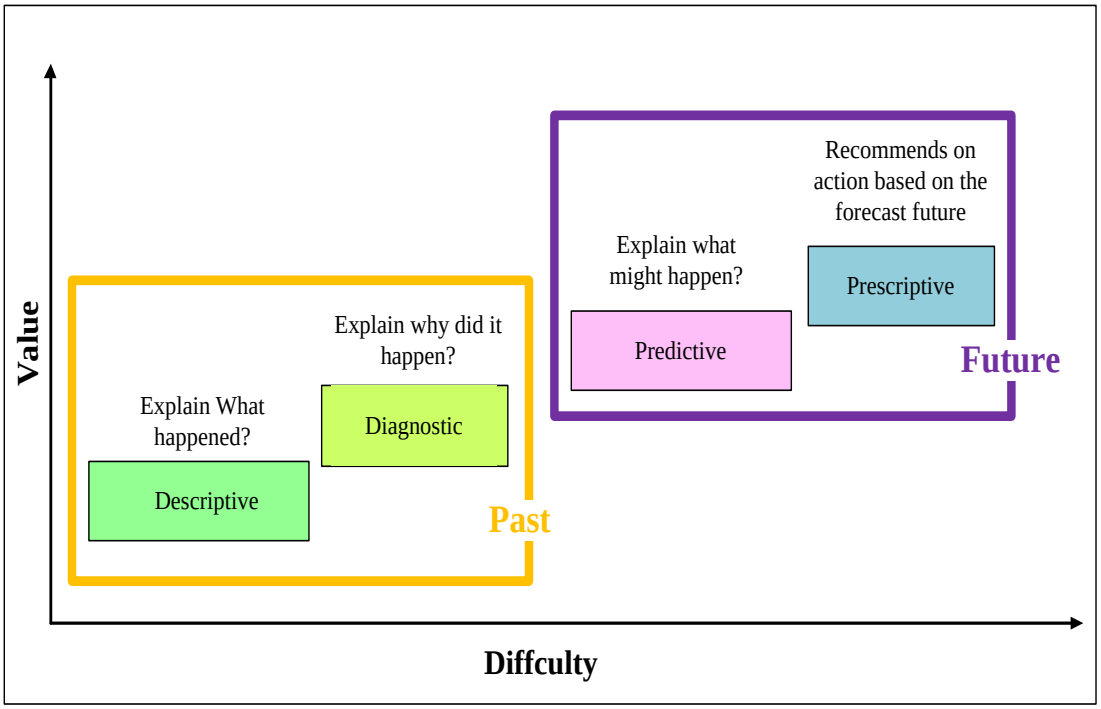


Figure 1.1 Four types of Data Analytics

1.1.1 Descriptive Analytics

The most frequent and fundamental type of analytics used by businesses is descriptive analytics. Descriptive analytics may be used by any area of the organization to maintain track of operational performance and track trends. Descriptive analytics focuses on summarizing and identifying trends in current and historical data, allowing businesses to better comprehend what has occurred thus far. Organizations can understand why something happened and anticipate probable future events and actions by combining descriptive analytics with diagnostic, predictive, and prescriptive analytics.

1.1.2 Diagnostic Analytics

Diagnostic analytics, also known as root cause analysis, supports analysts in determining why a strategic issue or occurrence in the data occurred. Diagnostic analytics is a type of advanced analytics that digs deeper into data to try to figure out what's causing events and behaviors. Diagnostic analytics are typically harder to execute and offer more meaningful insight than just reporting the existence of a problem since they involve more information and more diverse approaches to get to the heart of a given problem.

1.1.3 Predictive Analytics

Predictive analytics is a form of advanced analytics which analyzes historical and current data to forecast future events or actionable outcomes. In contrast to traditional analytics, predictive analytics identifies Big Data patterns that represent useful information and predicts future consequences. Predictive analytics can be used to detect fraud, minimize risk in financial operations, and increase computing efficiency. These capabilities can provide competitive advantages over competitors with the right strategy.

There is a promising potential in predictive analytics to harness the power Big Data behavior by exploiting the opportunities of the huge quantities of information generated on a variety of heterogeneous data sources. Predictive analytics is slightly different from other forms of big data analytics in that it is the only form that provides future predictions. Other actions, such as these prescriptive analyses, give directions on what should be taken to solve various business challenges. It is an advance analytics technique of Big Data activities to deliver unprecedented flexibility, precision and efficiency to prediction process.

1.1.4 Prescriptive Analytics

Having considered the available data, prescriptive analytics focuses on determining the best course of action in a situation. It is relevant to both descriptive analytics and predictive analytics, but focuses actionable observations rather than data analysis.

1.2 Predictive Machine Learning Methods: Supervised and Unsupervised

Machine learning algorithms are adopted for specific types of problems and data; therefore it is important to assure that there is a compatible between the natures of data, issues and algorithm to achieve optimal efficiency. Machine learning algorithm largely fall into two groups, supervised and unsupervised learning. Supervised Learning is a machine learning task of learning a function for gaining knowledge about input-output relationship of a system based on a series of paired input-output samples. Since output is regarded as the input data label or supervision, an input-output training sample is also referred to as labeled training data, or supervised information.

In supervised learning, the learning system is self-organized learning to find unrevealed patterns in datasets without pre-defined target variables whereas predictive analysis is the analysis of historical data as well as existing external data to find patterns and behaviors. Unsupervised learning is a machine learning paradigm that identifies trends in unlabeled data, or data without a given measure of response. However, analysts may use unsupervised learning techniques in the course of a predictive analytics project to understand the data and accelerate the model building process. Unsupervised learning techniques frequently used within the predictive modeling process include anomaly detection, graph and network analysis, Bayesian Networks, text mining, clustering, and dimension reduction.

1.2.1 Scalable Machine Learning for Predictive Analytics

As the Big Data grows, the challenge of Scalable Machine Learning (SML) has been intensified in a constant bias to deal with more complex problems. Nowadays, SML are used for PBA by analyzing for unraveling the hidden patterns, unascertained correlations, market trends and a lot other beneficial information. SML techniques make the knowledge acquisition and process the training dataset in batch or real-time PA. Batch data learning is that a a set of collected training data in a certain time, pre-processed data, build the models and produce the output results. In contrast, real-time big data processing take the continuous or stream data and process these massive data within a small period of time (near real-time). It can able to take immediate action to gain the insights needed continuously at the right time.

1.3 Motivation of the Thesis

The scale of big data which is generated from various sources is increasing overwhelmingly over the past decade. Big Data is the hottest topic in today's information technology, and it has to be examined in order to improve decision-making. Various researches reported that the continuous amount of data increase with the high speed [1], [2] and traditional data mining techniques are difficult to be processed such big data.

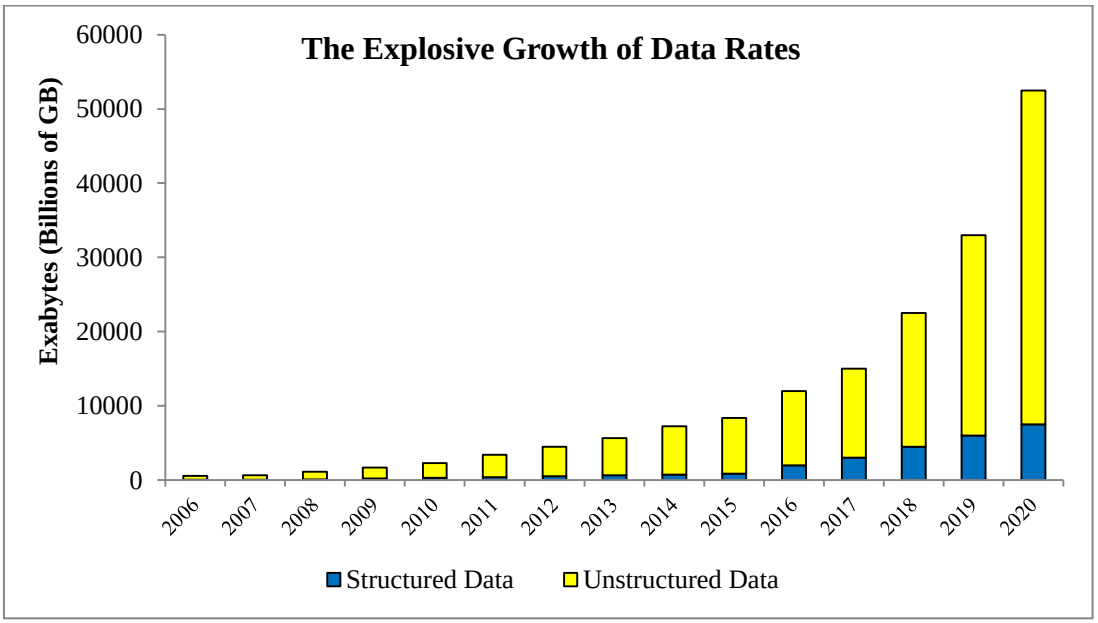


Figure 1.2 Explosive Growth of Data Rate in Worldwide

According to 2014 report from [70] on the explosive data rates of structure and unstructured data in worldwide as shown in Figure 1.2, the different types of data become more diverse and the amount of data increase dramatically. As analytics become a critical trend and more and more modern societies harness its power, the majority of Predictive Big data Analytics (PBA) also need to meet requirements for performance, scalability, timeliness, security and data governance [4]. Based on current and historical data, PBA uses statistical and machine learning algorithms to predict outcomes [5]. Efficient analytical techniques are required to understand risks by catching the suspicious trends optimize process and forecast the revenue generating patterns.

1.4 Problem Statement

The era of big data brings serious challenges and issues beyond profitable happiness. The main problem facing PBA is getting an efficient way to handle the big data. The PBA system can identify future risks and opportunities using patterns found in past and current transaction data. It captures the relationships among many factors using SML techniques to assess risk with a certain condition. PBA uses SML techniques to assess risk under specific conditions and capture relationships between many factors. Traditional data mining platforms must be scalable in order to manage broad scale of big data. The first issue in developing a scalable and efficient PBA system in a timely and effective manner is a major issue of the thesis.

The second issue is choosing a compatible processing engine to develop an efficient PBA system. This is required for large-scale machine learning in distributed systems. To ensure the computational efficiency of PBA system, it is necessary to analyze the performance of processing engines using SML techniques. The performance of same SML method on different processing engines is needed to analyze for efficiently processing the big data.

The third one is reducing the high dimensionality of Big Data. Using all the features of the entire dataset introduces model training overhead and affects the curse of the dimensionality. Therefore, high dimensionality reduction has become one of the critical issues of the PBA system.

The fourth one is choosing the suitable SML technique which considers the data nature of the system. The prediction accuracy of the system is needed to be enhanced to reduce error rates in an effective manner.

The last one is implementing high level performance of real-time PBA system. With the tremendous growth of Big Data, PBA requires real-time tracing and analytics because empower decisions can be provided with faster analytics which bring a better profit.

1.5 Objectives of the Thesis

The emergence of big data surges has been powered primarily by an unending tidal wave of web clicks, queries and routes. The main objective of this research is to

develop the efficient PBA system, which can achieve scalability and the ability to overcome the issue of the curse of dimensionality in a timely manner. The goals of this research are the following:

- To support the effective processing task on the proposed system
- To perform analysis of scalability test and processing engine selection
- To alleviate the training time of the proposed system in advance by considering the feature importance
- To minimize the generalization error by avoiding overfitting in the learning algorithm
- To address the high dimensionality problem by enhancing the performance of the SML algorithm
- To develop Real-time PBA system for faster decision making by exploiting the real-time data

This research develops an efficient and high-performance PBA system which meets the requirement of the high predictability and low latency for decision making process. In addition, it can offer real-time analytics with high prediction results.

1.6 Contributions of the Thesis

As more and more systems produce the vast amount of data, which is generated in strikingly large extent recently, an efficient and accurate predictive analytics system is essential for successful data management. With the increasing popularity and capabilities of SML techniques, predictive analytics is particularly suitable for the execution of distributed algorithms for big data analysis. There is a necessary of efficient processing of the PBA system with the high prediction accuracy. The contributions of this thesis are as follows.

An efficient PBA system on distributed environment is proposed to overcome the issue of the curse of dimensionality in big data. In order to reduce processing overheads while achieving high prediction accuracy, feature importance is considered in preprocessing before building the prediction model. The effectiveness of the reduction techniques is conducted by evaluating the two features reduction techniques: Information Gain (IG) and Principal Component Analysis (PCA). These techniques reduce the time and storage space required for PBA system. The

dimensionality reduction issue bear the curse of dimensionality relates to how adding more input features makes a predictive modeling task harder to model. The analytics algorithms and techniques built on big data can generate unsatisfactory results when data quality issues creep into big data systems. These problems can become more significant and harder to audit as data management and analytics system attempt to pull in more and different types of data. The dimension reduction techniques, PCA based technique (DR_PCA) and Information Gain based technique (DR_IG) are used for demonstrating the proposed system outperforms the other system with respect to the prediction accuracy, particularly for considering the feature importance of the datasets.

Another contribution of the thesis is hyperparameters optimization to enhance the scalable random forest algorithm. The number of predictors required for obtaining outcome predictors is minimized to improved efficiency. An interest is often to determine the most important predictors that should be included in a reduced, parsimonious model. This can be achieved by performing variable selection, in which optimal predictors are identified based on statistical characteristics such as importance or accuracy. Developing prediction models with optimal hyperparameters can reduce the training overhead and improve efficiency of prediction in practice.

Moreover, real-time PBA system is proposed on Big Data analytics platform to make informed investment decisions and its potential profits based on the prediction of stock market trend. Daily real-time stock data are captured with Pandas Data reader and stored in HDFS. As a technical analysis, the values of technical indicators are calculated based on the stock closing prices and then that values are used as input features of the training dataset. Market price movements are not random, but usually move in patterns and trends that repeat over time. Stock trading periods are classified as four periods: inactive, active, sub-active and strong active periods. The distinctive forecasting patterns of inactive, active, sub-active and strong active stocks are considered and the predictive powers of ESRF in different trends of activity for each stock are examined.

1.7 Overview of the Thesis

The main issue of this thesis is developing an efficient PBA system. To solve this issue, PBA system on distributed environment is proposed to extract valuable

information from huge quantities of data in an efficient and timely manner. The second issue is choosing the compatible processing engine for PBA system. The two popular processing engines, Hadoop MapReduce and Apache Spark are compared and analyzed for faster computing the massive growth of Big Data. Both frameworks are Apache-hosted data analytics frameworks, but performance varies greatly depending on the use case at the time of implementation. Therefore, more effective processing engine is chosen to develop the PBA system. The third issue is necessity of reducing the high dimensionality of Big Data. The effectiveness of the reduction techniques are conducted by evaluating the two features reduction techniques: Information Gain (IG) and Principal Component Analysis (PCA). The fourth issue is to choose the suitable SML technique based on the experimental datasets. The performances of four SML techniques are evaluated and selected to get the best predictor. In order to achieve the high accuracy of the PBA system, the performance of selected algorithm is enhanced. Finally, to solve the real-time problem of PA, the proposed Real-time PBA system (RPBA) is implemented by making the analysis timely and affordable. The experimental results show that Real-time PBA system achieves the highest accuracy 99% using strong-active stocks for long term prediction.

The work performed in Scalability Test for Big Data has been published [P1]. The work performed in Processing Engine Selection has been published in [P2]. The works related to the PBA system development have been published in [P3 and P4] of Author's publication Section.

1.8 Organization of the Thesis

This dissertation is organized with six chapters and it is structured as follow.

Chapter 1 outlines the study areas, Big Data key concepts and problem issues, motivation factors, objective and contribution of the thesis.

Chapter 2 discusses the state-of-the-art techniques and the related works of this thesis to construct the theoretical foundation of the study.

Chapter 3 highlights the Scalability test for Big Data, Processing Engine Selection and then choosing the best SML predictor for the PBA system.

Chapter 4 presents the implementation of proposed PBA system for handling the high features in bulk of big data. In this chapter, an in-depth look is taking at finding the optimal hyperparameters for SRF prediction models and model selection.

Chapter 5 provides the detailed description of Real-time PBA system for forecasting the stock market trend.

Chapter 6 concludes the overall research work and the effectiveness of the research by discussing the results, describing the scope and limitations of the research, and ultimately points out the direction of future research.

1.9 Chapter Summary

The explosive growth of data is fueled by the exponential growth of the internet and digital devices. Advancement in technology is making it economically feasible to store and analyze huge amounts of data. On the other hand, Big Data brings many complex situations such as performance problems, technical challenges, management strategy, and memory consumption besides profitable well beings. PA encompasses a variety of statistical techniques from modeling, SML, and data mining that analyze current and historical facts to make predictions about the future or otherwise unknown events. PA provides a methodology for tapping intelligence from large data sets. Therefore, the important role of big data technologies such as big data analytics platform and PBA techniques are discussed in this chapter. Moreover, the two main SML methods in predictive analytics are described to transform massive data into actionable intelligence. This chapter describes the motivation factors, problem statements, objective, and contribution of the thesis. The next chapters will present the detail explanations of the thesis.

CHAPTER 2

LITERATURE REVIEW

Big data and machine learning are the priority of data analytics in this fast-growing digital world. Big data is a broad collection of digital raw data that is complicated for conventional tools to gather and analyze. As digital information expands extensively in various forms, configurations, and scales, handling this vast volume of data is critical according to the organizational requirements. In addition, traditional tools can't even handle this bulk of information. Therefore, by using predictive analytics for exploration, modeling, data processing, tracking and many more operational requirements, various organizations have created products, making it an important aspect in data science. The significant problem of predictive analytics is to derive useful trends from the vast amount of data which can be used in decision - making processes and prediction.

With the amount of data increasing at an unprecedented rate, a system must be developed which is capable of harnessing the data and extracting value from it. However, there is still potential for exploring new predictive models that produce results with greater accuracies while still remaining computationally feasible using the predictive analytics method. It brings also a number of additional possible problems that emerge when the massive datasets have a large one with a tremendous number of dimensions. Many such dimensions trigger almost any observation to occur slightly further away from all others in the training dataset.

Recently, however, research has integrated diverse machine learning with high-speed hybrid learning and training frameworks to process data. Most of these strategies are scenario-specific, and therefore display low performance in common scenarios and learning features in big data, based on dimensional space. Furthermore, one of the reasons for such failure is the high human involvement in designing sophisticated and optimized algorithms based on machine learning techniques. In this research, we bring forward an efficient PBA system based on enhanced SRF algorithm for handling the curse of dimensionality in scalable machine learning. In

the following, the previous research area of academic literature is reviewed with respect to the proposed system.

2.1 Study of Predictive Analytics

Predictive analytics (PA) is a technical term that covers different mathematical and computational techniques used to create models that forecast upcoming events or actions. The structure of those predictive models will reflect on the expected action or event. Most predictive models generate high predictor scores indicating the more likely occurrence of certain acts or events. Data mining is a part of predictive analytics involving data analysis that identifies trends, patterns, or data relationships. The knowledge can be used in the predictive model construction. PA depends on more and more complex statistical methods, including the most linear models and data mining techniques, along with multivariate computational techniques such as sophisticated estimation and historical data models. Organizations may use these approaches to detect patterns in the data that are not readily exposed but can help predict future events or behavior.

The machine learning techniques used in PA are computationally intensive. Machine learning builds upon insights such as these in order to develop predictive capabilities, following a number-crunching, trial-and-error process that has its roots in statistics and computer science. Depending on the amount of training data, some require performing thousands or millions of calculations. Advances in computer technologies perform such calculations, allowing insures to efficiently analyze the data that produce and validate their productive models. The validity of any predictive models depends on the quality and quantities of data available to develop it. More and more, PA drives commerce, manufacturing, healthcare, government, and law enforcement.

Healthcare is a field where predictive analytics could have a huge impact, yet the field has lagged in both the adoption and the application of this technology. Healthcare has always used capabilities of PA, but mainly for simple accounting, reimbursement, actuarial, and fiscal projection purposes. More advanced machine learning and predictive analytics techniques have only recently begun to be deployed. The hidden purpose of PA in medicine is to predict and direct decision-making in

diagnosis and treatment. Among researchers in health prediction system, J. Ngo et al. [2] studied to create a PBA capability evaluation tool that can quantify the capacity of building organizations and discussed the organization's strengths and weaknesses to provide a benchmark in the process of implementing PBA system. Twenty-one determinants have been defined and analyzed in terms of their effect on the capacity of an enterprise to incorporate the PBA system. These determinants were grouped into five determinant classes and assigned weights, to form the basis for the method for assessing the potential of big data and predictive analytics. The developed tool was then validated with four construction organizations to represent the levels, strengths and limitations of their big data and predictive analytical capabilities. The results of this study contribute to awareness and practice by defining the determinants that impact the capacity of building organizations to embrace forecasting and creating a computerized evaluation tool that also serves as a benchmarking tool for building organizations in implementing PBA system.

PA has huge potential to create value to the construction industry such as in the selection of optimal site location [3], in predicting project delays [4], and to predict energy consumption of buildings [5]. Du. Dinisa et al. [6] proposed a predictive analytics tool (ForeSim-BI) to resolve the problem faced by maintenance organizations in predicting the workload of future maintenance activities and preparing an appropriate capacity to cope with the workload they would expect. Within the unique background of a Portuguese aircraft maintenance company, they planned to establish this method as a predictive analytics perspective. For projecting future and unparalleled maintenance workloads from observational data, a Bayesian inference is used as a forecast module. And then preceding workload assessments are converted into predictive forecasts once data on the expected maintenance interventions become accessible. They also used a simulation system to define the total workload through sets of random variables, including forms of maintenance work, phases of safety checks, and skills in management. A linear programming model was also being developed to ensure the quality of Bayesian network-supported decision-making processes. The evaluation was performed using the real industrial data and simulation model. They showed that greatly more precise forecasts were obtained with the proposed tool, due to the significant cost-saving capacity for maintenance organizations.

PA has also been helping the agricultural sector by providing insight and knowledge that could make farmers and other farm supply chain members more aware of the risks. In [7], the authors compared and analyzed the three machine learning techniques to predict the monthly prices of crop. In order to understanding the trends and seasonality of the price data, SARIMA, Holt-Winter's Seasonal method, and LSTM neural network have been examined using the ten years Arecanut dataset. Their exploratory analysis indicated that the LSTM model was the most parsimonious model to fit the non-linear dependence of the experimental data than the other techniques.

The computerization of law enforcement creates the opportunity to fully understand legislative and institutional processes execution. PA has an ability to provide the intelligence needed by police forces and people around the world to forecast violence both in real time and in the future. PA supports as a tool in political procedures which make it feasible to define the relationship between the features of the civil system and their connection to the scientific, technical elements and also environmental aspects. O. Metsker et al. [8] presented the analysis of court decision data and machine learning methods in law enforcement industry. They introduced the PA technique to identify the correlation analysis of crime compositions, the analysis of the size of penalties, types of criminals. They concluded that the proposed PA technique gave outstanding results in terms of intellectual analysis of the practical results of the law change with 89 % ROC curve.

2.1.1 Traditional Predictive Machine Learning: Supervised

For organizations overflowing with data but struggling to turn it into useful insights, predictive analytics and machine learning [24] can provide the solution. Predictive analytics using supervised machine learning algorithms are most commonly used for security, marketing, operations, risk and fraud detection. Two types of supervised learning include classification and regression techniques. In [46], the authors presented an analytics framework for STEM student access. They used conventional random forest predictors to deliver student assessment for the stakeholders by identifying key input thresholds, quantifying the impact of inputs on student success and evaluating the results of students. They intended to enhance the STEM graduation success by addressing the risk level of each student in effort. S. M.

Idress et al. [70] presented an effective model to forecast the time series data regarding the sales of vehicles in US. They used the ARIMA model proficiently enough to handle the time series data. The predicted time series of the proposed effective model were compared with the actual time series, which exhibits an aberration of nearly 5%, mean percentage error with respect to actual sales data on average.

In [69], the authors introduced a supervised learning based predictive analytics model for delivering the prevention services of school. Student information and patterns were extracted using logical analysis of data model from teacher observation of classroom adaption (TOCA) screening observation. These patterns were then applied to determine a suitable population using diagnostics interview scheduler for children (DISC) tools. Experimental results showed that up to 91.58% of the cost of administering DISC would be saved by correctly identifying participants without conduct disorder and excluding them from the DISC test.

J. H. Joloudari et al. [9] proposed a feature selection based PA technique to predict the liver disease. Hence, the different prediction model of random forest, Multi-Layer Perceptron (MLP) neural network, Bayesian networks, Support Vector Machine (SVM), and Particle Swarm Optimization (PSO)-SVM are compared to improve the performance of the model. They presented that the proposed PSO-SVM model achieved the best performance in terms of specificity , sensitivity, accuracy, Area under the Curve (AUC), F-measure, precision, and False Positive Rate (FPR) parameters. In addition, a 10-fold cross-validation method was used on a dataset of the liver disease to evaluate models. With regard to the above evaluation criteria, the hybrid PSO-SVM-based optimized model obtained the highest level of accuracy with the minimum number of features than the other techniques.

Q. Cui et al. [10] considered a new PA technique, max-linear model replacing the linear combination in the linear regression model by a max-linear structure. They compared the max-linear model with random forest, and sparse group lasso. To further uncover market indicators that are more important in predicting bond risk premia, they computed frequencies of non-zero estimates for each predictor among the ensemble models. The proposed technique is a natural extension of linear

regression model, where it has a considerably lower time complexity when compared to some of the other machine learning algorithms.

T.-Khoei [11] A. T.-Khoei proposed the two PA techniques which were based on independent prediction of complications with single-task learning and simultaneous prediction of multi-task learning complications. They experimented with a case study and compared the efficiency of these two approaches by predicting hypertrophic cardiomyopathy complications on 106 predictors in 1,078 inclusive electronic medical records from April 2009-April 2017. They have used logistic regression, artificial neural networks, decision trees and support vector machines to implement the proposed PA technique. The studies indicated that multi-task learning with logistical regression enhanced both discrimination and calibration efficiency of predictions.

2.1.2 Traditional Predictive Machine Learning: Unsupervised

The area of predictive analytics has taken a lot of prominence in the last couple of years due to the demands of analytical use cases, including text mining, image recognition [98], city planning and targeted marketing. This use cases differ from the predictive modeling use case because there is no predictive response measure; the analyst seeks to identify patterns but does not seek to predict a specific relationship. These use cases require unsupervised learning technique.

In [4], they proposed unsupervised learning technique based on local clustering and bootstrap aggregation. Firstly, Variational Bayesian Gaussian is applied in this ensemble learning approach for discriminating the outputs and driving the weighted values of clustered simulators output. Local cluster-weighted Bootstrap Aggregation method is performed for serving the purpose of weighted combination of the clustered ensemble of outputs from the individual simulators. Based on the experimental results, they highlighted that the number of input bin size, sample size, output dispersion and level of agreement amongst the simulators can affect the performance of proposed method. Moreover, the proposed method is evaluated with three different methods: classical Bagging, Bayesian Model Averaging and Stacking of predictive distributions. They demonstrated that the proposed method succeeds in generalization error to 0.511.

Many predictive analytics environments have implemented methods of extracting data from an enormous amount of big data. Based on the historical data, data analysts can run analytics to forecast the future trends. However, in many cases, computational costs of predictive data analysis call for increasingly efficient methods of determining which features and variables are most relevant in overall data [36, 37, and 38]. To solve this problem, L. C. Carly et al. [46] proposed the integrated technique to cluster the same groups within the L1000 Landmark genes dataset. PCA is used to select relevant features and k-means algorithm is applied to perform clustering. They presented that the clustering results of integrated methods can improve in predictive analytics of microarray data.

H. Li et al. [12] proposed an unsupervised PA technique to predict treatment response and survival of NSCLC patients receiving stereotactic body radiation therapy. Firstly, they developed the unsupervised two-way clustering method using a technique of matrix tri-factorization and simultaneously extracted meta-functions based on image data. This research was conducted on the basis of a dataset 18F-FDG-PET of 100 consecutive patients treated with SBRT for early stage NSCLC. Each patient's tumor had 722 radiomic characteristics. The proposed technique was carried out and at the same time groups of patients and radiomic characteristics were identified. In terms of survival and the freedom from nodal failure, patient groups were contrasted. For building survival models, meta-features were determined to predict survival and free from nodal failure. They demonstrated the disparities between 2 patient groups when the patients were grouped into 3 groups in terms of both survival ($p = 0.003$) and nodal failure independence ($p = 0.038$). Values of average concordance for predicting survival and nodal failure were $0{:}640 \pm 0{:}029$ and $0{:}664 \pm 0\ 0{:}063$ respectively, higher than those obtained from predictive models based on clinical variables ($p<0.04$). Results of the analysis showed that the proposed PA technique has been achieved to stratify patients and predict survival and liberation from nodal failure with better results than existing alternative approaches.

Whereas supervised learning leverages the error between the predicted patterns and the actual patterns from the data, unsupervised learning relies on internal metrics computed from the input data such as similarity measures, densities, and probabilities. In [12], the researcher explored the usefulness and feasibility of

unsupervised machine learning algorithms by detecting anomalous activity in heterogeneous network sensor data to detect possible cyber-attacks. To discover anomalies in a network-defense context, two methods were applied, namely: to detect rarity and to detect novelty. The former method is to locate the behaviors of anomalies which are least commonly found in a series of observations. And another is detecting events with the lowest estimated likelihood of occurrence, based on usual data prior observations. The performance of any chosen anomaly method can vary depending on the nature and procedure of the attempted malicious behavior. The malicious activities simulated for this study have been chosen for relevance to cyber security and for the variety of methodology of attack, but they are not nearly exhaustive of the space of threat. Consequently, the findings obtained in this analysis are illustrative but by no means indicative of success against zero-day attacks in the real world.

2.2 Big Data Wave and Predictive Analytics

The term "Big Data" was initially stated by John Mashey in 1997 in the context of a problem statement—the volume of data was fast becoming too excessive for available computer systems to store and process [1]. Big data usually includes data sets with sizes beyond the ability of commonly used software tools to capture, manage, and process data within a tolerable elapsed time. Data was rapidly becoming too large to store and process available computer systems. Pretty soon after that, due to the existence and exponential growth of big data wave, the world started to pay more attention to the volume and importance of the data. Big data has emerged as a new trend for the gathering and processing of vast volumes of data. Combined with scalable machine learning and parallel processing engines, big data processing has contributed to a new technology class called "Big data analytics".

Predictive Analytics (PA) is a special type of big data analytics and Predictive Big data Analytics (PBA) suffers from different challenging issues related to data collection, preprocessing, scalable distributed processing and enhanced decision-making process. Big data can gain an enormous advantage to any organization when used with PA which enables to make really strategic decisions at a great rate [39]. It is fundamentally a road map to better proprietary solutions. PBA is the advanced analytics which used the scalable machine learning algorithm to forecast the risks and

opportunities for future based on the huge historical data [40]. Machine learning is used to build predictive models by extracting patterns from the bulk of big data [21], [62]. Traditional techniques become computationally unfeasible for big data which exceeds the processing capabilities. Moreover, the bulk of big data are directly influence the predictive models of the PBA system and the capable of the system's output. The PBA system entails determining the best big data features and computing framework, which is then supplemented with scalable machine learning techniques.

2.2.1 Scalable Predictive Machine Learning: Unsupervised

Data volume is expanding quickly with the development of the technology. Effective analytical techniques are utilized for making intelligent, data-based decisions of PBA system. Data clustering, a popular data mining analytical tool is being utilized efficiently in data processing. For evaluate large data sets, the motivation for today's scenario is to develop the conventional methods.

R. Parasad and C. Aruna [82] proposed a scalable and flexible big data analytic Framework (SFBAF) to solve the problems of the data transformation and knowledge extraction of big data processing. The authors used the MapReduce based KNN clustering algorithms to find the relevance data as clusters. They highlighted that the scalability and flexibility of big data computing was improved using the separate ETL tools.

The authors from [4] proposed an enhanced grey wolf optimizer (MR-EGWO) as an effective clustering tool using MapReduce. They presented an enhanced Gray Wolf Optimizer for clustering large-scale data sets. Therefore, the proposed technique is used as a prominent analytical tool to cluster automatically.

N. Kolokas et al. [13] presented a scalable PA technique which is a generic methodology for fault forecasting or prognosis in industrial equipment. Particularly, the proposed technique used the unsupervised machine learning model and considered the significant feature importance in automatics feature selection phases. They applied the real industrial data related to aluminum and plastic production. Their simulation and experimental results presented the strong evidence that the proposed technique was capable of successfully forecasting the upcoming faults before the observation, despite the general difficulty to find the useful information for fault forecasting.

2.2.2 Scalable Predictive Machine Learning: Supervised

The tremendous growth of big data that need powerful analysis tools for efficiently processing. This is mainly due to the traditional analytics tools which are not fit for executing the exponential growth of structured and unstructured data [63]. PBA is the process of investigating big data to uncover useful information for better decision making. Information is valuable resource when building predictive models in the pursuit of the profitable PBA system [48]. When considering the problem of knowledge extraction from big data, the literatures [58] mostly consists of empirical work for decision makers.

N. Kann et al. [58] proposed a PAD-SVM system, the PA system using Support Vector Machine algorithm which is implemented on Pydoop architecture for demonetization data. This proposed system analyzed the current available sentiment in the tweets and predicted the pattern of the data to find its relative future trend. The predicted model of the proposed system obtained the accuracy of 96%. The graphical representation of the values showed that the 40.08% of people opposed the idea of demonetization, 31.81% conflicted about the idea and 28.11% of people agreed the idea of demonetization. In [5], the authors presented the Hadoop MapReduce based platform for PA system. They stored the three billion raw health data in HDFS and computed these data using existing WestGrid infrastructure. They found that data ingestion to HDFS required only three seconds and HBase took four to twelve hours to complete the Reducer of MapReduce. R. Parasad and C. Aruna [83] proposed a scalable and flexible big data analytic Framework (SFBAF) and addressed the problems of the data transformation and knowledge extraction of big data processing. The authors used the MapReduce based KNN clustering algorithms to find the relevance data as clusters. They highlighted that the scalability and flexibility of big data computing was improved using the separate ETL tools.

In big data era of rapid information growth, the emergence of high dimensional data becomes the curse of dimensionality in PA system. Popular shortcomings are the heavy processing time and generalizability reduction in ML. Due to this shortcoming factors, many researchers highlight various innovations to address the challenges of high dimensional data [6]. Generally, high dimensionality reduction techniques are categorized into two main subdivisions of feature extraction [67] and feature selection

[28]. Feature extraction methods transform the multidimensional space into lower dimensional subspace. In fact, by combining the existing features, fewer subsets are extracted, and therefore, the new latent features have the information contained in the original observations. In [55], the authors proposed a framework for selecting relevant and non-correlated feature subsets in cervical cancer dataset. In this paper, principal component analysis (PCA) is used as feature reduction technique to perform non-correlated feature selection and Decision Tree C4.5 algorithm are applied for the classification. Experimental results showed that the proposed framework was robust to enhance classification accuracy with 90.70% accuracy rates. T. Zang and B. Yang [79] proposed a new approach which is based on PCA for dimension reduction of big data. Linear regression approach is used for prediction in the follow up analysis of PCA. Unlike the other PCA based technique, they presented the PCA approach without computing principal components. They concluded that proposed approach using SVD significantly reduced the computational cost for big data processing. Z. Wu et al. [98] proposed a distributed PCA based feature extraction technique for hyperspectral data on Apache Spark platform. An experimental comparison is performed on Hadoop platform with MPI version of PCA. They found that the two platforms could provide the same results and the only difference was the computing performance. By taking the advantages of in-memory computing, the experimental results showed that the Spark based platform can provide faster data processing time than the Hadoop Platform. For effective prediction of network intrusion detection system, PCA was used as dimensionality reduction technique by determining its reduction ratio on two benchmark datasets. They highlighted that the prediction accuracy for 10 Principal Components was about 99.7% and 98.8%, nearly same as the accuracy obtained using original 41 features for KDD and 28 features for ISCX, respectively.

I. Tsamardinos et al. [31] proposed the parallel, forward–backward with pruning (PFBP) algorithm for feature selection in high dimensionality of a dataset. PFBP partitions the data matrix both in terms of rows as well as columns. By employing the concepts of p-values of conditional independence tests and meta-analysis techniques, PFBP relies only on computations local to a partition while minimizing communication costs, thus massively parallelizing computations. O. Alfarraj et al. [60] introduced the optimized feature selection technique using the fireflies

gravitational ant colony optimization (FGACO) approach to reduce the high dimensionality of PA system. This optimized technique could examine the feature importance [1] and characteristics during the selection process. The selected feature consists of all details about the particular predictive analytics. The system's efficiency was then evaluated using four different datasets. The experimental results show that FGACO performs better in terms of the sensitivity, specificity, accuracy, and the number of selected features based on time. E. S. Mosseini and M.H. Moattar [20] proposed a hybrid feature selection method using interaction method based Evolutionary Feature subsets Selection (IIEFS). To improve the prediction accuracy, candidate's features and candidate feature pairs were initially identified. The candidate feature subsets have formed in an evolutionary manner, and then, the best candidate feature subsets have chosen as the optimal feature subsets. They compared the proposed method with the other state-off-the-art feature selection methods in terms of the prediction accuracy, F-measures and stability. The experimental results show that IIEFS not only reduces the features effectively, but also increases the accuracy and F-measures.

Given a scale expansion of scalable machine learning for the PBA system, the ML regression technique is crucial for estimating correlations between variables and identifying key patterns in big and diverse data sets. The most widely used regression methods are linear regression [10], ensemble regression techniques [19] and decision tree algorithm [76]. Random Forests (RF) are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time[25, 26]. ML based RF regression techniques have been widely used in PA system. Y Li et al. [94] proposed RF regression for online battery capacity estimation. The proposed RF was developed to learn the dependency of the battery capacity on the features that are extracted from the charging voltage and capacity measurements. To improve the prediction accuracy of the proposed RF, they applied an incremental capacity analysis for feature selection. Experimental results showed that the proposed method was promising for online battery capacity estimation of RMSE of less than 1.3% and low computational costs. In the current big data era, the computational and scalable requirements of RF need implementation to deal with large datasets. A. Lulli et al. [7] proposed a new RF implementation called ReForeSt on scalable distributed environment for arbitrarily large datasets.

They demonstrated that ReForeSt can provide better computational efficiency than RF from Spark MLlib [90] without reducing the accuracy of the prediction model.

The values of the hyperparameters for RF have a direct impact on the prediction model accuracy and computational time. In [57], the authors studied the effects of hyperparameters optimization phase over the efficiency of ML techniques: RF, GBT and MLP. The study was performed on five different datasets. They concluded that prediction errors were different depending on ML algorithms and hyperparameters optimization techniques could provide high prediction scores on 50% of the datasets. B. F. Huang et al. [9] highlighted about the parameter sensitivity of RF in prediction accuracy. Their finding was demonstrated that parameterization can enhance the accuracy of the prediction model. S. Bernard et al. [66] presented that parameterization of feature splitting parameter (K). They especially outlined that the optimal K highly influenced the relevant predictor variables. The experimental results showed that K allows to control the strength of the randomization in the split condition, in such a way that the smaller the value of K, the stronger the randomization.

PA plays an important role in big data era with their potential to ascertain valuable insights for improving decision making process. PA is increasingly becoming a trending practice in real time to fully harness the power of PBA system [1]. Real time PA is the knowledge extraction from big data in real time for decision support. F. Sun [22] proposed a real-time PA system for public transportation service as a decision support system. The system uses the streaming real-time bus position data and produces more accurate arrival time predictions by combining the clustering analysis and Kalman filters techniques. Experiments results showed that the system reduce arrival time prediction error by 25 % when predicting the arrival delay an hour ahead and 47% when predicting within a 15minute future time window. C. Su [15] proposed a real-time predictive maintenance system (HD Pass) based on Apache Spark for the detection of imminent hard disk drive failures in datacenters. The proposed system learnt failure patterns from previous data and applied these data to subsequently predict upcoming breakdown of targeted devices. HD Pass used RF to accurately predict HDD malfunction and obtained the prediction error rate of 0.1416 over real time Backblaze’s data.

F. Sun [21] studied the prices fluctuation in crypto currencies using user's opinion and real-time prices. They used the Long Short Term Memory (LSTM) methods to predict the future bitcoin prices. To improve the prediction accuracy of the training model, some important features were considered over twitter data. Their predicted LSTM model obtained the precision 60 % and accuracy 50% by considering the highly volatile market.

2.3 Difference between Existing and Proposed System

The aim of this section is to compare some of the existing solution and the proposed system. After carefully reviewing the paper, the significant finding is that there is seldom research aiming at exploring the effect of dimension reduction techniques and hyper-parameter optimizations simultaneously on SRF in the financial and security domain. Therefore, motivated by the aforementioned studies, the feature importance of all the input variables is considered before the training process and then a series of experiments is set up that contain dimension reduction methods and hyper-parameter optimizations simultaneously, thereby exploring an accurate and comprehensive predictive model based on enhanced SRF technique. The proposed system used the automated hyper-parameter optimization method in SRF to overcome the drawbacks of manual search. The superiority of enhanced SRF over the SRF from Spark MLlib is evaluated via mean absolute error and mean square error. Moreover, the effect of two different dimension reduction techniques as well as hyper-parameter optimization methods on the model performance is comprehensively investigated through real-world experimental datasets. Finally, the features are ranked according to their importance score to enhance the model interpretation.

2.4 Chapter Summary

This chapter analyzes the current research works in traditional PA with conventional computing paradigm, PA with parallel computing paradigm: Hadoop MapReduce, Apache Spark Platform and Real-time PBA. Moreover, high dimensional reduction techniques are reviewed. The aim of this chapter is to highlight the actual requirements and efforts of the research area. This research emphasizes only on developing the scalable efficient PBA system in Big Data environment, and

thus various predictive machine learning techniques in existing trends are investigated to justify the design and develop the proposed PBA system in Big Data environment.

CHAPTER 3

BUILDING BIG DATA PLATFORM FOR PREDICTIVE ANALYTICS

With the popularity of electronic devices such as smart phones, laptops, computers, and increasing usages of social media and internet, the massive growth of daily generated data from different data sources have been rapidly increasing beyond the capabilities of traditional techniques. Predictive Analytics (PA) is becoming a critical asset for business manager, researchers and data scientists due to its quality of creating the usable information and knowledge from these data. The ability of PA to identify risks and opportunities from different data sectors could support the effective and valuable insights for decision making. The massive growth of data is forcing the researchers to bring the advanced computation to tackle the parallel and distributed processing. In particular, the task of selecting the processing engine and machine learning (ML) algorithm, which is at the heart of predictive big data analytics pipelines, has been challenged in the domain of scalable big data environment. In this chapter, PA is implemented on traditional analytics platform and big data analytics platform. Analysis of scalability test and processing engine selection are performed to develop an efficient PBA system. Then effective ML algorithm for PBA system is chosen by testing the prediction performance of four popular ML regression techniques.

3.1 The Growing Role of Integrated and Insightful PBA System

Digital transformation age is changing many industries that involve the way the data is interpreted and it is estimated that there are around 2.7 Zetta bytes of data in today's digital planet. By 2020, the data produced every second for each person will amount to approximately 1.7 megabytes and the data volumes will double every 2 years, thereby hitting the 40 ZB point by 2020. Interactive Data Corporation (IDC) reported e-commerce transactions B2B and B2C to reach 450 billion a day on the internet by the end of the year 2020. Big data are increasingly rising. The growth rate over the last decade has also been truly incredible, with the total quantity of huge data generated doubling around every single month. However, the amount of data is

becoming so great that traditional data analysis platform and methods can no longer meet the need to perform data analysis task in life sciences. As a result, both data analysts and computer scientists are facing the challenge of gaining a profound insight into the deepest analytics functions from big data. This in turn requires massive computational resources. Therefore, efficient computing platforms are highly needed as well as efficient and scalable algorithms that can take advantage of these platforms.

PA works as implicit analysis to predict the unknown future events and recovers the benefits of associated events. Though PA has existed for past decades, the power of PA is indeed growing. The core technology of PA is the ML analytics technique to convert the historical data into the prediction of the future. PA is used across the financial industry, educational sectors, healthcare services and fraud detection in a different way. By the use of advanced PBA tools such as Apache Hadoop, MapReduce, Spark, MLlib, Flume and Kafka, it has provided the value of awareness in a desirable form to improve the ability and actionable insights of PA applications [48, 49, and 50].

Several research studies of PA system have attempted to predict the future outcomes by using the advanced analytics techniques such as ML [52] and statistical analysis. To improve the effectiveness of PA system, choosing well-performing feature extraction and reduction techniques play a main important role to exploit potential efficient analytics. As the amount of data continues to grow exponentially, traditional tools are not able to address the issues of scalability and usability in big data era. ML algorithms can provide scalable techniques for analyzing large amounts of data at a broad scale. In the following subsection, the performance of ML on traditional analytics platforms and distributed analytics platform are analyzed and scalability test for big data with different computing nodes is performed. To develop the accurate PBA system, effective ML algorithm is selected by considering the prediction scores of ML techniques.

3.2 Performance Analysis of Predictive Analytics on Different Computing Platforms

Due to the exponential growth of big data, the computational ML field has taken significant strides forward with the availability of computing tools.

Conventional techniques of PA systems in data science typically have complex frameworks that seem to parallelize substantially more difficult. Computational ML methods are important to the advancement of experimental science in the Big Data era, and where algorithms and experimental techniques are built across from each other. Efficient and scalable platforms for even the most critical problems were rare and available. Therefore the effective use of PA platforms in big data processing will become increasingly important. This remains largely undiscovered territory, and is the underlying inspiration behind the work. In this work, the performance of traditional and distributed analytics platform is analyzed using ML algorithms.

3.2.1 PA on Traditional Analytics Platform (Conventional Computing)

PA exploits the unprecedented opportunities from vast amount of data and improves the decision-making process by identifying patterns and trends. By applying the machine learning techniques, PA constructs a predictive model that represents specific conditions between designated features and predictors. PA has achieved good performance for small datasets on a single machine by using traditional ML techniques. Figure 3.1 shows the process flow diagram of PA on traditional analytics platform.

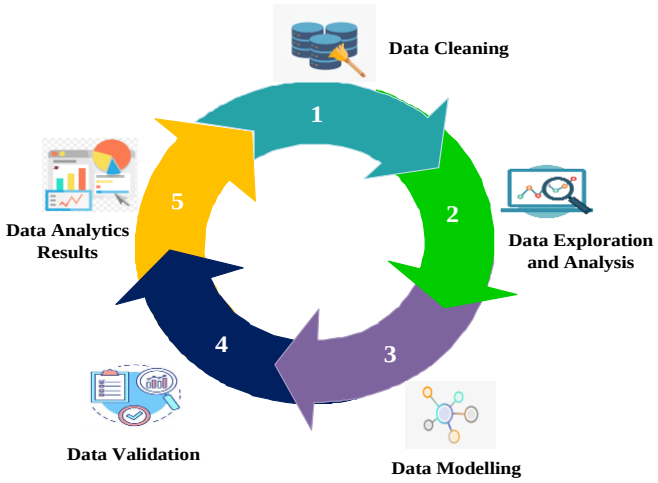


Figure 3.1 Predictive Analytics on Traditional Analytics Platform

The prediction model is built based on the training data and then the constructed model is conducted with testing dataset. Prediction is the task of predicting continuous values for given input. It is the form of data analysis that can be used to extract models to predict future data trends. Prediction also encompasses the

identification of distribution trends based on the available data. It can predict missing or unavailable numerical data values and this analysis can help a better understanding of the data at large.

3.2.2 PA on Big Data Analytics platform (Distributed Computing)

With the tremendous growth of big data, the issue of how to obtain valuables knowledge from these data becomes a challenge for data scientists. Predictive models are increasingly getting better in predicting the outcomes of big data through parallel computations that are based on scalable ML algorithms. With infinitely greater and more complex data volumes, scalable ML has begun evolving beyond their existing advanced analytics technologies and strategies. HDFS is a distributed file system that runs on commodity hardware and can handle massive data collections. Apache Spark is the most popular big data processing engine, with a wide range of features and capabilities. The Hadoop Distributed File System (HDFS) and Spark, when used together, can create a genuinely scalable big data analytics environment. Scalable NB (SNB) algorithm from Mahout Samsara is applied to predict the enormous data. Figure 3.2 shows the high-level architecture of SNB on distributed analytics platform. It consists of three layers: Application Layer, Processing Layer, and Storage Layer.

3.2.2.1 Application Layer

Naive Bayes, based on Bayes conditional probability rules, is utilized to perform classification tasks. The characteristic assumption of the naive Bayes classifier is to consider that the value of a particular feature is independent of the value of any other feature, given the class variable. Naïve Bayes from mahout Samsara retrieve the data as sequence file type. This algorithm fits the Spark processing engine particularly well as it only requires a fixed number of passes over the data, which compute easy-to-parallelize aggregates. Operations on in-memory matrices are enthusiastically performed, but operations on distributed matrices that are partitioned across the cluster's computers are postponed. The system saves the actions to be performed on these distributed matrices as a directed acyclic graph (DAG) of logical operations, with matrices as vertices and transformations as edges. The architecture of SNB on distributed big data analytics platform is shown in Figure 3.2.

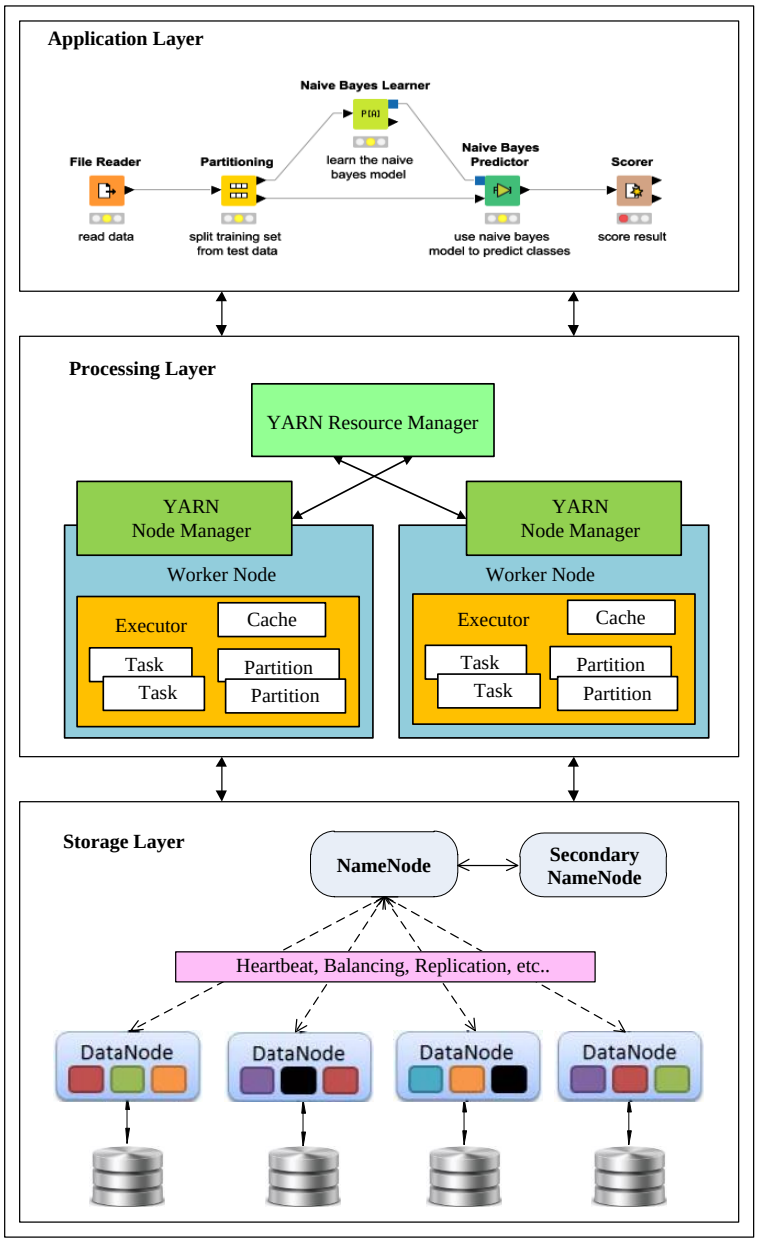


Figure 3.2 Architecture of SNB on Distributed Big Data Analytics Platform

In the training phase, SNB splits the given data set into training and test set. Predictive models are built based on training data, and the built model is run using a test dataset. SNB saves each vectorized document class label as a row key. It then extracts all probable document identifiers for each document. In the label assignment phase, SNB assigns labels to vectorized documents and predicts document classification. The structure of SNB is described in Figure 3.3.

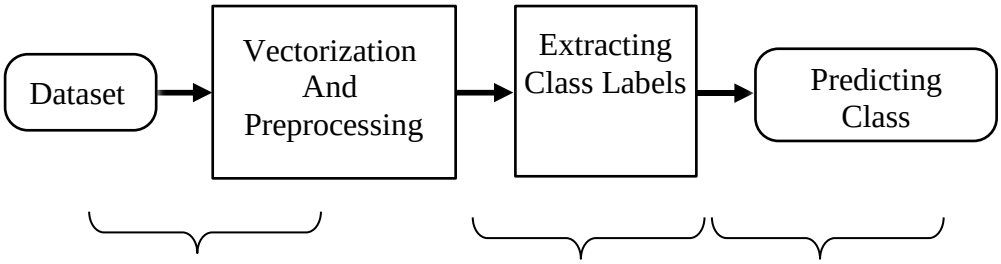


Figure 3.3 The Structure of SNB

3.2.2.2 Processing Layer

Apache Spark is used as a processing layer for distributed parallel computing. Apache Spark is an open-source big data processing framework built around speed, ease of use, and sophisticated analytics. The detail description about Spark processing engine for PA is described in section 3.3.2.

3.2.2.3 Storage Layer

HDFS is used for fast and memory efficient storage. When HDFS captures large amounts of data, the information is divided into individual blocks and distributed to different computing nodes in the cluster. The HDFS cluster works in a master – worker pattern with two node types, namely a NameNode (the master) and several DataNodes (workers). The NameNode manages namespace for the file system. It also declares the tree of the file system and the metadata of all the files and directories within the tree. The file-system mainstays are DataNodes.

In HDFS, the NameNode is the main point of contact. It keeps trying metadata on the file system. At a high level, the metadata is a description of all the files in the file systems that are mapped from each file to the file's list of blocks. On disk drive, this metadata continues to exist. As in many other file systems, the mapping from file blocks to the physical locations of the blocks is one of the significant attributes of the metadata that is built at runtime. The NameNode also handles client read / write access to the files. The NameNode takes advantage of the cluster nodes, the disk space that nodes provide and whether any node is dead. This relevant data has been used to arrange block replications for newly established files, as well as to keep acceptable file replicas.

The DataNodes are the slaves in the cluster. When a client requests that a file be created and writes data to it, the NameNode assesses specified DataNodes to write the data.For instance, if the replica of the file under construction is 3, a writing pipeline between the three DataNodes would be established. The blocks would be written to the first DataNode in the pipeline and DataNode would write the blocks to the next DataNode and in pipeline, etc. A work is considered successful when all of the replicas have been written successfully. This helps to ensure consistency with the data. Also the DataNodes serve up blocks when clients ask them to do so. They stay in touch with the NameNode, regularly send reports on disk utilization, and periodically send reports on blocks. The NameNode uses the block reports to map the blocks of a file to its locations.

Secondary NameNode in hadoop is especially dedicated HDFS cluster node whose primary purpose is to consider taking checkpoints of the metadata of the file system present on Namenode. It is not a namenode for backup and recovery. It is only check points the file system namespace of the Namenode. The Secondary NameNode is an assistant to the central NameNode but not a substitute for primary Namenode.

It is designed to improve the storage efficiency for big data. It supports data replication and distributes replicas to different servers. This allows PA to continue processing during data recovery. The willingness of using HDFS exclusively as a means of establishing a scalable and expandable file system to keep faster access to huge sets of data provides an appropriate value proposition from an information technology point of view: lowering the costs of unique to offer large-scale storage systems, having the capability to focus on commodity components, facilitating deployment using cloud-based services. Therefore, HDFS is much more efficient than the local storage strategy in terms of data uploading times, being an essential too in Big data environments.

3.2.3 Performance Analysis and Result Discussion

As stated before, this study aims to compare the effectiveness of a traditional ML and scalable ML methods and to analyze the scalability test on big data environment. To conduct these results, two-step statistical test procedure to carry out the performance comparisons. Firstly, this study addresses to the scalability test, considered to have a major importance, influencing the computational time implicitly,

the overall performance. For this purpose, a comprehensive experiment is carried out to compare the accuracy level and processing time of SNB with the traditional NB. WEKA 3.8 was utilized for traditional ML (NB) and Mahout Samsara was deployed for scalable ML (SNB).

Prediction models are increasingly being used with ML technique to inform decision making. Scalable ML is an emerging buzzword in the machine learning industry partially because it is an important and difficult feature of many machine learning initiatives to scale up machine learning processes. Secondly, scalable ML technique incorporated within a data analytics system may help estimate risk probabilities and effectively forecast for inventory and required production rates. All experiments were performed on a three-compute Hadoop cluster with one name node, and three data nodes.

3.2.3.1 Datasets Description

To analyze the functionality of ML on traditional and distributed analytics platforms, six real-world datasets, namely Breast-cancers, Iris, Adult [2], Movie Reviews [56], Reuter-21578 [85] and OHSUMED[61] are used for this experiment. The detail description of datasets is described as follow.

1 Breast Cancer Wisconsin (Diagnostic) Data Set is a medical record dataset from UCI machine learning repository. Features are computed from data digitized image of a fine needle aspirate (FNA) of a breast mass. It consists of the total number of 569 instance and 32 attributes.

2 Iris Data Set is also a real-world data set, which was collected by UCI machine learning repository. This is probably the most well-known dataset in the recent literature. The data collection comprises three classes, each with 50 instances, and each referring to a different variety of iris plant. One class is linearly separable from the other two; however, the former two are not linearly separable.

3 Movie Reviews Data set is a real-world sentiment analysis dataset. The files in the dataset are tab-separated and contain terms from the Rotten Tomatoes dataset. It is a corpus of movie reviews text analysis.

4 Reuter-21578 Data set is a public available version of well-known text analysis dataset. The dataset contains a collection of 10,788 documents from

the Reuters financial newswire service, organized into a training set of 7769 documents and a test set of 3019.

5 The Ohsumed Dataset is a collection of real-life medical abstracts organized by MeSH categories. The Ohsumed dataset is being used to classify the 23 cardiovascular disease classifications. The unique abstract number becomes 13,929 after selecting such a category subset (6,286 for training and 7,643 for testing).

3.2.3.2 Comparison Results (Traditional vs. Distributed Computing)

SNB runs in a distributed Hadoop cluster environment. To illustrate the advantages of the scalable NB model, it is presented to compare the traditional NB competitor (multinomial naive Bayes algorithm). The Weka workbench [89] is employed to implement the traditional NB for performance analysis. Figure 3.4 shows the exact measurement results for each data set.

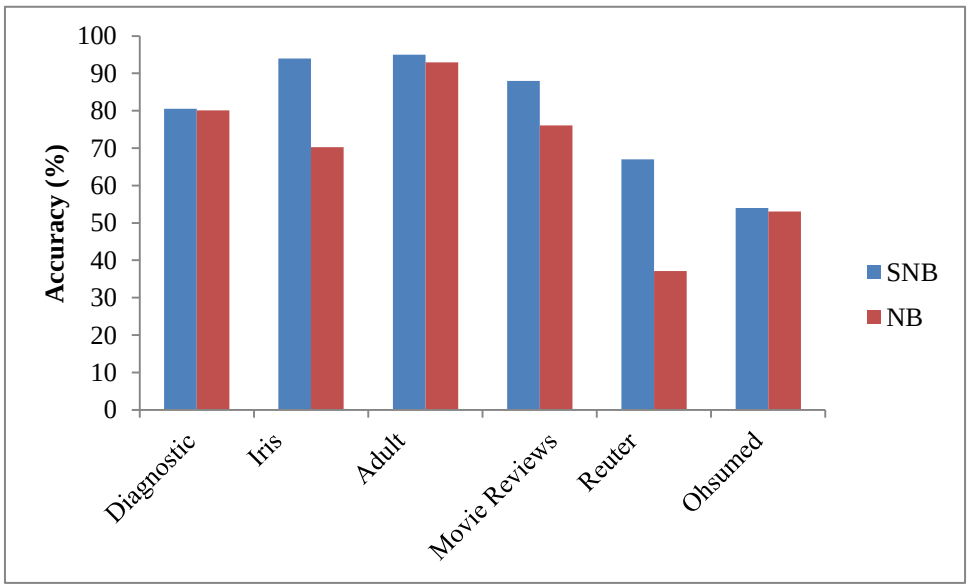


Figure 3.4 Comparison of SNB with traditional NB

According to the comparison results, it can be observed that SNB spends relatively less computational time than original NB, which is consistent with the fact that SNB bring the advantages of Mahout Samsara to improve the quality of predictions. The experimental results show that, for the given dataset, SNB performs better than traditional NB. For Reuter and Ohsumed dataset, it occurs when the class represented in a problem shows a skewed distribution. This case study may be due to

rarity of occurrence of a given concept, or even because of some restrictions during the gathering of data for a particular class. Therefore, SNB achieved higher accuracy than the traditional Naive Bayes classifier with six data sets.

3.2.3.3 Scalability Test for Big Data Analysis

In this section, the SNB scalability test is performed on different compute nodes of Hadoop cluster. The scalability of SNB is analyzed in terms of accuracy and processing time using Enron's email spam dataset. To verify the scalability of SNB on distributed big data analytics platform, the Enron email dataset is expanded due to its format. The detail presentation of Enron dataset is described in Table 3.2.

Table 3.2 Summary of Enron Email Datasets

Enron Dataset	Number of mails	
	Spam	Ham
E 1	8996	13545
E 2	12671	15045
E 3	17171	16545
Total number of mails 83973		

Figure 3.5 demonstrates the accuracy level of the SNB model in a Hadoop cluster computing environment, with 1, 2, and 3 data nodes. The results show that for each data node, SNB provides results with different data sets for better than 98 per cent accuracy.

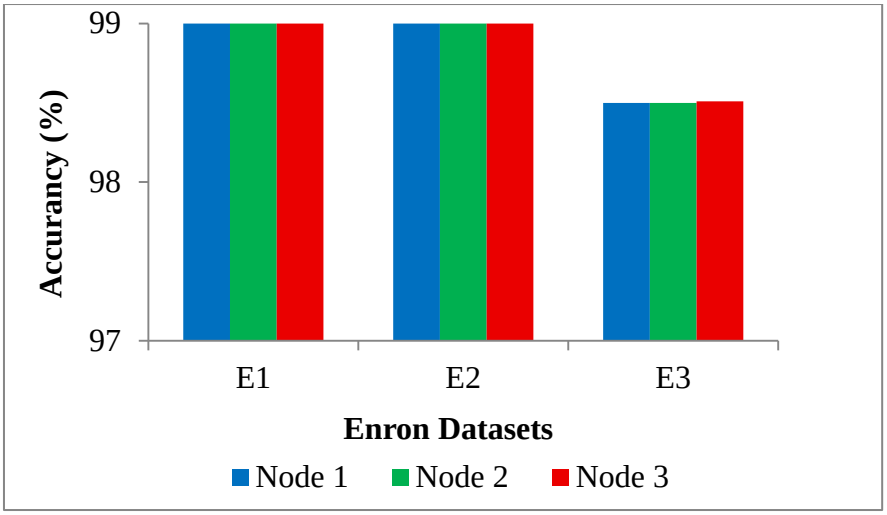


Figure 3.5 Performance Comparison of SNB on Each Node

Figure 3.6 shows in detail, for the prediction models of SNB, how the computational time changes according to the number of nodes. Multiple data nodes are parallelized to process large amounts of data. Processing time is reduced to a minimum and high-speed distributed computing is provided. From Fig. 3.6, it is clear that the number of distributed node has a prominent impact on the performance of the prediction model SNB. Specifically, the number of computational time alleviate as the number of data nodes increased.

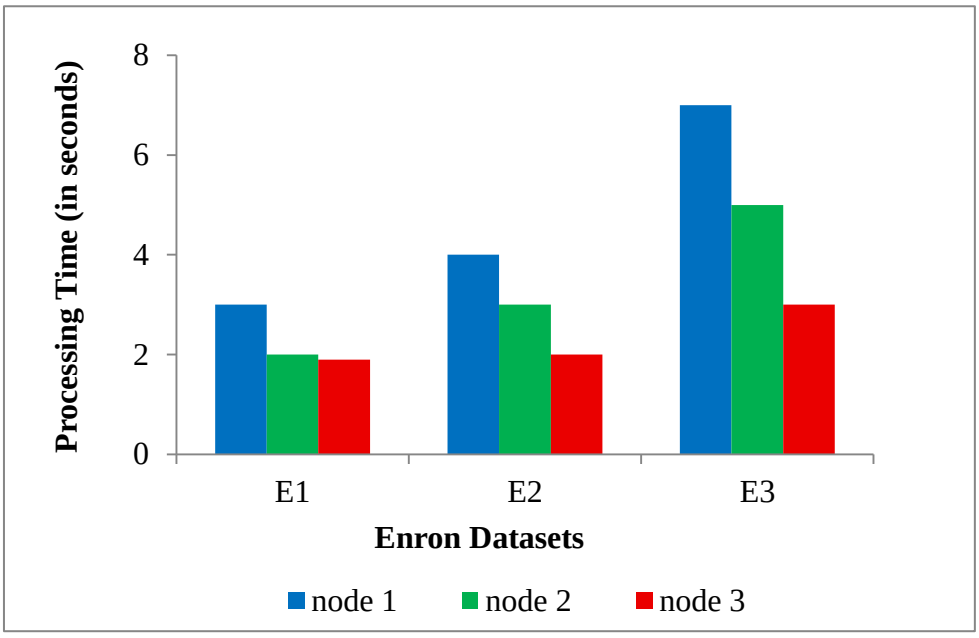


Figure 3.6 Processing Time Comparison of SNB

Scalability has become one of the Big Data core concepts. An effective distributed predictive model is necessary to manipulate massive amounts of data adequately. To get results, the difficulty of the analytical process is required to take as little time as possible. SNB performance in the distributed analytics application is discussed in this section. A scalable ML algorithm can accommodate vast amounts of data, and simultaneously perform multiple tasks. Research comparative study demonstrates that SNB is more reliable and faster on distributed platform than conventional NB. It also provides excellent performance that is scalable in a distributed environment.

3.3 Procession Engine Selection

Big data refers not only to large volume of data, but also to a series of technologies that turn large amounts of data into valuable information. Traditional data mining techniques are inefficient in handling the huge amount of big data. Therefore, an efficient and parallel processing engine is required to process and analyze such data. The processing engine is at the heart of PA functionality and can empower the analytics system to provide more relevant and accurate predictions. In this section, an effective processing engine is selected for developing the proposed PBA system by comparing two common parallel processing engines, Apache Map Reduce and Spark.

3.3.1 MapReduce Processing Engine for PA

MapReduce has been widely applied to reliable, scalable and maintainable applications of PA on commodity clusters. It is specially designed for handling problems that can be parallelized across a large dataset using a lot of computers (nodes) collectively called clusters. Each fragment can process on a node in the distributed cluster, and the results are ultimately aggregated. Despite its popularity, MapReduce has some limitations for scalable ML algorithms. For distributed parallel computing of large data, MapReduce consists of two main phases: map and reduce. Map takes a set of data and converts it to another set of data. Individual elements are broken down into <key, value> pairs, and Reduce takes the output from the map as input and processes it further. More specifically, in the Map phase, each node applies in parallel a Map function “Map()” to each pair of data from its partition and produces

a list of pairs that are stored in a temporary storage. There is a stage termed "Shuffle" in between Map and Reduce phases that is essential for grouping the pairs generated by the map tasks to the same key. Finally, in the Reduce phase each node applies in parallel a Reduce function “Reduce()” to each group generated in the previous phase and produce the corresponding pair as the final output.

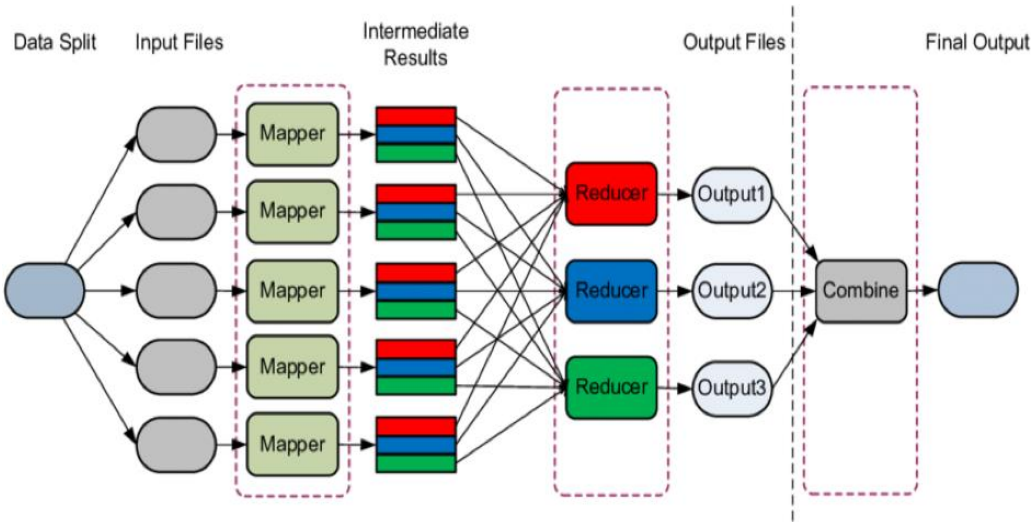


Figure 3.7 Hadoop MapReduce for Predictive Big Data Analytics

Figure 3.7 shows a typical MapReduce execution for PBA system. Intermediate results and output results for each phase are stored in HDFS. Because MapReduce requires a lot of time to perform these tasks, latency is high. This can be computing overhead and each job consumes a lot of space, so it is costly to run map reduction jobs repeatedly.

3.3.2 Spark Processing Engine for PA

Data processing and analysis is becoming more important, based on a very successful framework that was introduced to calculate large datasets. Thus, PA with Spark is gaining more and more attention to gain insights from large datasets. Apache Spark is a fast and popular engine which supports in-memory computing for large-scale data processing. Spark absorbs the benefits of Hadoop MapReduce. It makes the incredible things with predictive analytics and ultimately creates new growth opportunities. Figure 3.8 illustrated the components of Spark processing engine for PBA system.

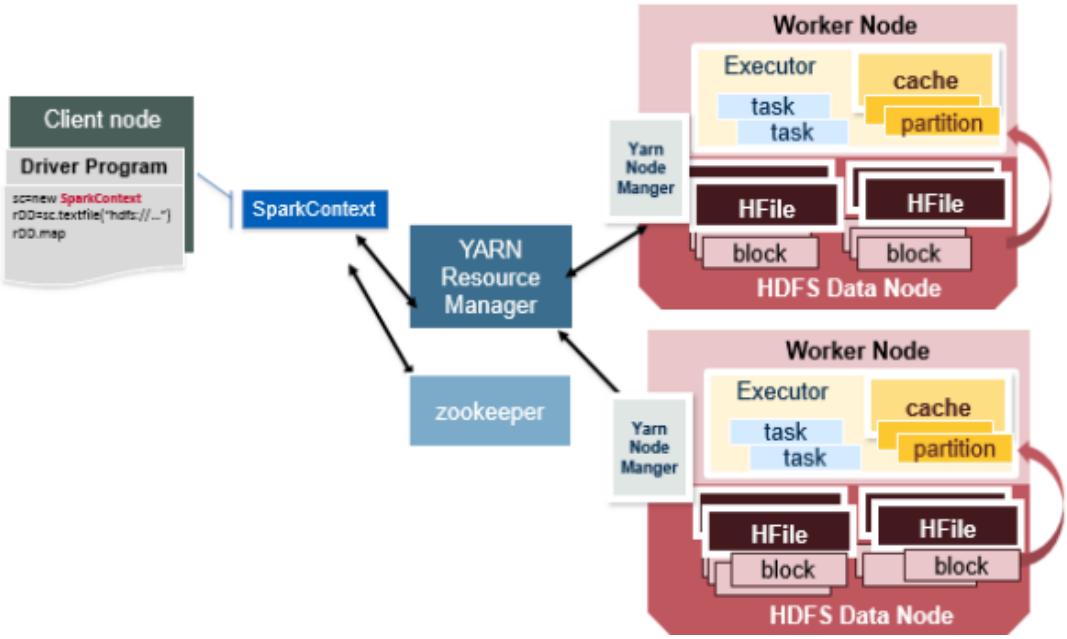


Figure 3.8 Spark Processing Engine for Predictive Big Data Analytics

The components of a Spark application are *driver*, the *master*, the *cluster manager*, and the *Executor(s)* running on the workers. Spark Context can communicate to different types of cluster managers, which assign resources throughout the application, particularly for cluster running. When linked, Spark will perform a task and receive the executor from the cluster nodes, which is a process initiated for the operation of the worker node to store work in memory or disk storage. The life of a Spark application begins and ends with the Spark driver. The *driver* is the process that a client uses to request application in Spark. It is also the duty of driver to schedule and organize the execution of Spark programs and to return the status and/or results (data) to the client. The driver program must be an executable program to receive and accept incoming connections during its lifetime. The Spark driver is responsible for Spark Session generation. The SparkSession object reflects a connection to a Spark cluster. SparkSession is instantiated and used programmatically at the start of a Spark application that includes an interactive shell.

Spark's core abstraction is a Resilient Distributed Dataset (RDD) with better computing power and fault tolerance. The first step of distributed computing on Spark is to break the input data into batch fragments and then transform these data segments into RDD. Operations on the incoming data are converted into operations on different RDD groups. The entire calculation process always produces intermediate data, which

can be stored and discarded according to specific needs [22]. This allows Spark to store the data cache in memory and perform the same data calculations and iterations. This saves a lot of disk I / O operation time. A scalable PBA system on Spark analytics framework can potentially run on millions and billions of data every day.

3.3.3 Experiment Environment for Processing Engine Selection

In this experiment, the cluster consists of three compute nodes (VMs). Table 3.3 shows the experimental parameter specification and the descriptions of datasets used in this experiment are presented in Table 3.4.

Table 3.3. Parameters Specification

Parameters	Specification
OS	Ubuntu 16.04 Linux
Host Specification	Intel ® Core™ i7-6500U CPU @ 2.50GHz, 8GB Memory, 1000GB Hard Disk
VMs Specification	1GB RAM, 150 GB Hard Disk
Software Component	Hadoop 2.6.0 Apache Spark 1.5.2 Mahout 0.9 Mahout Samsara 0.12.3

Table 3.4 Experimental Datasets for Processing Engine Selection

Dataset	Description
D1	Enron Data set [87] -Number of email (ham) : 52340 -Number of email (spam) : 30659
D2	Movie Reviews Data set [64] -Each record represents a movie available on Rotten Tomatoes -the scraping, movie tile, description, genres, duration, director, actors, users' ratings, and critics' ratings
D3	Twitter Data set [88] -Sentiment data for product reviews -contain 554470 positive and 494105 negative reviews

3.3.3.1 Performance Evaluation and Result Discussion

This subsection describes the performance evaluation of the two processing engines. To build a predictive model of the system, the preprocessed data set is divided into 60/40 ratios as training and test data sets. Figure 3.9 shows a comparison of the accuracy of each data set. The results clearly show that Spark's SNB can outpace large amounts of data with minimal processing time than MapReduce's SNB. In dataset D1, Spark's SNB can correctly classify 82999 emails within 39 seconds, while MapReduce's SNB took 119 seconds. In dataset D2, Spark's SNB processed movie review data within 40 seconds, and MapReduce's SNB took 150 seconds. In dataset D3, Spark's SNB took only 93 seconds to process tweet data, while MapReduce's SNB took 280 seconds. The benefits of SNB on Spark are maintained even for large datasets. Because it takes full advantage of Spark's in-memory computing and other great native features. In addition, it uses Samsara's text Vectorization pipeline which is slightly different TF and IDF transformations of MapReduce's SNB. Therefore, the sum of label extraction and TF_IDF observation for each label requires shuffle, and it takes several times to classify big data.

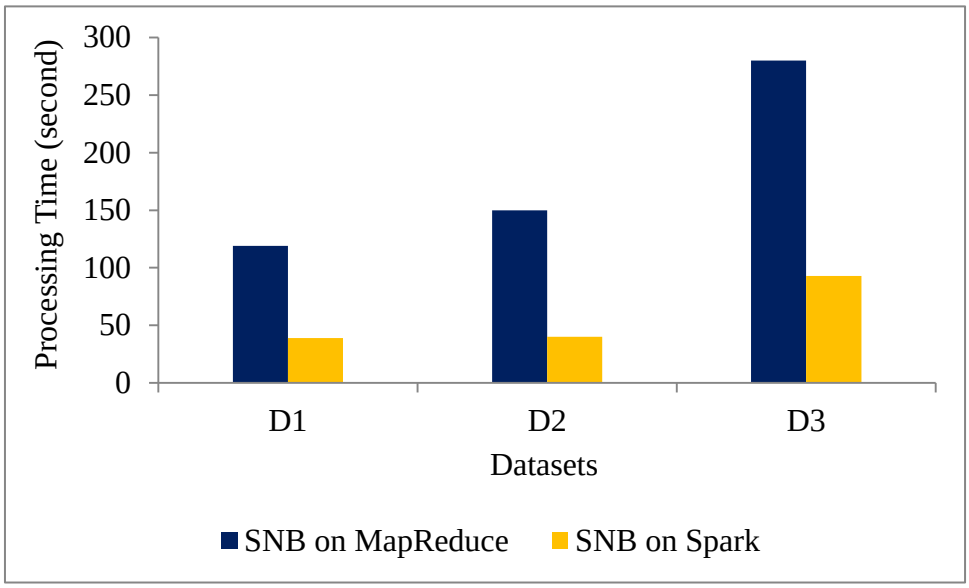


Figure 3.9 SNB on (MapReduce vs. Spark)

According to the comparison results, Spark's SNB takes less processing time than MapReduce's SNB for various experimental datasets. This work rarely solved the scalability problem using the actual scaling of machine learning. Figure 5 shows a comparison of SNB accuracy between MapReduce and Spark. SNB means having a

learning algorithm that can process any amount of data without consuming ever-increasing resources such as memory and providing accurate results.

Table 3.5 SNB on MapReduce vs. SNB on Spark (accuracy)

Dataset	MapReduce	Spark
D1	78.2936	81.2424
D2	80.4300	82.5600
D3	93.3300	98.0500

The explosive growth of computer technology has speeded up computation. While there is little difference between the two in accuracy, Spark's SNB offers higher efficiency not only in processing speed but also in accuracy. This is because Apache Spark is a highly developed engine that runs thousands of computing engines in parallel for huge volume of data processing. It maximizes the processor's control over those engines of calculation. Spark has the ability to handle multiple data processing functions with massive data such as terabytes and zettabytes, such as predictive analytics and scalable machine learning. Spark has become an ever-evolving engine in the field of fast big data processing with the concept of in-memory computing using RDD abstraction, DAG computing model, resource allocation and cluster manager schedule. Spark processing engine is chosen for the development of the proposed PBA system according to the analysis.

3.4 Machine Learning Algorithm Selection

In the era of big data, PA is still ubiquitous and is recognized as one of the major challenges of modern society. The majority of academic research [40] has attempted to identify and explain potential causes and consequences of breakage at various levels of granularity, mainly through theoretical lenses, using correlation and regression-based analysis. Machine learning (ML) is a computational method for automatic learning from experience and improves the performance to make more accurate predictions. In this subsection, the prediction scores of ML regression techniques are compared from a PA perspective and find the most important failure perception predictors based on linear and nonlinear models with a high level of

prediction accuracy. This comparative study seeks out a fair judgment and knows which algorithm is best to fit in with the experimental dataset.

The following study case is based on four regression algorithms: Random Forest, Decision Tree, Linear Regression and Gradient Boosting Tree. These algorithms have been chosen from Spark ML library, MLlib, to cover the most common data analytics tasks. The primary purpose of this study is to improve the quality of prediction process which has a notable influence on the efficiency of PA system, depending on the size of the dataset to be processed. The brief descriptions of four different ML techniques are as follow:

1. **Decision tree** is a decision support tool that uses a tree-like decision model and its potential effects, including the implications of chance events, resource costs and utility. It is one way of viewing an algorithm which contains statements of conditional influence only. The decision tree is a valuable algorithm for predictive modeling, which can be used to describe decisions functionally and obviously.
2. **Random forest** is an ensemble method that can be used in predictive analytics. It requires an ensemble of decision trees to construct their model. The idea is to take a random sample of data from the training and make them vote to choose the best and strongest model [44].
3. **Linear regression** is a statistical approach that analyzes the relations between two variables and identifies them. It can be applied in predictive analytics to predict a potential numerical value of a variable. For discrete data linear regression is ideally suited. But the data points are very sensitive towards outliers. The outliers in training data will affect the model significantly.
4. **Gradient boosting** is a predictive ML technique that enables a poor learner to adapt the differential recursively with a progressively growing number of iterations to improve model efficiency. And in the sense of hundreds, thousands or tens of thousands of possible predictors, it will automatically discover complex data structure, including non-linearity and high-order interaction.

The comparative performance of three different predictors is shown in Table 3.6. Key Performance Indicators (i.e., MAE, RMSE and MSE) are used to evaluate

the performance of four different predictors. The KDD dataset is used for testing the predicted scores of predictors.

Table 3.6 Comparative Results of Four Different Predictors

ML Predictor	MAE	RMSE	MSE
Random Forest	0.05778	0.00339	0.01046
Linear Regression	0.39652	0.15720	0.38889
Decision Tree	0.12082	0.01459	0.02671
Gradient Boosting	0.08231	0.00677	0.15191

Specifically, RF regression is found to be the most accurate predictive model within the predictive modeling setup employed here, followed by gradient boost tree, decision tree, and linear regression. As a result, RF has been selected the approximately best predictor with lowest error rate (MAE= 0.05778, RMSE= 0.00339 and MSE 0.01046). Due to the Linear Regression is sensitive to outliers and it considers only the features individually, the prediction accuracy is decreased by comparing with the other methods. Decision Tree has obtained the fairly obvious prediction powers on experimental dataset but it may cause the over fitting problem. Even though Gradient Boosting has been explored as the moderately high predictor, it can be harder to fit its parameters than RF on distributed big data environment. According to results, the selected ML algorithm (RF) is used to develop the effective PBA system.

3.5 Chapter Summary

The role of processing engine and ML techniques for PA system can be studied in this chapter. Firstly, the scalability test of PA system is conducted on traditional and big data analytics platform. Second, processing engines are compared and analyzed for parallel computing the big data. The evaluation results show that Spark offers the efficient processing performance than MapReduce. Moreover, the experiment has presented that Spark can perform operations on disk as well as memory, but MapReduce only performs processing on disk. Third, an effective ML

technique (RF) is selected to construct the accurate prediction model. For developing the proposed PBA system, Apache Spark is used as a parallel processing engine and scalable RF is applied as the predictive ML predictor.

CHAPTER 4

EFFICIENT PREDICTIVE HIGH DIMENSIONAL DATA ANALYTICS

Predictive Analytics (PA) is an emerging concept in the flood of big data. The emergence of big data brings an opportunity to answer many unresolved scientific questions, but there are some interesting challenges. The main features of modern datasets are large sample size and high dimensionality with several features. The high dimensionality of datasets poses challenges such as large search space for PA systems and reduced potential generalization. The curse of dimensionality brings the problem that as the number of features grows, the error grows as well. Dimension reduction techniques are more difficult to build and often have a running duration that is proportional to the dimensions. Data objects are typically converted to vectors in a lower dimensional space using feature extraction and selection [86] approaches to overcome the curse of dimensionality. In many cases, a small sample size for high-dimensional problems [91] can make the model more suitable for training data, resulting in poor generalization and ultimately poor predictor performance. Therefore, before further processing, it is necessary to know an effective reduction strategy that is compatible with the real-world data nature. This research brings forward an efficient big data analytics system by comparing various analytics techniques for processing large scale data with high dimension.

4.1 Implementation of Proposed System Architecture

Big data technology allows the systematic analysis in high-dimensional data sets to reveal hidden patterns. Furthermore, predictive concepts take a glance at risk factors and opportunities for making effective decisions based on user demand. It makes it easy to build and use PBA systems in a range of applications, such as financial services, stock market analysis [12, 14], capacity planning, fraud detection, and healthcare systems. Apart from successful well-being, the age of big data brings difficulties and problems. Excessive dimensions of the data impact machine learning performance in PBA systems. Existing learning algorithms cannot fully effectively manage high-dimensional Big Data.

The aim is to design and implement the proposed system as it promises to provide better insights from huge and heterogeneous data. This research work performs to optimize the hyperparameters of SRF to deliver the highest accuracy and minimal computing time of the proposed system. An overview of the proposed system that leverages predictive analytics for high dimensional data is depicted in Figure 4.1. The proposed system consists of three components, namely data analytics, data processing and data storage.

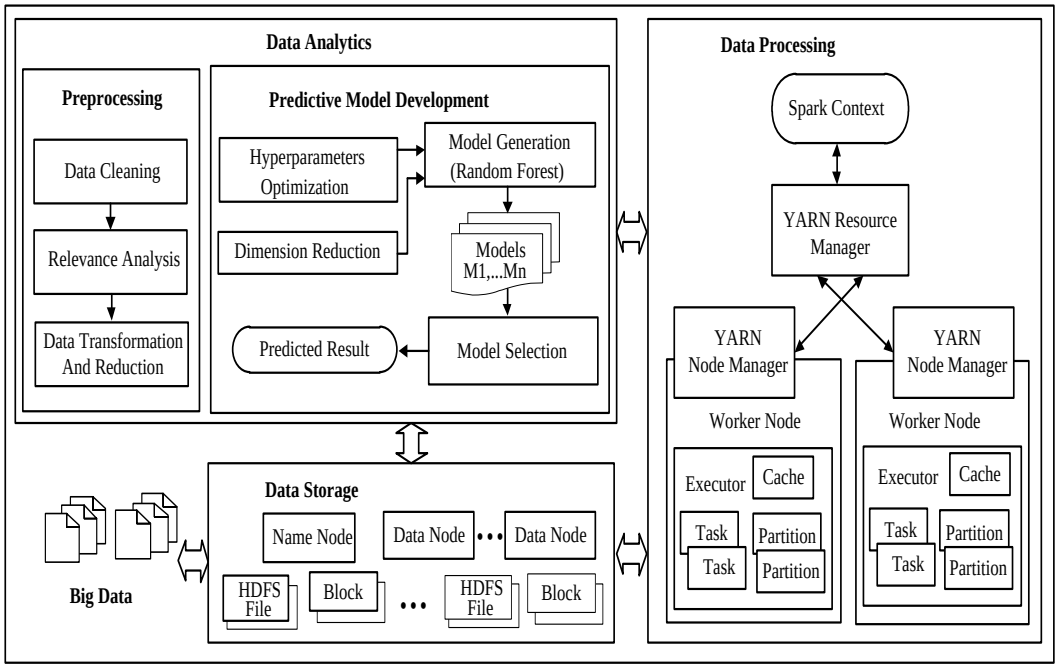


Figure 4.1 Architecture of Proposed PBA System

In data storage component, HDFS is applied to handle the large datasets and to store heavy file sizes which can range from gigabytes to terabytes. Hadoop clusters run on commodity hardware clusters. HDFS stores large quantitative data amounts and provides high-throughput access to application data. The HDFS architecture is similar to the master-slave architecture. It contains a name node and a data node which can be installed on the same cluster or different clusters. A data node with a name node that becomes a single cluster node (the node contains one machine) and 100 machines (the data maintains multiple machines).

In data processing component**,** Apache Spark is applied as a fast in-memory processing engine for computing large datasets from HDFS. Spark stores data objects in a main abstraction called Resilient Distributed Dataset (RDD). RDD provides an

interface for data conversion and parallelization. These RDDs are distributed across the cluster nodes. Two types of operations are possible with RDDS in Spark. The first type is a transformation that converts an RDD to another RDD. The second type is an action, which works with RDD to produce the final result that is sent to the master or saved to memory or disk. Transformation is performed with a delay. That is, no transformation is performed unless an action is called. This allows creating a transformation of job DAG before invoking the action and performing the optimizations on this DAG before execution. The RDD interface also provides caching of data in memory and therefore, it can be processed in the next iteration of the job without I/O latency.

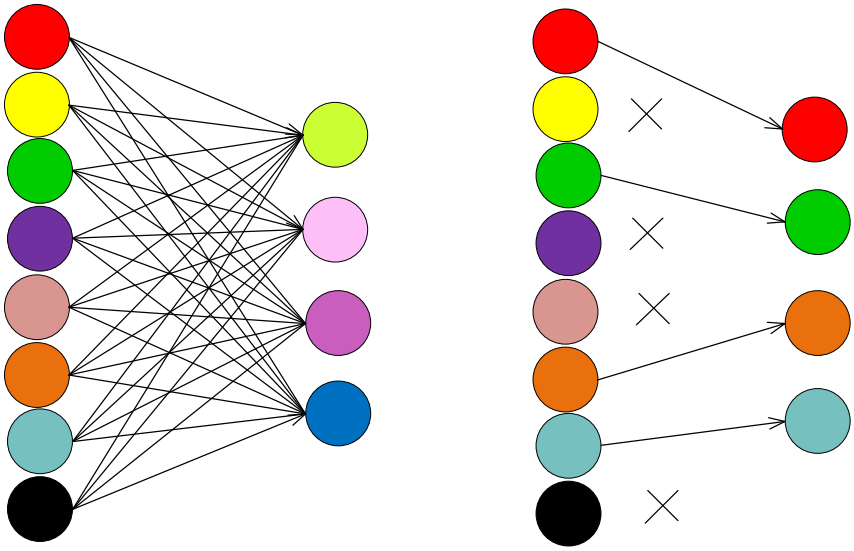
This involves two phases in the data analytics component: data pre-processing and the predictive model development. Pre-processing of data is a critical process for improved predictive accuracy of the proposed system. Big data usually includes unreliable, repetitive data that cannot be used directly for predictive analysis, whereas pre-processing steps are performed to increase predictive process accuracy and performance. Firstly, the data cleaning step is accomplished by applying the smoothing technique to eliminate or reduce noise. It enables the removal of outliers and the resolution of conflicts with the handling of missing value. Second, steps of data transformation and reduction are done using standard techniques of normalization. That provides a more predictable pattern of forecasting. Second, relevance analysis is accomplished to quantify the relevance of an attribute with respect to a given class or concept. Otherwise, the learning procedure may be slow and misleading to include those attributes. This analysis can assist to increase the effectiveness of predictive measures. SRF is utilized to offer correct decision-making process during the prediction model development phase. The Prediction Model is firstly established based on the observations over the experimental prediction results of the historical data by applying the SRF from Spark MLlib. During this process, dimension reduction and hyperparameters optimization are performed. This generalized prediction model is evaluated with the accuracy measurement in terms of error matrices, MAE, RMSE and MSE. Then the best model is selected to guess the actual predicted results.

4.2 Classical Methods for High Dimensionality Reduction

The wave of new technologies has opened up the opportunity to cost-effectively generate high-throughput profiles for PBA systems. The marriage between "dimensionality reduction" and "big data" comes with the huge opportunities as well as challenges to these communities. It is thus leading us towards the "big data" era which is creating a pressing need to bridge the gap between high-throughput technological development and the ability for managing, analyzing, and integrating the big data. To harness the maximum out of it, sufficient expertise needs to be developed for managing and analyzing big data.

The "Curse of Dimensionality" becomes difficulty in finding any observable or comprehensive patterns to fit the prediction models as the numb er of feature variables in the data space increases [41, 42]. When the number of dimensions increases, the number of features often increases, sometimes much faster, means that certain high-dimensional features are correlated with tremendous sparseness. Additionally, there might be some connection between different dimensions, and thus features may be hard to describe. A number of linear and non-linear dimensionality reduction techniques [77] have been developed to combat the dimensional curse. These methods are intended to reduce the number of dimensions (variables) in a dataset by selecting features or extracting features without losing significant information. To address the high dimensionality problem of PBA systems, dimension reduction methods based on various feature selection approaches have been proposed [38, 39].

Recent research has combined several dimension reduction strategies (feature extraction [68] and feature selection [14]) with ML and high-speed data processing processes. Feature extraction methods reduce the feature space with consideration by a set of principal features. In fact, combining the existing features reduces the number of extracted features, so the new potential features contain the feature contained in the original observation. In the feature selection method, some of the more important features are retained and the remaining features are deleted without any operation. It would also facilitate a deeper understanding of complex and high dimensional models. Figure 4.2 shows the two-dimension reduction techniques for big data.



(a) Feature Extraction (b) Feature Selection

Figure 4.2 Dimension Reduction Techniques (a) Feature Extraction and (b) Feature Selection

4.2.1 DR_PCA based Feature Extraction Technique

PCA is a feature exaction technique for dimension reduction that can be used to decrease the large set of variables to a small set that still contains information in the large set [17]. PCA's primary purpose is to represent the multidimensional data with fewer variables by preserving the data's main features. Reducing a data set's number of variables inevitably comes at the cost of accuracy, but the strategy in reducing dimensionality is to compensate for simplicity with a little reliability. Even though fewer data sets are simpler for machine learning algorithms to explore and visualize, which make analyzing data significantly simpler which faster without processing extraneous variables. PCA transforms the original n dimensional data into the k-dimension of the data ($k < n$) by creating the new uncorrelated variables that successively maximize variance, called principal components. The first principal component provides for as much variation as possible in the data and as much of the remaining uncertainty as possible is accounted for through each successive variable. Figure 4.3 show the procedure of dimension reduction with PCA.

Procedure: Dimension Reduction with PCA (DR_PCA)
Input: S, Spark dataframe k, the number of required PCs m, list of numeric data features in S Output: F, the set of k component with updated Spark dataframe Method: 1. Standardize the Spark dataframe S 2. Calculate the mean-centered matrix $\mathbf{M_c}$ 3. Calculate the covariance matrix $\mathbf{X_c} \leftarrow \mathbf{M_c}' \times \mathbf{M_c}$ 4. Calculate the Eigen value decomposing of covariance matrix $\mathbf{X_c}$ 5. Get the Eigenvectors and Eigenvalues from decomposition of $\mathbf{X_c}$ 6. Sort Eigenvalues in descending order 7. Choose the Eigenvector which has largest Eigenvalues 8. Construct the matrix by projecting the selected Eigenvectors 9. Obtain the k principal components of $\mathbf{X}$: A= $(a_1, a_2, \ldots, a_d)$ 10. Return F

Figure 4.3 Procedure of Dimension Reduction with PCA (DR_PCA)

The method of lowering the number of variables under consideration is referred as DR_PCA in this research work. It can be used to extract latent features from noisy and raw data, as well as compress data while keeping the structure. On the RowMatrix class, it adds functionality for dimensionality reduction. It gets Spark DF as input, along with the subset of numerical features to be reduced and the targeted number of features to be produced for the PCs. After attaching PCs to it, the function returns the same Data frame input along with a list of variances of generated PCs.

4.2.2 DR_IG based Feature Selection Technique

Information Gain (IG) has emerged as crucial feature selection approach in applications where the large dimension space of input data affects the prediction algorithm. The basic concept of this method is to find the data subsets of features by

measuring the value of the Gain Ratio for each feature variable. IG values of the variables are obtained with information entropy by qualifying the uncertainty of predicting the target variables. Specifically, these techniques improve the predictive machine learning process from the data by choosing the most important features from the noisy, unnecessary and irrelevant features. The procedure for DR_IG based feature selection technique is shown in Figure 4.4. The information entropy of each variables is defined according to the following formulas:

$$Entropy\ (S_i) = -\ \sum_{c=1}^{d} q_c \times log q_c \tag{4.1}$$

$$Entropy\ \left(y_{ij}\right) = \sum_{v \in V\ (y_{ij})}^{d} \frac{|S_{(v,i)}|}{|S_i|}\ Entropy\ (v(y_{ij}\ )) \tag{4.2}$$

$$I\left(y_{ij}\right) = \sum_{c=1}^{m} -q_{c,j} \times \log(q_{c,j}) \tag{4.3}$$

$$G\ \left(y_{ij}\right) = Entropy\ (S_i) -\ Entropy\ \left(y_{ij}\right)$$

$$= Entropy\ (S_i) - \sum_{v \in V\ (y_{ij})}^{d} \frac{|S_{(v,i)}|}{|S_i|}\ Entropy\ (v(y_{ij}\ )) \tag{4.4}$$

DR_IG is used for feature reduction, by evaluating the gain of each variable in the context of the target variable. Improved information gain values mean better accuracy for decision making at SRF. It is affective in removing irrelevant data, increasing learning accuracy, and improving result comprehensibility.

Procedure: Dimension Reduction with IG (DR_IG)

Input:

S, the training dataset

k, the number of selected k features

m, the input number of feature variables in S

Output:

F, the set of m important feature variables of S

Method:

1. Standardize the training subset S_i
2. Calculate the entropy of target feature variables,
3. **for** each variables y_{ij} in training dataset S
4. Calculate entropy of y_{ij} for each variable,
5. Calculate splitting node information of y_{ij} for each variable,
6. Calculate information gain value for each variable y_{ij},
7. Calculate gain ratio value for each variable y_{ij},
8. End **for**
9. Calculate the feature importance y_{ij} for each variable,
10. Sort the Fe-imp(y_{ij}) value in descending order
11. Choose top k features to F
12. Return F

Figure 4.4 Procedure of Dimension Reduction with IG (DR_IG)

4.3 Prediction Model Construction of SRF

The main building process for prediction models of research work is presented in this section. It composes of two functions; (i) Dimension Reduction: dimension reduction process brings the advantages of minimizing the data analytics time; and reducing the storage spaces. Moreover, this process ensures the simplicity and improves the efficiency of decision making. Irrelevant or partially relevant features can poorly impact the quality of model performance. The influence rates of all features are not identical (the same) and the consideration of all features offers consequently overheads in model development and prediction. Therefore, feature (dimension) reduction is performed by eliminating features with lowest influence for the model. (ii) Parameter Optimization: the model fitness and prediction accuracy

depends on parameter values of RF and thereby this research contributes Parameter Optimization for the split point parameter of RF to get the predicted resources accurately.

This research work aims to develop a big data and predictive analytics system that can ensure the efficient prediction model construction in PBA system implementation. The Enhanced SRF (ESRF) model construction process is provided in Figure 4.5.

Procedure: Model Construction of ESRF

Input :

S : Input Instances

P_{nt} :Parameter(number of trees)

P_{md} : Parameter(maximal depth)

Output:

F : Error of Models $\{M_1, M_2, \ldots, M_{nt}\}$

Method:

1. Preprocess the data from S
2. Create the Fi, Fi { } empty string array
3. Divide the preprocessed data into the S_{tr} training dataset and check the S_{test} dataset
4. Eliminate irrelevant S_{tr} feature
5. Build S_{tr} bootstrap samples $\{S_{tr(1)}, S_{tr(2)}, \ldots, S_{tr(nt)}\}$
6. Determine the set of values for hyperparameters
7. **for** nt=1 to nt_{max} **do**
8. **for** md=1 to md_{max} **do**
9. Produce the RF model with combination of hyperparameters (nt, md)
10. Fit the model on S_{tr}
11. Predict the result based on M features
12. Predict the regression evaluator values
13. Put the error value to F_i, F_i {0,1,2,…,| nt_{max} | × | md_{max} |}
14. End **for**
15. End **for**
16. Sort error values of F_i in ascending order
17. Return F_i

Figure 4.5 Procedure of the Model Construction of ESRF

4.4 Experimental Evaluation

In this research work, the efficiency and effectiveness of the proposed system is evaluated via extensive experiments on various real-world data sets and discuss the implications of the findings. The software and hardware component specifications are shown in Table 4.1.

Table 4.1 Testing System Specification of PBA System

Parameters	Specification
OS	Ubuntu 16.04 Linux
Host Specification	Intel ® Core™ i7-6500U CPU @ 2.50GHz, 8GB Memory, 1000GB Hard Disk
VMs Specification	1GB RAM, 150 GB Hard Disk
Software Component	Hadoop 2.7.1 Apache Spark 2.4.4

4.4.1 Experiment Dataset

All together 5 data sets are selected from Parallel Workload Archives (PWA) and UCI Machine Learning Repository [2]. A summary of data sets is presented in Table 4.2.

Table 4.2. Summary of Experimental Datasets.

No	Datasets	#Repository	#Instances
1	DAS2-fs0-2003-1 (DAS)	PWA	225,711
2	High-Performance Computing Center North (HPC2N)	PWA	527,371
3	Credit-card	UCI	284,808
4	KDD-cup 1999 (KDD)	UCI	400,000
5	Susy	UCI	500,000

There are enormous collections of workload datasets of a variety of High Performance Computing especially for predictive analytics. In this work, two workload traces: DAS2-fs0-2003-1 (Distributed ASCI in Netherlands Supercomputer-2) and High-Performance Computing Center North (HPC2N) obtained from Parallel Workload Archives [18] are examined. Each workload dataset contains several thousands of jobs. The number of features containing in DAS2-fs0-2003-1 dataset referred as DAS in this work are 17 features and it is listed in Table 4.3. Moreover, dataset referred as High-Performance Computing Center North workload trace, HPC2N in this work has the same features.

Table 4.3 DAS Workload Dataset

Number	Features of DAS Workload Dataset
1	Job Number
2	Submit Time
3	Wait Time
4	Run Time
5	Number of Allocated Processors
6	Average CPU Time Used
7	Used Memory
8	Requested Number of Processors
9	Requested Time
10	Requested Memory
11	Status
12	User ID
13	Group ID
14	Executable (Application) Number
15	Queue Number
16	Partition Number
17	Preceding Job Number
18	Think Tim Form Preceding Job

Parallel Workload Archives Logs, DAS and HPC2N hold some noisy data in some features called irrelevant fields with a value of -1. The workload preprocessing step is conducted to eliminate these features and the ongoing prediction development and evaluation process take into account only on the purified features, specifically 12 features for DAS workload and 13 features for HPC workload. Feature **Requested Number of Processors** is the predicted variable in this works.

The credit-card dataset contains the transactions made by credit cards in September 2013 by European cardholders. It contains 28 features [v1, v2… v28] which are only numerical variables. Due to confidentiality issues, there are not provided the original features and more background information about the data. To identify fraudulent credit card transaction, feature **class** is used as the predicted variable.

The KDD-cup 1999 dataset contain many fraud detection records. The number of features containing in The KDD-cup 1999 dataset referred as KDD are 39 types of features and it is listed in Table 4.4.

Table 4.4 KDD Dataset

Number	Features of KDD Dataset
1	duration
2	protocol-type
3	src_byte
4	dst_bytes
5	land
6	wrong_fragment
7	urgent
8	hot
9	num_failed_logins
10	logged_in
11	num_compromised
12	root_shell
13	su_attempted

14	num_root
15	num_file_creations
16	num_shells
17	num_access_files
18	num_outbound_cmds
19	is_host_login
20	is_guest_login
21	count
22	srv_count
23	serror_rate
24	srv_serror_rate
25	rerror_rate
26	srv_rerror_rate
27	same_srv_rate
28	diff_srv_rate
29	srv_diff_host_rate
30	dst_host_count
31	dst_host_srv_count
32	dst_host_same_srv_rate
33	dst_host_diff_srv_rate
34	dst_host_srv_diff_host_rate
35	dst_host_serror_rate
36	dst_host_srv_serror_rate
37	dst_host_same_src_port_rate
38	dst_host_rerror_rate
39	dst_host_srv_rerror_rate

All the types of attack classes are basically categorized into four main types: DoS- Denial of service, Probing- Surveillance and other probing attacks, U2R-

Unauthorized access to local super user, R2L- Unauthorized access from a remote machine. Those are assigned to attack or normal type.

Susy dataset has been generated using Monte Carlo simulations. It contains the number of 18 features to discriminate between a signal process which produces super symmetric particles and a background process which does not. The features of Susy dataset are listed in Table 4.5.

Table 4.5 Susy Dataset

Number	Features of Susy Dataset
1	lepton 1 pT
2	lepton 1 eta
3	lepton 1 phi
4	lepton 2 pT
5	lepton 2 eta
6	lepton 2 phi
7	missing energy magnitude
8	missing energy phi
9	MET_rel
10	axial MET
11	M_R
12	M_TR_2
13	R
14	MT2
15	S_R
16	M_Delta_R
17	dPhi_r_b
18	cos(theta_r1)

4.4.2 Predictors Measurement Metrics

A reliable estimate of the accuracy of the predictor is measured in terms of errors. Let DT be (x_1, y_1), (x_2, y_2), ..., (x_n, y_n). Where x_i is the n-dimensional test tuple

related to the known value y_i of the response variable y, and n is the number of tuples in DT. The accuracy of the predictor is estimated by calculating the error based on the difference between the predicted value of each test tuple X and the actual known value of y. The loss function measures the error between the actual value y_i and the predicted value. The most common loss functions can be performed using Equations 4.5, 4.6 and 4.7. MAE, MSE and RMSE are used to calculate the predictive accuracy of the system as measurement metrics.

$$Mean\ Absolute\ Error\ (MAE) = \frac{1}{n}\sum_{i=1}^{n}(y_i - y_i') \tag{4.5}$$

$$Mean\ Squared\ Error\ (RMSE) = \frac{1}{n}\sum_{i=1}^{n}(y_i - y_i')^2 \tag{4.6}$$

$$Root\ Mean\ Squared\ Error\ (RMSE) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - y_i')^2} \tag{4.7}$$

RMSE exaggerates the existence of outliers, while MAE does not. RMSE can be achieved by taking the square root of the mean square error. This is useful because it allows the measured error to be as large as expected. MAE and RMSE vary from 0 to infinity.

4.4.3 Optimization of hyperparameters for SRF

Hyperparameter values impact the predictive model's accuracy [3], and choosing the best hyperparameters values combination plays a key role in constructing successful predictors. This section portrays optimal hyperparameter observations: number of trees and maximum tree depth. For each dataset, the numbers of 63488 prediction models are developed and the experimental results are analyzed. A more technical case is elaborated in Figure 4.6. The comparative study shows that the same error value is observed in a forest with 128 and 2048 trees. The error rate cannot be significantly altered for more than 128 trees. Due to the big data dimension curse, deeper trees have influenced long decision-making response times. Additionally, the tree size rose the RF computational time. The output of each model produced was therefore analyzed. The best predictive model with the minimum error rate and the lowest calculation time was chosen for the respective dataset.

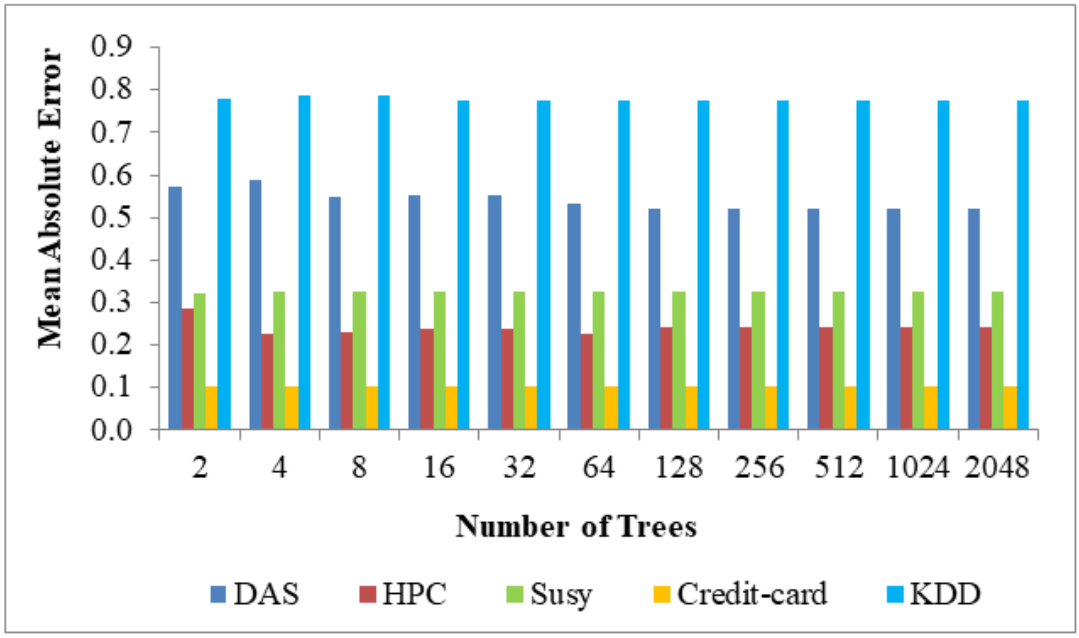


Figure 4.6 MAE Comparison of Each Prediction Model

The accuracy of the models in each data set is quite different based on content of the training data. SRF needs the best optimal parameters to construct the model, and predictive performance relies on the settings of the parameter. The default parameters settings for all datasets are fair but not sufficient. Thus, optimizing SRF's hyperparameters is an essential step in model configuration. A more technical case is presented in Table 4.6 with a comparison of default and optimized hyperparameters.

Table 4.6 MAE comparison of default and optimized hyperparameters

No	Datasets	MAE (Default Parameter)	MAE (Optimized Parameter)
1	DAS	1.4182	0.5191
2	HPC2N	1.5476	0.242
3	Susy	0.335	0.325
4	Credit-card	0.1018	0.0017
5	KDD	0.778	0.7753

The experiment results demonstrate that performing optimization of the hyperparameters gives greater accuracy than the parameter settings set by default. In addition to parameter selection, the choice of input samples also affects the accuracy

of the final model. By properly selecting the input parameter, the higher limit of model accuracy is obtained.

4.4.4 Performance Evaluation

In this study, feature reduction techniques were applied to reduce the inefficiency and redundant features of the experimental data set. This section presented the comparison performance of the proposed models described in Section 4.2. The DR_PCA and DR_IG algorithm-based dimensional reduction techniques were used on the preprocessed data and an ESRF model with the best hyperparameters was developed.

4.4.4.1 DR_PCA based ESRF Results

ESRF analysis using DR_PCA is carried out using actual experimental data sets. The best PCs are considered effective in selecting the element that has the greatest impact on the score of the prediction. The number of features included in traces of DAS workload is 13, as shown in Table 4.1. Each ESRF predictive model was developed using 2 to 13 PCs to find the best PC for this dataset. Table 4.7 shows the three error comparison values of ESRF using DR_PCA on DAS workload traces.

Table 4.7 Performance Measurement Matrices for DAS Dataset

Number of Principal Components	ESRF with DR_PCA		
	RMSE	MSE	MAE
2	8.82429	77.86818	3.1392
3	5.12333	26.24850	2.04689
4	1.24364	1.546662	0.38685
5	1.11158	1.23561	0.35151
6	1.16235	1.35106	0.39417
7	1.06698	1.13845	0.31649
8	1.05903	1.12155	0.32183
9	1.05802	1.11942	0.32545
10	0.95125	0. 90487	0.27674
11	0.98687	0.973922	0.30183
12	0.95917	0.95093	0.35295
13	0.97463	0.83654	0.27681

The principal components were created using DR_PCA on the basis of the training dataset only to avoid bias, which ensures that the data from the testing dataset is not released into the training dataset. If the entire dataset is used to measure the principal components, the model will not work as well when actual new unknown data is fed to the model. Similarly, if the PCAs are measured separately on both sets, two incompatible data sets will be generated. The system cannot train a predictor in one space and adapt it to a different space. Thus, the same statistics of the training samples were used to transfer the testing dataset using the DR_PCA into the same feature space. The minimum error values used by DR_PCA for the DAS data set are 0.95125, 0.90487 and 0.27674. The largest error values are obtained (PCs =2) with RMSE 8.82429, MSE 77.86818 and MAE 3.1392. As a result, a model with ten principal components is considered more efficient and therefore, this model is the best.

Table 4.8 Performance Measurement Matrices for HPC2N Dataset

Number of Principal Components	ESRF with DR_PCA		
	RMSE	MSE	MAE
2	0.49634	0.24636	0.19508
3	0.60941	0.37138	0.27851
4	0.07746	0.00555	0.02543
5	0.12477	0.01556	0.03730
6	0.20393	0.04158	0.06955
7	0.08203	0.00672	0.01418
8	0.11078	0.01227	0.02372
9	0.11956	0.01430	0.02753
10	0.07255	0.00526	0.01392
11	0.10371	0.01075	0.02458
12	0.12309	0.01515	0.02990
13	0.07981	0.00637	0.01571

A comparative study of various principal components and their RMSE, MSE and MAE values is shown in Table 4.8. The prediction model with ten principal component obtained the maximum RMSE 0.60941, MSE 0.37138 and MAE 0.27851.

The prediction model with 4 PCs, 7 PCs and 13 PCs reaches approximately low error rates, but the prediction model with 10 PCs has a higher level of prediction for the HPC2N dataset (RMSE = 0.07255, MSE 0.00526, MAE 0.01392).

Table 4.9 displays the ESRF error values in the Susy dataset, using DR_PCA. For MAE and RMSE the maximum error rate is 0.47582 and 0.48799. The error value increases slightly around two and four principal components. The research values fluctuates approximately 0.1 from 6 PCs to 15 PCs. According to this experimental result, the prediction model with 10 PC has a relatively low error values and this model is considered the best.

Table 4.9 Performance Measurement Matrices for Susy Dataset

Number of Principal Components	ESRF with DR_PCA		
	RMSE	MSE	MAE
2	0.44359	0.196777	0.39
3	0.41983	0.17626	0.36301
4	0.4139	0.17131	0.34776
5	0.40438	0.16352	0.33701
6	0.40438	0.16352	0.34004
7	0.40358	0.16288	0.33626
8	0.39854	0.15883	0.33333
9	0.39707	0.15767	0.33436
10	0.38774	0.15035	0.31763
11	0.38791	0.15047	0.31854
12	0.38804	0.14908	0.32049
13	0.38611	0.14908	0.31594
14	0.38526	0.14843	0.31614
15	0.85543	0.24856	0.32718

The detail of the increase in number of PCs and different performance matrices for KDD dataset are shown in Table 4.10. The experiment was performed several times to conduct the ESRF predictor's performance, in which the PBA system start increasing the number of PCs for each predictor gradually. As shown in Table 4.6, the

performance of the models did not increase after the 26 PCs. The DR_PCA technique based prediction model achieves the lowest error rates (RMSE 0.05913, MSE 0.00349, and MAE 0.01163) with 25 PCs and it outperforms all the other individual predictors.

Table 4.10 Performance Measurement Matrices for KDD Dataset

Number of Principal Components	**ESRF with DR_PCA**		
	RMSE	MSE	MAE
2	0.16749	0.02805	0.06498
3	0.13726	0.01884	0.05152
4	0.1018	0.01036	0.02375
5	0.09797	0.00959	0.02352
6	0.09224	0.00851	0.02319
7	0.08817	0.00777	0.02045
8	0.08899	0.00792	0.02094
9	0.08831	0.00779	0.02121
10	0.07643	0.00584	0.01699
11	0.07142	0.0051	0.01559
12	0.068829	0.00473	0.0151
13	0.06505	0.00423	0.01297
14	0.06417	0.00411	0.01332
15	0.06351	0.00403	0.01302
16	0.06396	0.00409	0.01261
17	0.06172	0.00381	0.01261
18	0.06263	0.00392	0.01298
19	0.06062	0.00367	0.01208
20	0.06003	0.0036	0.0121
21	0.06125	0.00375	0.01238
22	0.05985	0.00358	0.01209
23	0.06045	0.00365	0.01216
24	0.05954	0.00354	0.01226
25	0.05847	0.00341	0.01193

26	0.05956	0.00354	0.01201
27	0.05996	0.00347	0.01241
28	0.05872	0.00344	0.01220
29	0.06126	0.00375	0.01303
30	0.06027	0.00363	0.01291
31	0.06051	0.00366	0.012593
32	0.06028	0.00363	0.01267
33	0.05983	0.00357	0.01291
34	0.06037	0.00364	0.01309
35	0.061533	0.00378	0.01336
36	0.06076	0.00369	0.01316
37	0.06025	0.00363	0.01289
38	0.05976	0.00357	0.01268
39	0.05801	0.00436	0.02205

Table 4.11 represents the prediction performance of the ESRF predictor for Credit-card dataset in the context of MAE, RMSE and MAE. Several Prediction models are constructed to examine the predictor's performance, in which the numbers of PCs were increased from 2 to all features for each predictor gradually. Generally, the output of the prediction model is the number of variables in a data set, but the key is to sacrifice a very limited accuracy for efficiency in reducing dimensionality. Since it is simpler to explore and simulate smaller data sets and to make data analysis far easier and more convenient for ESRF predictor without processing extraneous variables.

As shown in Table 4.11, it can be observed that the performance of the ESRF's models did not improve after 22 PCs and it achieves the lowest error rates (RMSE 0.05884, MSE 0.00346 and MAE 0.01121) with 22 PCs. In contrast, the prediction model of ESRF obtained low error (RMSE 0.05913, MSE 0.00349 and MAE 0.01163) with 26 PCs and (RMSE 0.05888, MSE 0.00346 and MAE 0.01245) with 30 PCs. The ESRF predictor reduced the computational cost using DR_PCA technique and the prediction model with 22 PCs outperforms all the other individual predictors.

Table 4.11 Performance Measurement Matrices for Credit card Dataset

Number of Principal Components	ESRF with DR_PCA		
	RMSE	MSE	MAE
2	0.16523	0.02730	0.06345
3	0.13604	0.01850	0.049011
4	0.10066	0.01013	0.02330
5	0.09850	0.00970	0.02351
6	0.09354	0.00875	0.02281
7	0.09027	0.00815	0.02051
8	0.09052	0.00819	0.02126
9	0.08945	0.00800	0.02082
10	0.07870	0.00619	0.01757
11	0.07328	0.00537	0.01587
12	0.07126	0.00507	0.01547
13	0.06599	0.00435	0.01295
14	0.06571	0.00431	0.01311
15	0.06697	0.004485	0.01385
16	0.06581	0.00433	0.01271
17	0.06384	0.004076	0.01255
18	0.06182	0.00382	0.01236
19	0.06007	0.00361	0.0114
20	0.05906	0.00348	0.01145
21	0.05900	0.00348	0.01148
22	0.05884	0.00346	0.01121
23	0.05904	0.05904	0.01161
24	0.05913	0.00349	0.01194
25	0.05932	0.00351	0.01168
26	0.05913	0.00349	0.01163
27	0.05947	0.00353	0.01206
28	0.05861	0.00343	0.01209
29	0.05991	0.00358	0.012148

30	0.05888	0.00346	0.01245
31	0.06025	0.00363	0.01243
32	0.06052	0.00366	0.01252
33	0.05928	0.00351	0.01234
34	0.06018	0.003622	0.01247
35	0.06024	0.00362	0.01285
36	0.06075	0.00369	0.01296
37	0.05968	0.003562	0.01242
38	0.05914	0.00398	0.01227

As the number of dimensions increases, so does the number of features, which indicates that such high-dimensional features have a lot of heterogeneity. Furthermore, various dimensions seem to have some connectivity, making it difficult to identify characteristics. According to this performance analysis, DR_PCA consider the he features with large variance as important features. However, DR_PCA is sensitive to the scaling of the variables. For these data sets, using DR_PCA for dimensionality reduction will not significantly affect the accuracy of the prediction.

4.4.4.2 DR_IG based ESRF Results

This section outlines ESRF's performance analysis on five research data sets using DR_IG. Some research has been carried out on ESRF predictor using DR_IG as feature reduction for generating the prediction model. To produce more accurate predictions, the best prediction is chosen based on the lowest error rate of the model. Therefore, the most important features for each dataset have been carefully considered. The feature hierarchy was adopted using DR_IG. In the field of predictive analytics, IG is also deployed as a term-goodness standard. It is determined on the basis of a system's entropy, i.e., the feature ranking of the system. Therefore, to calculate IG, the entropy of a subset is a fundamental measure. Finally, the information gain is calculated for each variable based on the groups created for individual value of the variable and the measured entropy. Table 4.12 shows the performance matrices of ESRF using DR_IG for DAS workload traces. It can be seen that how the different number of features sets affects the performance of the ESRF. It is interesting to find that, when features set increases from 2 to 6, the prediction performance cannot improve significantly. The experimental results clearly show that

the performance of the prediction model cannot obtain high accuracy after 7 features and it is considered the best (RMSE 1.42204, MSE 2.02220 and MAE 0.30464) with 7 features. The model with seven features has relatively low error values for DAS workloads.

Table 4.12 Performance Measurement Matrices for DAS Dataset

Number of Features	DR_IG based ESRF		
	RMSE	MSE	MAE
2	3.34707	11.20284	1.36787
3	1.47452	10.87289	3.29741
4	2.19455	4.81603	0.56841
5	2.71331	4.94896	2.22462
6	2.18367	4.94896	0.71331
7	0.71331	2.02220	0.30464
8	2.00143	4.00573	0.42844
9	2.68277	7.19728	0.62181
10	1.87324	3.50905	0.43201
11	1.80849	3.27065	0.43535
12	2.01293	4.05188	0.57166

Table 4.13 shows the different performance metrics of ESRF in the HPC2N workload traces using DR_IG. DR_IG attempts to measure how much information there is in a random variable or, more precisely, the distribution of probability for HPC2N dataset. A distorted distribution has low entropy, whereas there is greater entropy in a distribution where occurrences have same probability. One interpretation of entropy from information theory of DR_IG technique is that it defines the minimum number of bits of information required to encode the prediction of the original HPC2N dataset as an arbitrary member of the feature subset. The ESRF prediction model's error value for the HPC2N dataset decreased from 13 features to 7 features, and with 8 features it increased steadily. Nonetheless, with six features, the error value increases again from four features to two features. As a result, the

prediction model with seven features has a relatively low error values and this model is considered the best.

Table 4.13 Performance Measurement Matrices for HPC2N Dataset

Number of Features	DR_IG based ESRF		
	RMSE	MSE	MAE
2	0.22191	0.04924	0.10884
3	0.38221	0.45496	0.67451
4	0.10109	0.01022	0.09889
5	0.07381	0.02076	0.07381
6	0.47764	0.22814	0.28551
7	0.03984	0.00519	0.09052
8	0.23139	0.05354	0.14471
9	0.19031	0.09907	0.31476
10	0.31295	0.09794	0.18477
11	0.36971	0.13669	0.22545
12	1.33636	1.78586	0.83294
13	0.37227	0.13859	0.230134

Table 4.14 shows the different performance metrics of ESRF in the Susy dataset using DR_IG. To produce more accurate predictions, the best prediction is chosen based on the lowest error rate of the model of Susy dataset. Therefore, it is necessary to take into account the best features of each dataset and analyze the results of the models that use different features. It is observed that the prediction performance of ESRF unchanged significantly when the features set increased from 2 to 13. The prediction model with 14 features has a relatively low error values (RMSE 0.27059, MSE 0.13734, and MAE 0.29020) and this model is considered the best.

Table 4.14 Performance Measure Matrices for Susy Dataset

Number of Features	ESRF with DR_IG		
	RMSE	MSE	MAE
2	0.39701	0.15762	0.31870
3	0.38972	0.15188	0.31233
4	0.38502	0.14824	0.30150
5	0.38049	0.14477	0.29796
6	0.38017	0.14453	0.29904
7	0.37741	0.14221	0.29450
8	0.37456	0.14029	0.29316
9	0.37667	0.14188	0.29518
10	0.37496	0.14059	0.29250
11	0.37507	0.14068	0.29316
12	0.37301	0.13914	0.29264
13	0.37186	0.13827	0.29030
14	0.27059	0.13734	0.29020
15	0.29123	0.13770	0.29123
16	0.37045	0.13724	0.29039

Table 4.15 depicts the different performance metrics of ESRF in the KDD dataset using DR_IG. One of the most significant shortcomings in the KDD data set is the huge number of replicated records, which causes the ESRF predictor to be biased towards the frequent records, and thereby preventing them from learning infrequent records that are typically more dangerous to networks such as cyber-attacks. Furthermore, the presence of these repetitive records in the test set would cause the results of the assessment to be skewed by the techniques on the frequent records that have higher detection rates. Therefore, the individual prediction model is trained according to the number of features included in the dataset. The model with the best results is selected. As shown in Table 4.11, it can be observed that the performance of the ESRF's models did not improve after 31 features and it outperforms all the other individual predictors with the best results (RMSE 0.03669, MSE 0.00135, and MAE 0.00735). According to these results, the prediction model with 14 features is considered the best.

Table 4.15 Performance Measure Matrices for KDD Dataset

Number of Features	DR_IG based ESRF		
	RMSE	MSE	MAE
2	0.22164	0.04912	0.11435
3	0.19444	0.03781	0.09812
4	0.16696	0.02788	0.06721
5	0.09959	0.00992	0.03273
6	0.10642	0.01132	0.03318
7	0.07669	0.00588	0.02094
8	0.08100	0.00656	0.02262
9	0.07388	0.00545	0.01802
10	0.05514	0.00304	0.01125
11	0.04878	0.00238	0.00850
12	0.06153	0.00388	0.01292
13	0.04565	0.00208	0.00821
14	0.04852	0.00235	0.00905
15	0.04768	0.00227	0.00946
16	0.04491	0.00202	0.00835
17	0.04874	0.00238	0.00892
18	0.04147	0.00172	0.00775
19	0.04005	0.00160	0.00720
20	0.04323	0.00187	0.00830
21	0.04265	0.00181	0.00817
22	0.03872	0.00149	0.00701
23	0.03965	0.00157	0.00701
24	0.04189	0.00175	0.00787
25	0.03811	0.00145	0.00683
26	0.03990	0.00156	0.00746
27	0.44230	0.00178	0.00820
28	0.04204	0.00176	0.00776
29	0.03838	0.00147	0.00778
30	0.04218	0.00177	0.04218

31	0.03669	0.00135	0.00735
32	0.03985	0.00159	0.00735
33	0.04116	0.00169	0.00788
34	0.03957	0.00156	0.00699
35	0.04142	0.00172	0.00769
36	0.03956	0.00156	0.00748
37	0.04153	0.00172	0.00798
38	0.04106	0.00168	0.00769
39	0.04240	0.00180	0.00818

When constructing an ESRF perdition model in credit card fraud detection dataset, the initial set of features (raw features) include information regarding individual transactions. By using the DR_IG technique for feature extraction as described earlier, a total of 39 features are predicted. However, it may not be the most appropriate evaluation criteria when evaluating fraud detection models because they tacitly assume that wrong prediction errors carry the same cost, similarly with the correct predictive instance. This assumption does not hold in practice, when wrongly predicting a fraudulent transaction as legitimate carries a significantly different financial cost than the inverse case. Also, for these experiments, the individual prediction model is trained according to the number of features included in the credit-card dataset. The prediction model with the best results is selected.

Table 4.16 Performance Measure Matrices for Credit-card Dataset

Number of Features	**DR_IG based ESRF**		
	RMSE	MSE	MAE
2	0.20195	0.04078	0.09541
3	0.19384	0.37576	0.09804
4	0.11601	0.01346	0.03961
5	0.10430	0.01087	0.03477
6	0.13641	0.18608	0.05079
7	0.10668	0.01138	0.03231
8	0.08281	0.00686	0.02024
9	0.06406	0.00410	0.01512

10	0.05134	0.00263	0.01041
11	0.06155	0.00378	0.01330
12	0.05215	0.00272	0.01105
13	0.04706	0.00221	0.00864
14	0.04455	0.00198	0.00838
15	0.04676	0.00218	0.00921
16	0.04274	0.00182	0.00740
17	0.00844	0.00205	0.00844
18	0.40660	0.00165	0.00746
19	0.03602	0.00129	0.00617
20	0.03923	0.00153	0.00685
21	0.04087	0.00167	0.00810
22	0.04134	0.00171	0.00757
23	0.0.3865	0.00149	0.00688
24	0.04289	0.00184	0.00789
25	0.03822	0.00146	0.00683
26	0.03957	0.00156	0.00751
27	0.03794	0.00144	0.00693
28	0.03690	0.00143	0.03780
29	0.04095	0.00167	0.04095
30	0.03829	0.00147	0.00691
31	0.04181	0.00175	0.00779
32	0.03744	0.00140	0.00669
33	0.04092	0.00167	0.00781
34	0.04120	0.00169	0.00774
35	0.03761	0.00141	0.00675
36	0.04192	0.00176	0.00773
37	0.03784	0.00143	0.00675
38	0.04111	0.00169	0.00780

Table 4.16 shows the different performance metrics of ESRF in the Credit-card using DR IG. The produced model's error value decreased the prediction model with 19 features, and the prediction model's error value was slightly increased under

19 features. Based on these results, the prediction model of ESRF with 19 features is selected as the best predictive model for the credit-card data set.

4.4.5 Result Comparison

A comparative study of the proposed dimension reduction approaches is presented in this section. The results of comparisons of the RMSE, MSE and MAE for various data sets are shown in Table 4.17 to 4.19. There is a dimensional difference between the input data. To resolve this discrepancy and speed up model operations, all experimental data is preprocessed and well suited for high-dimensional data. However, the corresponding input variable values used in the modeling process may be associated with each variable and affect the accuracy of the PBA system.

Table 4.17 Performance Comparison among the Dimension Reduction Techniques.

Dataset	ESRF	DR_PCA based ESRF	DR_IG based ESRF
DAS	3.74179	0.95125	0.71331
HPC2N	2.27283	0.07255	0.03984
Susy	0.39282	0.38774	0.27059
Credit-card	0.10338	0.05884	0.03602
KDD	0.10638	0.05847	0.03669

Through the analysis of evaluation results on different datasets, the predictors of ESRF using DR_IG obtains the best prediction results. It is worth nothing that losing too much information from reducing the dimension can drastically degrade the performance of PBA system. In these data features, PCA cannot work efficiently as it captures features with a broader variance dimension as an important feature. However, some data set's valuable information is low variance, resulting in poor results of prediction. Relative to other approaches, in all experimental data sets, ESRF using DR_IG delivers better predictive ability. When the features are independent of each other, DR_IG based ESRF technique can improve the prediction result of the PBA system.

Table 4.18 MSE Comparison among the Dimension Reduction Techniques.

Datasets	ESRF	DR_PCA based ESRF	DR_IG based ESRF
DAS	4.05188	0. 90487	2.02220
Susy	0.01392	0.00526	0.00519
HPC	0.31763	0.15035	0.13734
KDD	0.01121	0.00346	0.00129
Credit-Card	0.01193	0.00341	0.00135

Table 4.19 MAE Comparison among the Dimension Reduction Techniques.

Datasets	ESRF	ESRF with DR_PCA	ESRF with DR_IG
DAS	2.02131	0.27674	0.30464
Susy	1.41049	0.01392	0.09052
HPC	0.32434	0.31763	0.29020
KDD	0.03958	0.01121	0.00617
Credit-Card	0.04473	0.01193	0.00735

In order to compare the advantages and effectiveness of the proposed system, it is competed with another competitor, SRF from Spark MLlib. The number of trees in SRF has an impact on the overall prediction. Too large or too small predictors lead to a reduction in prediction accuracy. Many predictors take too long to calculate each data, and memory consumption during processing requires a lot of space. Conversely, a few predictors cannot learn the data sufficiently enough, leading to overfitting. The number of predictors of the proposed system is considered based on the best values of the hyperparameters optimization process. The MAE relation between SRF and ESRF is shown in Table 4.20.

Table 4.20. MAE Comparison (SRF vs. ESRF).

Predictors	Experimental Datasets				
	DAS	HPC	Susy	Credit-card	KDD
SRF	2.02131	1.41049	0.32434	0.03958	0.04473
ESRF	0.30464	0.29020	0.09052	0.00735	0.00617

Nevertheless, traditional learning techniques are often susceptible to the problem of the curse of dimensionality which refers to the degradation in the performance of the system as the number of features increases. To deal with this issue, DR_IG technique is applied as part of the data analysis to simplify the data model. By working with DR_IG, the proposed system can yield more accurate and readily interpretable results, while computational costs are significantly reduced. The results reveal that the proposed system can significantly outperform the SRF in terms of MAE and computational time. Figure 4.7 illustrates the processing time comparison of SRF and ESRF.

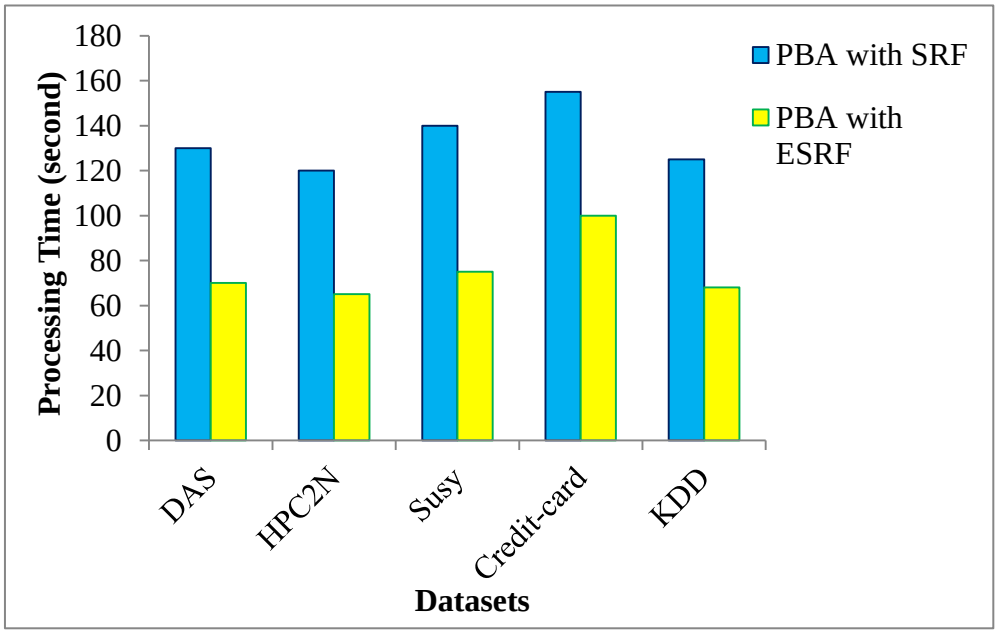


Figure 4.7 Comparison of Processing Time (ESRF vs. SRF)

The comparison results reveal that all the results of the proposed system achieved the minimum running duration compared with the competitor. The key

benefit of this study is that reducing the unimportant feature brings the ability to reduce computing cost and improved performance. The experimental results show that the proposed PBA system can not only provide good prediction ability, but also provide efficient performance with the shortest processing time for the entire experimental data set.

4.5 Chapter Summary

In this chapter, for processing large amounts of high-dimensional data, an efficient PBA system using ESRF is implemented. Many features that do not impact the model are discovered to minimize overhead processing time during the construction phase of the model. And, using dimensional reduction techniques, these features are reduced. To reduce unrelated functional variables in the data set, the proposed system employs proposed efficient dimension reduction approaches. In addition, to set the optimal hyperparameters, an effective strategy has been established. Two common reduction techniques have been deployed and the evaluation results for the PBA system have been analyzed. As a result, the advantages of the information gain, that is, the characteristics of the data in the predictive analysis, are absorbed by the proposed system. The important finding of this research is that the combination of hyperparameters optimization and dimension reduction technology can greatly enhance the proposed system's predictive performance. The comparative study shows that the proposed PBA system achieved the higher accuracy based on the power predictor, ESRF.

CHAPTER 5

REAL-TIME PBA SYSTEM

The advent of huge real-time data has caused disruptive changes in recent research areas, and real-time Predictive Big Data Analytics systems require the development of a scalable platform that can fuse multiple data layers to intelligently process data. With a big data approach, the challenge is not to collect data, but to analyze it properly and draw valuable conclusions in a timely manner [92, 93]. In this study, a Real-time Predictive Big Data Analytics (RPBA) system for big stock data is developed. In RBPA, a real-time analytics platform is implemented for large-scale daily stock data by applying HDFS and Apache Spark. ESRF is used for improving the accuracy of Predictive Analytics (PA) which includes dimensions reduction approach in the training process and hyperparameters optimization in the prediction process. It consists of choosing optimal hyperparameters that maximize the desired measure of stock market prediction models. The proposed system is constructed for the useful data analysis operations for both researchers and investors. For this purpose, an efficient RPBA system has been developed to provide constructive insights from huge amounts of stock data.

5.1 Real-time PBA System for Stock Trend Prediction

Forecasting the movement of stock trend is a challenging problem from both an academic and practical viewpoints due to the complexity of the stock direction. In addition, the stock index can be affected by numerous incidents involving local organization and external incidents such as political and diplomatic issues. PA examines the augmented raw stock data by forecasting the future stock trend in order to determine the behavior of the system [96, 97]. Nevertheless, there is no specific way to improve the prediction performance of PA system by only using the large amount of stock training data [43]. This leads to the overfitting problem [95], resulting in poor generalization performance [80]. Instead of using the raw stock data for prediction model building, feature engineering process is performed as a first step of the proposed RPBA system. The input stock data is transformed into new features using the financial Technical Indicators (TIs) because of the input data can include

bias of the model predictor's pre-knowledge of market situation. In order to handle the overfitting problem, the proposed RPBA system uses a powerful ESRF predictor to learn efficiently the incoming features which are created by TIs and produce the accurate predicted results. This section describes a proposed RPBA system for predicting stock market direction in a timely manner. The architecture of RPBA system for stock trend prediction is illustrated in Figure 5.1.

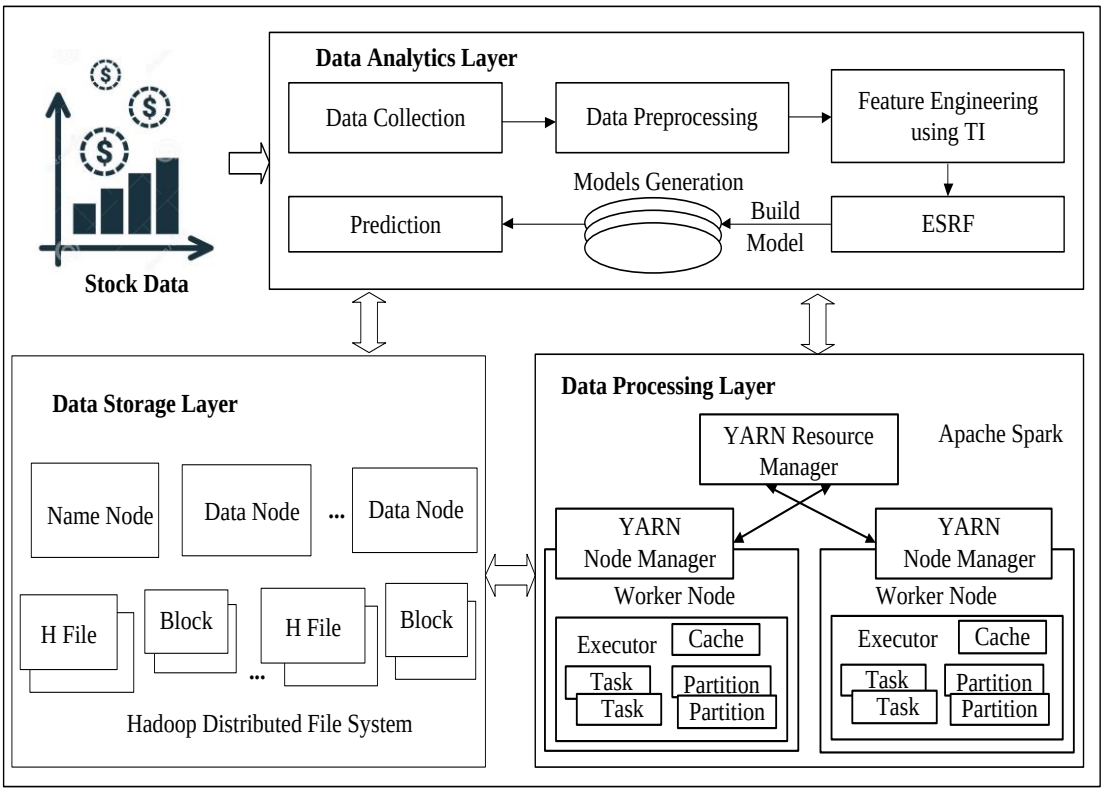


Figure 5.1 Architecture of RPBA System for Stock Trend Prediction

5.1.1 Data Storage Layer

To store a large amount of inventory data, HDFS is used as the storage layer in the RPBA system, providing a flexible and linearly scalable system, which can also provide high-throughput access to stock data and event distribution.

5.1.2 Data Processing Layer

Apache Spark is used as a processing layer to leverage big data and perform real-time stock data analysis in a distributed computing environment. Spark is the ultra-fast cluster computing engine of HDFS. The goal of the deployment is to be able to take profit of computation resources provided by Apache Spark. To achieve

fault tolerance efficiently, it provide a restricted form of shared memory. Spark can provide three main programming languages to write the user application. In this study, the Python API for Spark (PySpark) is selected to build the proposed RPBA system. The Spark driver node is used for the application context, and the Spark master node is used for resource allocation. *SparkContext* connects to the YARN cluster manager, which allocates resources among applications. After connecting, Spark gets *executors* on the nodes in the cluster, the process of performing calculations and storing large amounts of stock data in HDFS. The first step of distributed computing on Spark is to break the incoming stock data into batch fragments by Streaming computing, and then transform these data segments into Resilient Distributed Dataset (RDD). Operations on the input data stream are converted into operations on different RDD groups. Based on RDD abstraction, PySpark supports a powerful ESRF predictor for efficient predictive analytics in stock big data. By matchin

5.1.3 Data Analytics Layer

It includes five main components: data collection, preprocessing, feature engineering using Technical Indicators (TIs), prediction models development using ESRF, and stock market trend prediction for upcoming n-days. RPBA system collects real-time stock data from financial markets and stores these data in HDFS. To generate the prediction model of the RPBA system, the data is pre-processed, and then features spaces are constructed and stock trend patterns are identified according to various technical indicators. RPBA generates the stock forecasting model using ESRF (the ESRF forecasting model generation has been described in Section 4.3) and predicts whether the stock prices will rise or fall relative to the price n days ago. By using the Spark processing engine, all processes in the analytical layer are carried out in parallel computing manner in a distributed environment.

5.2 Data Collection

The real-time stock data from Yahoo! Financial Web Services API are read and captured by Pandas Data Reader. Pandas Data Reader leverages powerful features to greatly simplify data and allows the RPBA system to develop the volume trading strategies.

5.3 Data Preprocessing

Data used in the experiment is gathered through the S&P 500 index [73] and in terms of market companies from different sectors, namely AAPL (Information Technology), MSFT (Information Technology), BRK-B (Financials), GE (Industrials), JNJ (Health Care), T (Telecommunication Services), VZ (Telecommunication Services) and XOM (Energy).

In this research, stock trend prediction is considered as a classification problem with three targeted values (labels): up, down and stable. These target values represent the U.S Equities trend direction y_i, where $y_i \epsilon \{y_1$ =Up, y_2= Stable, y_3= Down$\}$. The experimental dataset include eight U.S stocks and the selected data sets cover the period from January 2009 to the present. The data is pre-processed not to contain null values in the datasets. As the Pandas data reader ingests the stock data as the JSON format and these data are transformed correctly to suitable CSV format for RPBA system. Each time point in the selected datasets consists of the following features, used to calculate the technical indicators: date, open, close, high and low price.

5.4 Feature Engineering using TIs

The abundance of big stock data meets the needs of investors traded on the stock market. Effective PA system are enable decision makers to identify the trend and patterns of emerging stock market. The nature of stock data always fluctuates and it is difficult to predict using the previous raw data. In order to minimize the risks of stock predictive analytics system, Technical Indicators (TIs) is used to ensure minimal risk. Minimizing forecast errors can minimize the investment risk of intraday trading [74, 75]. For accurately predicting the trends in U.S stocks prices and maximizing capital gain and minimizing loss, TIs are created based on the previous information and produced the valuable information to help the decision markers.

In RPBA system, a vector of attributes is calculated which is representing the values of several TIs in addition to the daily closing price for each trading day. These daily data instances and labels (for each stock) are used to generate the datasets for eight stocks from the U.S stock market. Twelve different technical indicators are selected for model inputs, namely SMA (Simple Moving Average), WMA (Weighted

Moving Average), ROC (Rate of Change), EMA (Exponential Moving Average), MOM (Momentum), STCK% (Slow Stochastic), STCD% (Fast stochastic), MACD (Moving Average Convergence Divergence), RSI (Relative Strength Index), WILLR% (Williams), A/D Osc (Accumulation/Distribution Indicator) and CCI (Commodity Channel Index).

5.4.1 Simple Moving Average (SMA)

SMA is an arithmetic moving average that is calculated by adding the most recent closing prices and dividing by the number of periods in the calculated average. The formula for calculating SMA is defined as follows:

$$SMA = \frac{C_t + C_{t-1} + \cdots + C_{t-n+1}}{n} \quad (5.1)$$

Where C_t means the closing prices at day t and n is the total number of period, n-days. SMA indicator is used as one of the TI features to smooth the stock prices signal.

5.4.2 Weighted Moving Average (WMA)

WMA calculates the average of a set of input values over a specified period of time. It focuses on recent prices rather than old prices. The data for each cycle is multiplied by a weight that is determined by the number of cycles selected. The formula for calculating WMA is defined as follows:

$$WMA = \frac{nC_t + (n-1)C_{t-1} + \cdots + C_{t-n+1}}{n + (n-1) + \cdots 1} \quad (5.2)$$

Where C_t means the closing prices at day t and n is the total number of period, n-days. WMA is used as a feature to smooth the data sequence. This reduces noise and makes it easier to spot data trends.

5.4.3 Momentum (MOM)

The MOM indicator compares the present price with the historical price over multiple selected periods. The formula for calculating MOM is defined as follows:

$$\text{MOM} = C_t - C_n \quad (5.3)$$

Where C_t means the closing prices at day t and C_n is the closing prices of previous n-days ago. MOM is used to validate trading strategies based on price trends.

5.4.4 Stochastic Oscillator (%K)

The Stochastic Oscillator is a momentum indicator that compares the specific closing price of a security during a specific period, n-days with the price in that range. The oscillator's sensitivity to market movements is reduced by adjusting the duration of the oscillator or moving average of the result. It is used to generate overbought and oversold trading signals using values in the range between 0 and 100. The formula for calculating Stochastic Oscillator (%K) is defined as follows:

$$\%K = \left(\frac{C_t - LL_n}{HH_n - LL_n}\right) \times 100 \quad (5.4)$$

Where C_t means the most recent closing prices at day t, LL_n is the lowest low prices in n-days, and HH_n denotes the highest high price in n-days. %K is also called as the slow stochastic indicator.

5.4.5 Stochastic Oscillator (%D)

It measures the relative position of the closing price relative to the price fluctuation range for n-days. This indicator is based on the assumption that the closing price continues to fall to the upper region of price changes in the preceding period as prices increase. If prices fall, the opposite is true. The formula for calculating Stochastic Oscillator (%D) is defined as follows:

$$\%D = \frac{\sum_{i=0}^{n-1}(\%K_{t-i})}{n} \quad (5.5)$$

Where %D is the simple days moving average of %K values. The values of %K and %D can vary from 0 to 100. If the values are greater than 80, the stock direction tends to overbought. On the other hand, the stocks are oversold if those values are less than 20.

5.4.6 Moving Average Convergence Divergence (MACD)

The MACD is momentum stock indicator which evaluates between the long term and short- term moving average of prices. MACD is calculated by subtracting

the EMA (26 days) values from the EMA (12 days). Exponential moving average (EMA) is exponentially weighted moving average values, which has greater weight and importance on the latest data points. In this research, EMA (9 days) of the MACD is considered as the signal line to buy or sell the stocks. The formula for calculating MACD is defined as follows:

$$\mathrm{MACD} = \mathrm{EMA}_{12}(\mathrm{C}) - \mathrm{EMA}_{26}(\mathrm{C}) \tag{5.6}$$

$$\mathrm{SignalLine} = \mathrm{EMA}_{9}(\mathrm{MACD}) \tag{5.7}$$

$$\mathrm{EMA(n)_t} = \mathrm{EMA(n)_t} + \alpha(\mathrm{C_t} - \mathrm{EMA(n)_{t-1}}) \tag{5.8}$$

Where C denotes the closing price series, EMA_n is the exponential moving average for n-days, α is a smoothing factor, 2/n+1(n is the number of data points).When MACD is under the SignalLine, a sell signal is indicated. On the other hand, it points out a signal for buying the stock.

5.4.7 Commodity Channel Index (CCI)

CCI measures the difference between the current price and the previous average price. CCI is a momentum-based oscillator used to determine overbought or oversold. The formula for calculating CCI is defined as follows:

$$\mathrm{CCI} = \frac{\text{Typical Price} - \mathrm{MA}}{0.015 \times \mathrm{D_t}} \tag{5.9}$$

Where typical price means $\sum_{i=1}^{n}(\mathrm{High} + \mathrm{Low} + \mathrm{Close})/3$, MA is the moving average values of the stock prices and D_t is the mean absolute deviation of the typical price. When CCI values is larger than 100, the stock is considered to overbought. If the CCI values are less than minus 100, it is known as a signal to be oversold.

5.4.8 Relative Strength Index (RSI)

RSI indicator is used as a feature that measures the magnitude of recent price changes and assesses the overbought or oversold status of stock prices. The values of the range are between 0 and 100. If the RSI values are beyond 70, stock trend indicate to be overbought and if RSI value is below 30, it shows to be oversold. The formula for calculating RSI is defined as follows:

$$\mathrm{RSI} = 100 - \frac{100}{1+(\frac{\sum_{i=0}^{n-1} \mathrm{Up}_{t-i}}{n})/(\frac{\sum_{i=0}^{n-1} \mathrm{DW}_{t-i}}{n})} \quad (5.10)$$

Where Up_t is the upward and DW_t is the downward price change at day t. In this research, standard value, 14 periods are used for calculating initial RSI value.

5.4.9 Williams Percentage Range (W%R)

W%R is used as TI, a momentum indicator that moves between 0 and -100 and measures overbought and oversold levels. When the W%R value is greater than -20, it points out to be sold and when the value is lower than -80, it shows to be bought.

$$\mathrm{W\%R} = \frac{\mathrm{HH_n} - \mathrm{C}}{\mathrm{HH_n} - \mathrm{LL_n}} \quad (5.11)$$

Where HH_n means the highest high prices, LL_n is the lowest low prices in the lookback and C the recent closing prices for n-days. In this work, 14 days periods are used as standard value.

5.4.10 Accumulation/Distribution (A/D) Oscillator

It is also called Chaikin A/D Oscillator which indicates the future stock trend. It needs the two parameters: a short duration n and a long duration m values. It is calculated by taking the EMA value of the n-period of the advertising line subtracted from the EMA value of the m-period of the advertising line.

$$Chaikin\ A/D\ Oscillator\ =\ (m\ day\ EMA\ of\ ADL - n\ day\ EMA\ of\ ADL) \quad (5.12)$$

Where EMA is the exponential moving average and ADL is the Chaikin A / D line. The A / D line or cumulative distribution line is a quantity-based indicator used to measure the cumulative flow of funds in and out of securities. This indicator essentially indicates whether most stocks participate in the market direction.

5.5 ESRF Model Generation for RPBA System

RF is an ensemble method that consists of several classification or regression trees created by randomly selecting training data samples. The advantage of random selection is that there is less correlation between trees in the forest. The detailed

procedure for generating ESRF prediction models is presented Section 4.3. In RPBA system, SRF classification is used to predict the price change direction of the SP 500 based on U.S stocks. For each training process, a feature spaces (feature variables) and the target variables (response variables) are generated. SRF can be considered to have evolved from a decision tree, and it is also a fast and non-specific way.

Hyperparameters optimization of SRF is performed to improve the prediction accuracy of stocks trend direction. The detail procedure of hyperparameters optimization in SRF has already described in Section 4.3 and 4.4.2.

5.6 Experiments and Results Discussion of RPBA system

This section describes all aspects of the RPBA system in all experiments. The requirements of the system specifications and experimental data sets are provided to implement the RPBA system. Table 5.1 lists detailed specifications of the software and hardware components.

Table 5.1 Testing System Specification of RPBA System

Parameters	Specification
OS	Ubuntu 16.04 Linux
Host Specification	Intel ® Core™ i7-6500U CPU @ 2.50GHz, 8GB Memory, 1000GB Hard Disk
VMs Specification	1GB RAM, 50 GB Hard Disk
Software Component	Hadoop 2.7.1 Apache Spark 2.4.4 Pyspark 3.5.2

In this research work, the stock market data are retrieved from eight companies. The stock market can be interpreted to some extent as a predictive market because it can be expressed using a variety of investor opinions. To use this knowledge in trading strategies, many prediction models have been implemented to

predict the future state of the stock market. In order to conduct the performance of RPBA system, different prediction models are developed based on four periods: inactive, sub-active, active and strong-active. For inactive training, daily stock data (from 1st January, 2009 to 3rd February, 2021) are collected as training dataset. For sub-active training, daily stock data (from 1st January, 2017 to 3rd February, 2021) are collected as training dataset. For active training, daily stock data (from 1st January, 2018 to 3rd February, 2021) are as training dataset. For strong-active training, daily stock data (from 1st January, 2019 to 3rd February, 2021) are as training dataset. The trading window is varied as 5, 10, 15, 20, 30, 60 and 90 days.

5.6.1 Performance Measurement Metrics

For evaluating the prediction performance of RPBA system, classification accuracy and F1 score are used. The classification accuracy of the system is calculated by using the following equation:

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{5.13}$$

Where TP is the percentage of positive cases correctly classified as positive. TN is the percentage of negative cases that are correctly classified as falling into the negative category. FP is the percentage of negative cases that are misclassified as positive. FN is the percentage of positive cases that were misclassified as negative.

F1 Score is computed with weighted average of precision and recall values. The formula for calculating F1 Score is defined as follows:

$$\text{F1 Score} = 2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \tag{5.14}$$

Where precision is the exact prediction ratio of positive predictions to total positive prediction observations and recall is the ratio of correctly predicted positive observations to all observations in the actual class.

5.6.2 Results Discussion of Hyperparameters Optimization

The success of an RPBA system for forecasting stock movements in real time depends on the performance of the forecasting model. This research work is to design intelligence models that learn from stock data using ESRF, a powerful machine

learning technique, and to predict future stock directions. Since the accuracy of the SRF depends on different combinations of hyperparameter values, various predictive models of the ESRF are developed and analyzed to enforce the model validity of the RPBA system. In this experiment, past and current stock data are captured in real time and preprocessed in off-line training. Then TIs are extracted to provide insight into future stock movements. For demonstrating the efficacy of the presented approach, the results of two stocks (Apple Inc. and Microsoft Inc.) are provided. Figure 5.2 shows the prediction accuracy of the different prediction models.

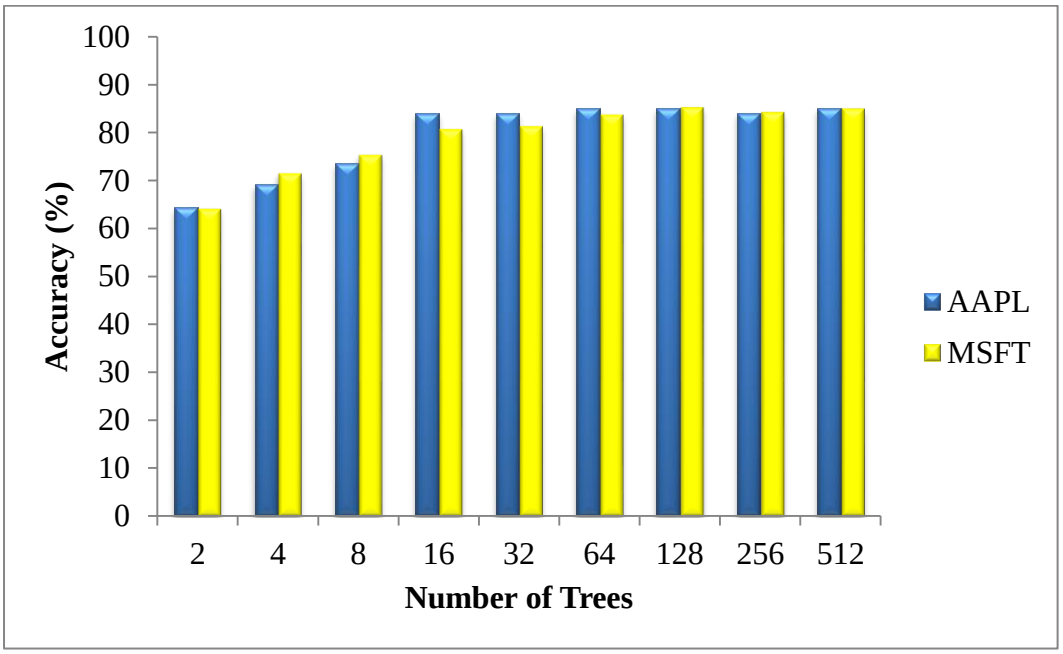


Figure 5.2 Prediction vs. Number of Trees in term of Accuracy

A greedy approach has been employed in this study to find the optimal combination of SRF hyperparameters (the number of trees and the maximum tree depth). The dependence of predictive performance on model parameters is described to inspect the prediction effectiveness of the ESRF model on the AAPL and MSFT stock market. As can be seen in Figure 5.2, the effect of tree numbers is very significant because the AAPL and MSFT data show very different profiles due to different number of trees. It is observed that forests with 128 and 512 trees have almost the same accuracy. The accuracy can't improve dramatically for far more than 128 trees. For the two datasets, it is observed that forest developed at 128 trees and up to 5 depths yields the highest accuracy of 85%.

5.6.3 Performance Comparison between Trading Periods

The trading period is a vital role in strategy design of RPBA system since it determinate how many samples should be considered in the training model to be used for prediction in the subsequent training period. In this research work, the straightforward trading periods are adopted as inactive, active, sub-active and strong active periods to test the prediction made by ESRF classifier for each distinct configuration. Based on the above trading periods, 5, 10, 15, 20, 30, 60 and 90-day-Ahead Prediction have calculated for each stock and compare the predictive models results. The accuracies of eight stock datasets for 5, 10, 15, 20, 30, 60 and 90-day-Ahead Prediction are illustrated from Figure 5.2 to 5.8.

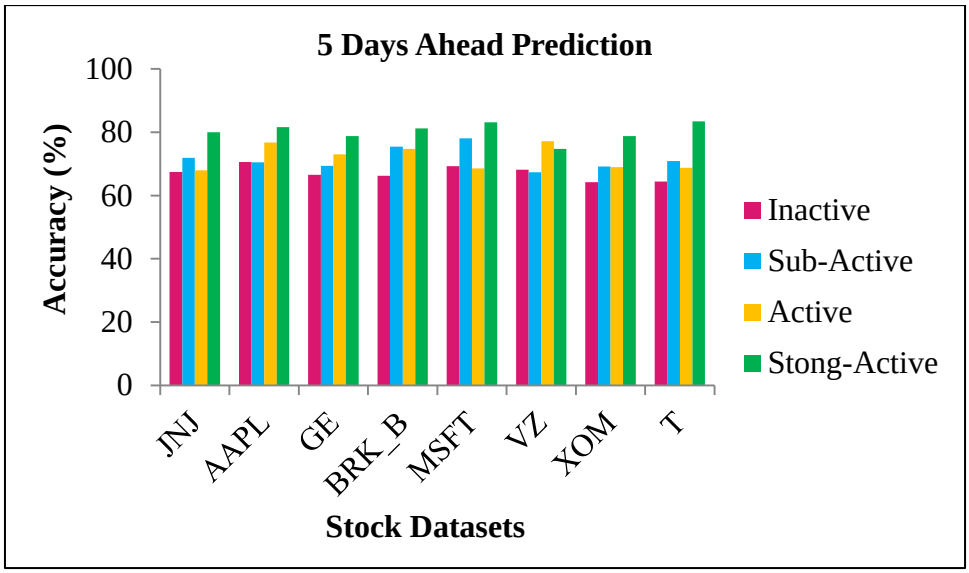


Figure 5.3 Accuracy for 5-day-Ahead Prediction

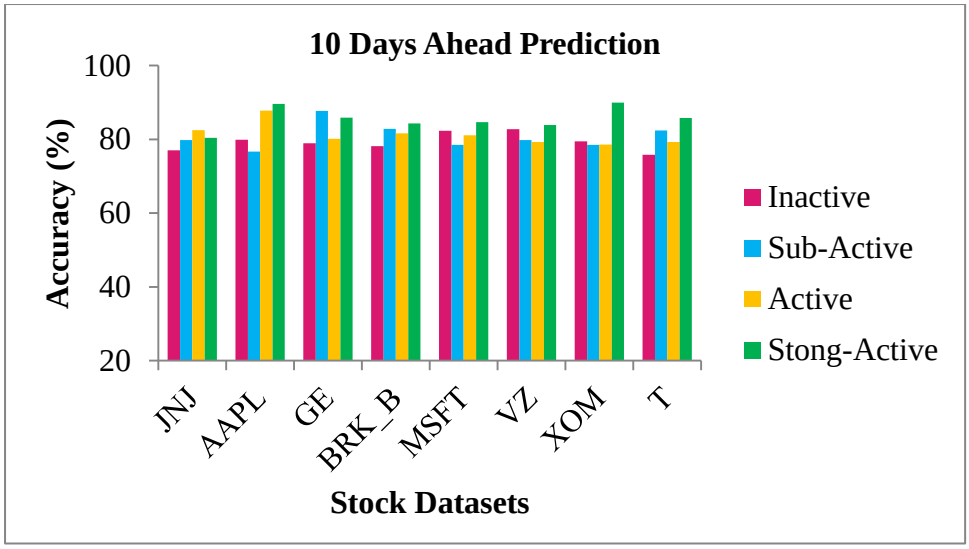


Figure 5.4 Accuracy for 10-day-Ahead Prediction

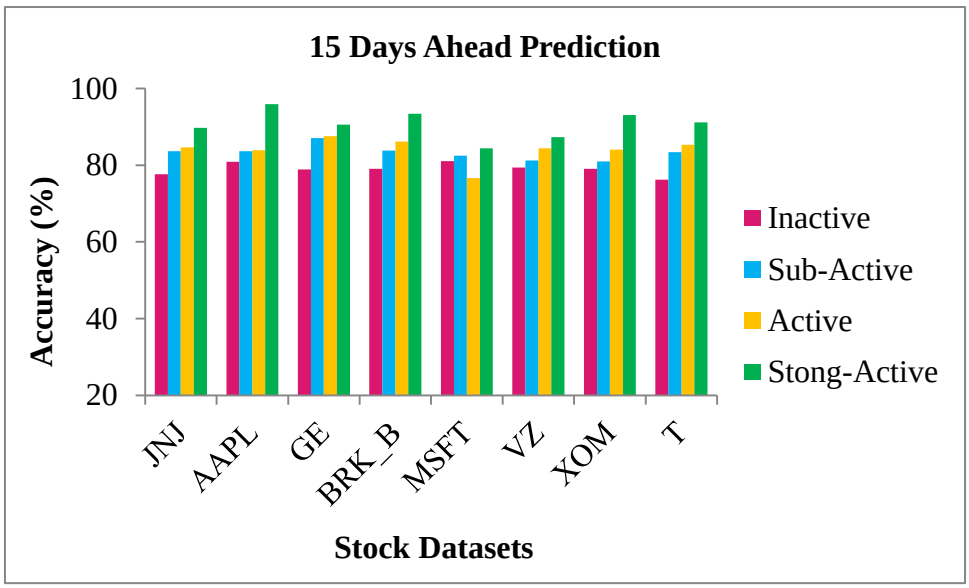


Figure 5.5 Accuracy for 15-day-Ahead Prediction

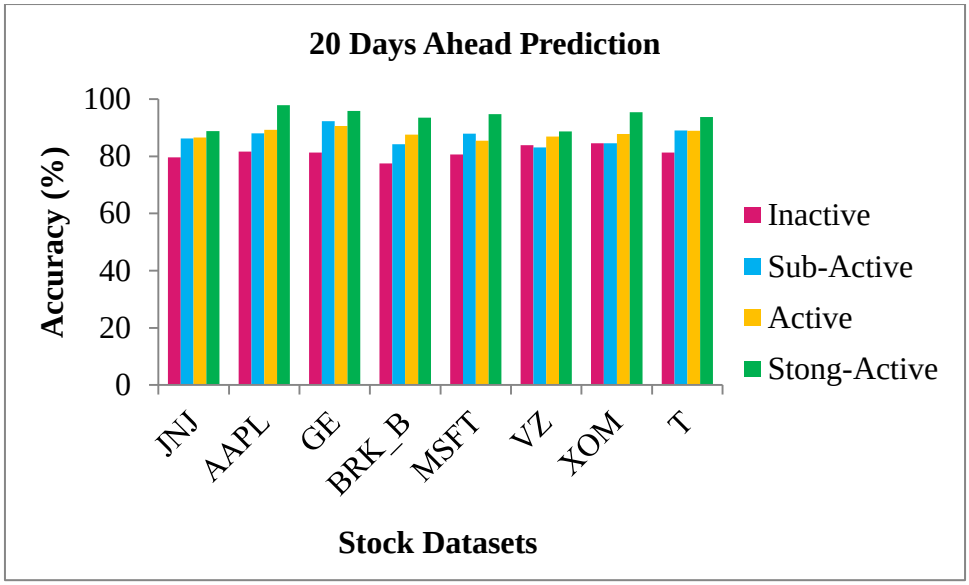


Figure 5.6 Accuracy for 20-day-Ahead Prediction

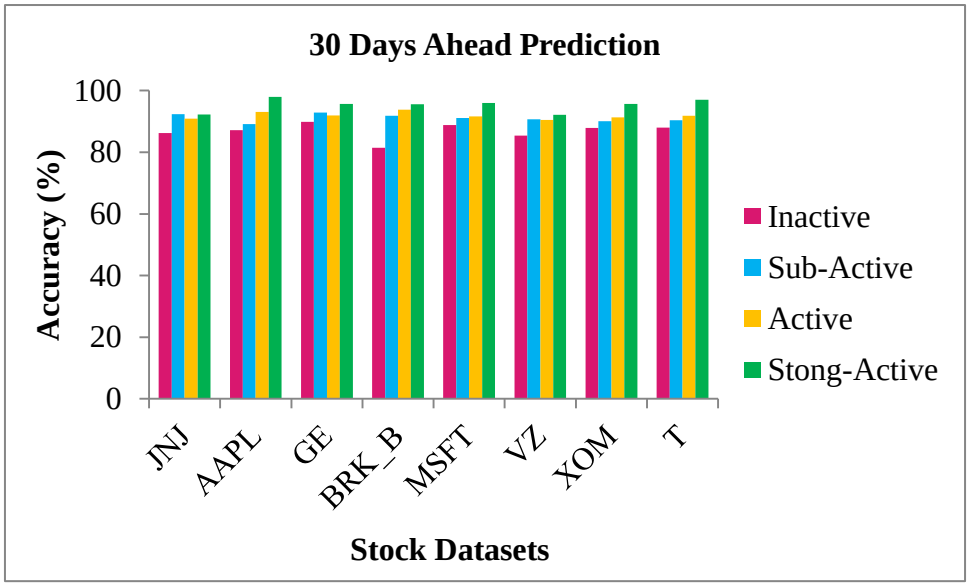


Figure 5.7 Accuracy for 30-day-Ahead Prediction

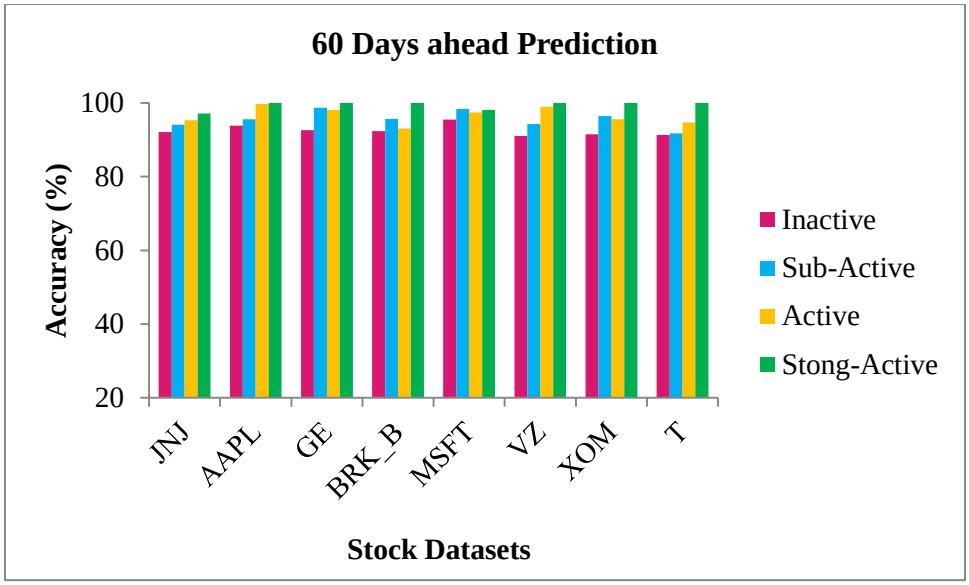


Figure 5.8 Accuracy for 60-day-Ahead Prediction

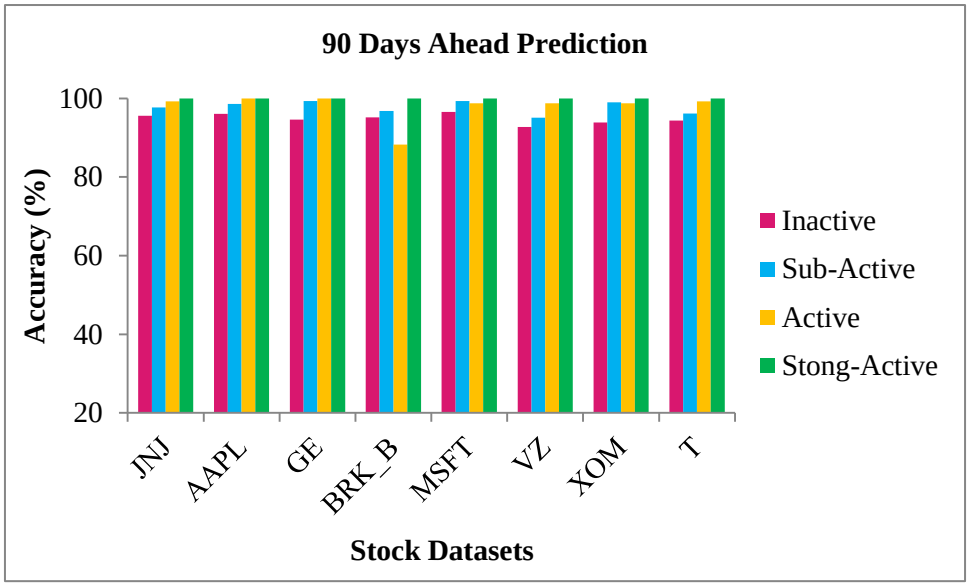


Figure 5.9 Accuracy for 90-day-Ahead Prediction

Knowledge discovery from comparative analysis creates new frontiers, such as trading strategy, based on the strength of classification accuracy and investigates the behavior of a particular trend (class) of stocks. As show in competitive results, the highest accuracy rate of stocks forecasting models reached with strong active training data. The accuracy of stocks predictive models against the trading period considered.

The accuracy of prediction generally increases as the training windows increase on all over the experimental datasets.

TABLE 5.2 F1-Score for 5-day-Ahead Prediction

Datasets	Trading Periods			
	Inactive	Sub-Active	Active	Strong-Active
JNJ	0.72358	0.72957	0.70701	0.83707
AAPL	0.76059	0.78252	0.74945	0.88407
GE	0.65362	0.61869	0.71890	0.80806
BRK_B	0.69580	0.78471	0.74350	0.80624
MSFT	0.74951	0.83816	0.74945	0.89333
VZ	0.71614	0.69498	0.78951	0.73988
XOM	0.67365	0.71671	0.68013	0.78045
T	0.67616	0.72759	0.70464	0.86963

F1 score results for eight stocks using ESRF classifier model are shown from Table 5.2 to Table 5.8. Trading period that are too short or too long period provide a degrade outperformance.

Table 5.3 F1-Score for 10-day-Ahead Prediction

Datasets	Trading Periods			
	Inactive	Sub-Active	Active	Strong-Active
JNJ	0.83869	0.81648	0.85625	0.84118
AAPL	0.87074	0.82979	0.90958	0.94177
GE	0.78103	0.77706	0.7867	0.83336
BRK_B	0.83389	0.85950	0.85826	0.87189
MSFT	0.86518	0.86125	0.87254	0.88824
VZ	0.80445	0.80353	0.83769	0.87323
XOM	0.79387	0.79682	0.79369	0.86906
T	0.77745	0.81130	0.82134	0.89801

Table 5.4 F1-Score for 15-day-Ahead Prediction

Datasets	Trading Periods			
	Inactive	Sub-Active	Active	Strong-Active
JNJ	0.81886	0.85113	0.86728	0.88816
AAPL	0.85499	0.88959	0.87553	0.97778
GE	0.78914	0.81.996	0.87457	0.89055
BRK_B	0.82573	0.87379	0.88804	0.94247
MSFT	0.85739	0.88777	0.83832	0.9037
VZ	0.80435	0.83096	0.87536	0.87934
XOM	0.78155	0.82938	0.84443	0.91707
T	0.78822	0.83894	0.87319	0.94755

Table 5.5 F1-Score for 20-day-Ahead Prediction

Datasets	Trading Periods			
	Inactive	Sub-Active	Active	Strong-Active
JNJ	0.83639	0.88584	0.8943	0.90066
AAPL	0.88453	0.91359	0.9264	0.98901
GE	0.8284	0.86239	0.8876	0.95833
BRK_B	0.85001	0.87163	0.89236	0.94363
MSFT	0.88401	0.92775	0.9036	0.96856
VZ	0.83694	0.94464	0.89147	0.90834
XOM	0.8156	0.86585	0.89147	0.94073
T	0.79818	0.89267	0.88052	0.95652

Table 5.6 F1-Score for 30-day-Ahead Prediction

Datasets	Trading Periods			
	Inactive	Sub-Active	Active	Strong-Active
JNJ	0.89851	0.93683	0.91943	0.93381
AAPL	0.90796	0.91177	0.95467	0.98795
GE	0.90962	0.89569	0.92305	0.96563
BRK_B	0.89220	0.93300	0.94095	0.96361
MSFT	0.91330	0.94766	0.95083	0.97689
VZ	0.88303	0.95460	0.91480	0.93532
XOM	0.82727	0.89723	0.90919	0.94763
T	0.86469	0.90534	0.93075	0.98154

Table 5.7 F1-score for 60-day-Ahead Prediction

Datasets	Trading Periods			
	Inactive	Sub-Active	Active	Strong-Active
JNJ	0.99900	0.94195	0.96452	0.97436
AAPL	0.94139	0.97361	0.99873	0.99999
GE	0.95865	0.97617	0.97357	0.99999
BRK_B	0.92621	0.92707	0.98228	0.99999
MSFT	0.94384	0.99517	0.95798	0.98983
VZ	0.96761	0.94972	0.99283	0.99999
XOM	0.90856	0.96388	0.93292	0.99999
T	0.92204	0.92753	0.96329	0.99999

Table 5.8 F1-score for 90-day-Ahead Prediction

Datasets	Trading Periods			
	Inactive	Sub-Active	Active	Strong-Active
JNJ	0.9655	0.94195	0.99221	0.99999
AAPL	0.97508	0.97361	0.9999	0.99999
GE	0.95122	0.97617	0.9999	0.99999
BRK_B	0.96741	0.92707	0.98606	0.99999
MSFT	0.97667	0.99517	0.99227	0.99999
VZ	0.93901	0.94942	0.99058	0.99999
XOM	0.9367	0.96388	0.98022	0.99999
T	0.94774	0.92753	0.97866	0.99999

According to the competitive results, the greatest value of F1 score for strong period data obtained almost 1 and the other predictive models with active and sub-active trading data are keeping within the reasonable limit. However, long-term stocks forecasts (60-days and 90-days ahead) are little discrimination between different training periods, indicating consistent classification accuracy with respect to the number of training samples. There is an improvement in performance with the increase in long-term prediction. The main finding of this study is that too much training data cannot impact the accuracy of predictive models in forecasting the stock trend. The model with strong period data proves to be robust in predicting the direction of stock movements. The RPBA system can achieve high accuracies for long-term predictions. The contribution of the current research work relates the selection of TIs and their application as features. In addition, ESRF by serving hyperparameters optimization is used to achieve the efficient and accurate prediction models of the system. By constructing the predictive models with strong active period, the propose RPBA system offers the excellent accuracy for the predictive ability of the developed models to predict whether stock prices are going to rise or fall beyond forecasts 90 days in advance.

5.7 Chapter Summary

In this chapter, an efficient RPBA system is implemented to forecast the direction of stock in a timely manner. Real-time U.S stock data from eight companies

are collected and preprocessed to have a better sign predictability. However, the stock is properly valued to avoid significant bias. The lack of the effect of bias in experimental datasets has also been ascertained by conducting the accuracy and F1 score values. ESRF classifier is used as a powerful predictor to build the various stock forecasting models that produce impressive results. This model has been observed to be robust in predicting stock directions. The proposed RPBA system is indeed a useful way to minimize stock market investment risk by predicting long-term stock direction.

CHAPTER 6

CONCLUSION AND FUTURE RESEARCH DIRECTION

The current era of Big Data has challenged scientists and data analysts to explore analytical approaches to generate useful insight into the massive amount of data generated in a wide range of real-life situation. Obviously, a great deal of attention has been dedicated to the creation of new efficient analytics system for analyzing the information available from mobile devices, social media, the Internet of Things, just to name a few of the major big data producers. While this massive volume of data can be quite useful to organizations as well as businesses, due to their high computational cost, it takes a lot of too much time for analytical and retrieval operations. A possible solution depends on the idea of a robust yet flexible analysis of big data. PA is an enabler of Big Data by analyzing information from current data sets to recognize patterns and reflect actual trends. For developing the PBA system, it has a high demand for efficiency in computing, exploration of information, problem-solving, and prediction situations on Big Data. In a distributed big data environment, such as the curse of dimensionality, the implementation of a PBA system has faced such critically important difficulty. This research focuses primarily on the development of an efficient PBA system and it has five main contributions:

1. A scalable and predictive big data analytics system is implemented for interpreting big data in a timely and effective manner.
2. Processing Engines selection (with the performance analysis of MapReduce and Spark) is carried out to support the computing efficiency of the proposed PBA system.
3. The proposed system is established by comparing the efficacy of the two reduction methods (DR IG and DR PCA) to improve the system's predictability and effectiveness.
4. To improve the prediction accuracy of the model, the proposed system is implemented by contributing Hyperparameters Optimization in SRF.
5. A real-time PBA system is implemented in an effective and timely manner to make informed investment decisions and subsequent potential profits based on stock market trend prediction.

6.1 Thesis Summary

This proposed research explores the greatest performance for a PBA system to deal with the Big Data problem of high dimensions. Firstly, the scalability test for processing the massive amount of Big Data is carried out by contrasting conventional and SML techniques. The study conducted scalability testing has been published in the publication [P1]. The selection of processing engines is then performed to determine effective parallel computing for the PBA system development. Consequently, the Spark processing engine is chosen to improve data computing performance. The work performed processing engines selection has been published in the publication [P2].

SML algorithm selection is performed to choose the best methodology for the PBA system by comparing four different SML algorithms: Random Forest, Gradient Boosting Trees, Decision Tree, and Linear Regression. According to the experimental results, Scalable Random Forest (SRF) has been selected (approximately) as a best predictor with lowest error rate (MAE= 0.05778, RMSE= 0.00339 and MSE 0.01046).

The explosion of high-dimensional big data is inherently large-scale because these data has spatial attributes and can also be combined with vast quantities of measurement data that grow over time. PBA plays a critical role in uncovering from these data business intelligence and forecasting results [83]. In this work, an effective ESRF-based PBA system is presented to detect faults in predictive analytics while overcoming some of big data challenges. Specifically, ESRF is proposed in conjunction with hyperparameters optimization and dimension reduction techniques. The proposed dimension reduction techniques can tackle challenge to reduce the overhead processing time of the prediction models. A satisfactory strategy of determining the optimal hyperparameters is provided as a description. The proposed system applies the two effective dimension reduction techniques and analyzes the evaluation outcomes. The proposed system brings forward an efficient dimension technique for processing tremendous data with high dimension. The detail procedure of implementing and evaluation of efficient PBA system has published in [P4].

In stock market environment, because of its profitability and difficulty, stock market analysis based on real-time data is considered as a difficult challenge for both investors and researchers. For that reason, RPBA is implemented to predict the future

stock direction based on four stock periods. The off-line training models are created and future patterns are predicted for the next n-days. There are two key stages to the proposed system: stock features engineering and stock prediction. First, stock features are constructed using the emerging TIs. Second, stock directions are predicted on TIs for the next period, which profiles fast computing speed and good performance in generalization.

6.2 Scope and Limitation

In this research, the proposed PBA system is developed with four main modules: data collection, preprocessing, dimension reduction, and prediction. The proposed system is implemented with different analytics architectures. However, the proposed RPBA system gathers and analyzes Yahoo Finance's real-time stock data.

In addition, stock prediction success depends primarily on historical data. In addition, stock market data is vulnerable to non-economic factors such as natural disasters and political decisions; thus, it is chaotic and volatile of course. The unpredictability of stock data is also attributed to incomplete information from the stock market's historical behavior to capture the correlation between future and previous prices. Incomplete stock market data information is often considered to be noisy features, making it a challenge to forecast a stock's future price. For developing the complete RPBA system for stock prediction, the need for effective tools and methods to mitigate risks and optimize profits equally accelerated due to the rapid growth in trade and investment.

The incoming livestock data is divided into batches of the predefined interval when predicting with SRF from Spark MLlib and each batch of data is treated as Spark Resilient Distributed Database (RDD). Such RDDs are then processed using operations such as mapping, reducing, joining, etc. The result is handed back in batches from these operations. Therefore, it is not processed in real-time, but Spark is processing in near real time. Therefore, the system is needed to work for full support of real-time processing in another real-time analytics platform.

6.3 Future Research Direction

Exponential data expansion has put tremendous pressure on PBA systems that have to keep the details flowing from systems to analytics platforms at high speeds. Data processing and storage are just a subset of the data-driven demands of today. Analytical value extraction is the preferred target. The key point to bear in mind is that prediction model building needs significant historical data for predictive results, involves intensive experimentation and is a time-consuming process. Future work could include more Feature Weighting Algorithms for improved classifier performance and run the experiment by varying the sample size. In the experimental part, different cases can be studied as an investigation of the behavior of classifiers varying the number of characteristics.

Another alternative would be to include the issues of dealing with various types of issues and datasets. To carry out a run-time prediction on a continuous stream of event data to enable real-time decision making, a predictive model based on a collection of aggregated data is applied. The two things are involved to get this point. It will be appropriate to export the predictive model developed through a stand-alone tool in a consumable form. A streaming operational analytics platform will also need to ingest the model and convert it into the appropriate predictive feature and feed the processed streaming event data to determine the expected outcome. This implementation of a complex predictive model, from its parent machine learning environment to an operational analytics environment, is a potential route to real-time, streaming event data to achieve a constant run-time prediction. Big data performance models that can handle dynamic and time-sensitive data analysis will be implemented.

Another interesting topic would be to extend the scoring measure and develop other scoring system. Since in this work we focus on optimizing the hyperparameters in SRF (using grid search), different hyperparameters tuning approaches for the efficient classifier should be analyzed using the high dimensional big data. It is crucial to build an infrastructure that can carry out fast data analysis that allows for real-time predictive analytics, and machine learning, as new applications generate increased volume and data complexity.

AUTHOR'S PUBLICATIONS

[P1] Myat Cho Mon Oo, Thandar Thein, "Performance Analysis of a Scalable Naïve Bayes Classifier on Beyond MapReduce", in Proceeding of 1st International Conference on Advanced Information Technologies (ICAIT,2017), Yangon, Myanmar, November 2017, pp. 8-13

[P2] Myat Cho Mon Oo, Thandar Thein, "Performance Analysis of a Scalable Naïve Bayes Classifier on MapReduce and Beyond MapReduce", in Proceeding of 16th International Conference on Computer Application (ICCA 2018), Yangon, Myanmar, February 2018, pp.58-64

[P3] Myat Cho Mon Oo, Thandar Thein, "Hyperparameters Optimization in Scalable Random Forests for Big Data Analytics", in Proceeding of IEEE 4th International Conference on Computer and Communication Systems (IEEE ICCCS 2019), Singapore, February 2019, pp.125-129

[P4] Myat Cho Mon Oo, Thandar Thein, "An Efficient Predictive Analytics System for High Dimensional Big Data", Journal of King Saud University- Computer and Information Sciences, (SCImago index –Q1) [Accepted]

BIBLIOGRAPHY

[1] A. Frank and A. Asuncion, “UCI machine learning repository”, 2010. [Online] Available: http://archive.ics.uci.edu/ml

[2] A. Kuo, D. Chrimes, P. Qin, and H. Zamani, “A Hadoop/MapReduce Based Platform for Supporting Health Big Data Analytics”, Journal of Studies in health technology and informatics, Vol. 257, pp. 229-235, 2019.

[3] A. Lulli, L. Oneto, D.Anguita, “Mining Big data with Random Forest”. Cognitive Computation, Springer, USA, pp.294-316, 2019.

[4] A .Tripathi, K. Sharma, M. Bala “A Novel Clustering Method Using Enhanced Grey Wolf Optimizer and MapReduce”. Journal of Big Data Research, vol. 14, pp. 93-100, December 2018.

[5] B. F. Huang, P. C. Boutros, “The Parameter Sensitivity of Random Forests”. BMC bioinformatics, Vol. 17, No. 331, 2016.

[6] B. V. Prabhu and M Dakshayini, “Performance Analysis of the Regression and Time Series Predictive Models using Parallel Implementation for Agricultural Data”, Procedia Computer Science, Vol. 132, pp. 198–207, 2018.

[7] C. H. Chen and C. Shih, “A Stock Trend Prediction Approach based on Chinese

[8] News and Technical Indicator Using Genetic”, In Proceeding of IEEE Congress on Evolutionary Computation, pp.1468-1472, 2019.

[9] C. Lohrmann and P. Luukka, “Classification Of Intraday S&P 500 Returns With A Random Forest”. International Journal of Forecasting, (2018). [Online] Available: https://doi.org/10.1016/j.ijforecast.2018.08.004

[10] C. Su and S. Huang, “Real-Time Big Data Analytics for Hard Disk Drive Predictive Maintenance”. Journal of Computers and Electrical Engineering, Vol.71, pp. 93-101, October 2018.

[11] D. Borthakur, 2008. "The Hadoop distributed file system: architecture and design, "Hadoop

[12] E. M. M. van der Heide, R. F. Veerkamp, M. L. van Pelt, C. Kamphuis, I. Athanasiadis, and B. J. Ducro, “Comparing Regression, Naive Bayes, And Random Forest Methods In The Prediction Of Individual Survival To

Second Lactation In Holstein Cattle". Journal of Diary Science, Vol. 102, pp. 9409-9421, 2019.

[13] E. S. Mosseini and M. H. Moatter, "Evolutionary Feature Subsets Selection Based on Interaction Information for High Dimensional Imbalanced Data Classification". Applied Soft Computing Journal, Vol. 82, 2019.

[14] F. Sun, G. Huang, Q. M. J. Wu, S. Song and D. C. Wunsch, "Efficient and Rapid Machine Learning Algorithms for Big Data and Dynamic Varying Systems", IEEE Transactions on Systems, Man, and Cybernetics: Systems, Vol. 47, and Issue. 10, October 2017.

[15] F. Sun, Y. Pan, J. White and A. Dubery, "Real-Time and Predictive Analytics for Smart Public Transportation Decision Support System". In Proceeding of 2nd IEEE International Conference on Smart Computing, 2016.

[16] F. Yang, Z. Chen,J. Li and L. Tang, "A Novel Hybrid Stock Selection Method with Stock Prediction", Journal of Applied Soft Computing,pp.820–831, 2019.

[17] G. Chandrashekar, F. Sahin, "A survey on feature selection methods", Computer and Electrical Engineering, Vol. 40, pp. 16–28, 2014.

[18] H.Hua, L. Tangb, S. Zhangb, H. Wang, "Predicting The Direction Of Stock Markets Using Optimized Neural Networks With Google Trends, Journal of Neurocomputing, (2018), https://doi.org/10.1016/j.neucom.2018.01.038

[19] I. Tsamardinos, G. Borboudakis, P. Katsogridakis, P. Pratikakis and V. Christophides, "A greedy feature selection algorithm for Big Data of high dimensionality". Journal of Machine Learning, Vol. 108, Issue 2, pp. 149-202, February 2019.

[20] J. Creighton and F.H. Zulkernine, "Towards Building a Hybrid Model for Predicting Stock Indexes". IEEE International Conference on Big Data, pp.4128-4133, 2017.

[21] J. C. Ang, A. Mirzal, H. Haron, H. N. A. Hamed, "Supervised, Unsupervised, A Semi-Supervised Feature Selection: A Review On Gene Selection", In proceedings of IEEE/ACM Trans .Comput. Biol. Bioinform.Vol. 13, pp.971–989, 2016.

[22] J. Xianyaa, H. Mo, L. Haifeng, "Stock Classification Prediction Based on

Spark Stock Classification Prediction Based on Spark". In Proceeding of 7th International Conference on Information Technology and Quantitative Management, Procedia Computer Science, pp. 243-250, 2019.

[23] K. KeerthiVasan, B. Surendiran, "Dimensionality Reduction Using Principal Component Analysis For Network Intrusion Detection, Perspectives in Science, 2016. [Online] Available: http://dx.doi.org/10.1016/j.pisc.2016.05.010

[24] L. A. Laboissierea, R. A.S. Fernandes and G. G. Large, "Maximum and Minimum Stock Price Forecasting Of Brazilian Power Distribution Companies Based On Artificial Neural Networks". Journal of Applied Soft Computing (2015). http://dx.doi.org/10.1016/j.asoc.2015.06.005

[25] L.He, R.A. Levine, A.J. Bohonak, J. Fan and J. Stronach, "Predictive Analytics Machinery for STEM Student Success Studies", Journal of "Applied Artificial Intelligence, Vol. 32, Issue 4, pp. 361-387, 2018.

[26] L. Khaidem, S. Saha and S. R. Dey, "Predicting the Direction Of Stock Market Prices Using Random Forest". Journal of Applied Mathematical Finance, 2016.

[27] M. Assefi, E. Behravesh, G. Liu, A. P. Tafti, "Big Data Machine Learning Using Apache Spark MLlib". In Proceedings of IEEE International Conference on Big Data, pp. 3492-3498, 2017.

[28] M. Kima, E. L. Parkb and S. Choa, "Stock Price Prediction through Sentiment Analysis of Corporate Disclosures Using Distributed Representation". Journal of Intelligent Data Analysis, vol.22, pp.1395–1413, 2018.

[29] M. Z. F. Nasution , O. S. Sitompul and M. Ramli, "PCA Based Feature Reduction To Improve The Accuracy Of Decision Tree C4.5 Classification". Journal of Physics: Conf: Series, Vol. 978, 2018.

[30] Movie Review Data. [Online] Available: https://www.kaggle.com/c/sentiment-analysis-on-movie-reviews

[31] N. D. C. Garc, A. L. M. Castaneda, D. E. Garc and M. V. Carriegos, "Effect of the Sampling of a Data set in the Hyperparameter Optimization Phase over the Efficiency of a Machine Learning Algorithm". Hindawi Complexity, Vol. 2019, pp. 1-16, 2019.

[32] N. Kannan, S. Sivasubramanian, M. Kaliappan, S.Vimal and A. Suresh, “Predicitve big Data Analytics on Demonetization Data Using Support Vector Machine”, Cluster Computing, pp. 1-12, 2018.

[33] O. Alfarraj, A. Alzubai and A. Tolba, “Optimized feature selection algorithm based on fireflies with gravitational ant colony algorithm for big data predictive analytics”, Neural Computing and Applications, Vol. 31, Issue. 5, pp. 1391-1403, May 2019.

[34] Ohsumed collection. [Online] Available: http://disi.unitn.it/moschitti/corpora.html

[35] P. Geneves, T. Calmant, N. Layaïda, M. Lepelley, S. Artemova, J. LucBosson, “Scalable Machine Learning for Predicting At-Risk Profiles Upon Hospital Admission”. Journal of Big Data Research, Vol.12, pp. 23-34, 2018

[36] P. Mohanty, D. Patel, P. Patel and S. Roy, “Predicting Fluctuations in Cryptocurrencies' Price using users' Comments and Real-time Prices”. In proceedings of IEEE 7th International Conference on Reliability, Infocom Technologies and Optimization, 2018.

[37] Pang and L. Lee. The Movie Review Data on kaggle. [Online] Available: https://www.kaggle.com/c/sentiment-analysis-on-movie-reviews, [Accessed: 15-Aug-2017]

[38] S. Basak,S. Kar, S. Sahaa,L. Khaidema and S.R. Deya, “Predicting The Direction Of Stock Market Prices Using Tree-Based Classifiers”. North American Journal of Economics and Finance (2018), [Online] Available: https://doi.org/10.1016/j.najef.2018.06.013

[39] S. Bernard, L. Heutte, and S. Adam, “Influence Of Hyperparameters On Random Forest Accuracy”. In MCS of Lecture Notes in Computer Science, Vol. 5519, pp.171–180, Springer, 2009.

[40] S. Ding, H. Zhu, W. Jia, C. Su, A Survey on Feature Extraction for Pattern Recognition, Artificial Intelligence Review, Vol. 37, No. 3, pp.169–180, 2012.

[41] S. Jeon, B. Hong and V. Chang, “Pattern Graph Tracking-Based Stock Price Prediction Using Big Data”. Journal of Future Generation Computer Systems, 2017. [Online] Available: doi:

http://dx.doi.org/10.1016/j.future.2017.02.010

[42] S. Lee, Y. Kang, N. S. Ialongo, V. V. Prabhu, “Predictive Analytics for Delivering Prevention Services”, Journal of Expert System with Application, Vol. 55, pp. 469-479, 2016.

[43] S. M. Idrees, M. A. Alam, P. Agarwal and L. Ansari , “Effective Predictive Analytics and Modeling Based on Historical Data”, Advanced in Computing and Data Sciences, pp. 552-564, July 2019.

[44] S&P 500 Companies with Financial Information. [Online] Available: https://datahub.io/core/s-and-p-500-companies-financials [Accessed: Dec 25, 2019]

[45] S. Y. Yang, S. Y. Mo , A. Liu , A. A. Kirilenko, “Genetic programming optimization for a sentiment feedback strength based trading strategy”. Journal of Neurocomputing, Vol. 264, pp. 29–41, 2017.

[46] T. Manojlović and I. Štajduhar, “Predicting Stock Market Trends Using Random Forests: A Sample of the Zagreb Stock Exchange”. In Proceeding of 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2015. [Online] Available: doi: 10.1109/MIPRO.2015.7160456

[47] T. Marma, A. Swetapadma and M. Abdar, “Decision Tree Predictive Learner- Based Approach for False Alarm Detection”. Journal of Medical Systems, Vol.43, No. 191, pp. 1-13, 2019

[48] T. Nguyen and S. Yoon. “A Novel Approach to Short-Term Stock Price Movement Prediction using Transfer Learning”, Journal of Applies Sciences, vol.9, Issue. 22, 2019.

[49] T. Zang and B. Yang, “Big Data Dimension Reduction using PCA”. In Proceeding of IEEE International Conference on Smart Cloud Computing, pp. 152-157, 2016.

[50] R. A. Kamble, “Short and Long Term Stock Trend Prediction using Decision Tree”. International Conference on Intelligent Computing and Control Systems, pp. 1371-1375, 2017.

[51] R. Prasad and C. Aruna, “Scalable and Flexible Big Data Analytic Framework (SFBAF) For Big Data Processing and Knowledge Extraction”, International Conference on Engineering Technologies and Big Data

Analytics, pp. 51-55, January 2016

[52] R.Venkatesh, C.Balasubramanian and M.Kaliappan1, “Development of Big Data Predictive Analytics Model for Disease Prediction using Machine Learning Technique”. Journal of Medical Systems, vol.43, Issue.272, 2019. [Online] Available: https://doi.org/10.1007/s10916-019-1398-y

[53] Reuters-21578 collection Apte' split. [Online] Available: http://disi.unitn.it /moschitti/ corpora.html

[54] V. Bolón-Canedo, N. Sánchez-Marono, A. Alonso-Betanzos, J. M. Benítez, F. Herrera. A Review of Microarray Datasets and Applied Feature Selection Methods, Inform.Sci.282 pp.111–135, 2014.

[55] W. Cukierski. The Enron Email Datasets on kaggle. [Online] Available: https://www.kaggle.com/wcukierski/enron-email-dataset, [Accessed: 15-Aug-2017]

[56] W. Debki. Twitter Data on Data world. [Online] Available: https: //data.world /datasets/ twitter, [Accessed: 12-Aug-2017]

[57] Weka tool. [Online] Available: http://www.filehorse.com/download-weka-64/

[58] X. Zhanga, A. Li and R. Panb, “ Stock Trend Prediction Based On A New Status Box Method And Adaboost Probabilistic Support Vector Machine”. Journal of Applied Soft Computing (2016). [Online] Available: http://dx.doi.org/10.1016/j.asoc.2016.08.026

[59] Y. Alsubaie, K. E. Hindi and H. Alsalman, “Cost-Sensitive Prediction of Stock Price Direction: Selection of Technical Indicators”. In IEEE Access, vol. 7, pp.146876 – 146892, [Online] Available: doi: 10.1109/ACCESS.2019.2945907

[60] Y. Li, C. Zoub, M. Berecibar , E. Nanini-Mauryc, J. C. Chand, P. Bosschea, J. V. Mierloa and N. Omar, “Random Forest Regression For Online Capacity Estimation Of Lithium-Ion Batteries”. Journal of Applied Energy, Vol.232, pp.197-210, 2018.

[61] Z. PENG, “Stocks Analysis and Prediction Using Big Data Analytics”. In Proceeding of International Conference on Intelligent Transportation, Big Data and Smart City, pp.309-312, 2019.

[62] Z. Tan ,Z. Yan and G. Zhu, “Stock Selection With Random Forest: An

Exploitation Of Excess Return In The Chinese Stock Market". Journal of Heliyon, vol.5, 2019. [Online] Available: doi: https://doi.org/10.1016/j.heliyon.2019.e02310

[63] Z. Wu, Y. Li, A. Plaza, J. Li, F. Xiao and Z. Wei, "Parallel and Distributed Dimensionality Reduction of Hyperspectral Data on Cloud Computing Architectures". IEEE Journal Of Selected Topics In Applied Earth Observations And Remote Sensing, Vol.9, No.6, June 2016.

[64] Z. Zhou,M. Gao,Q. Liu,H. Xiao, "Forecasting Stock Price Movements With Multiple Data Sources: Evidence From Stock Market In China". Journal of Physica (2019), https://doi.org/10.1016/j.physa.2019.123389

LIST OF ACRONYMS

%K	Slow Stochastic Oscillator
%D	Fast Stochastic Oscillator
adosc	Accumulation/Distribution Oscillator
CCI	Commodity Channel Index
DR_IG	Dimension Reduction with Information Gain
DR_PCA	Dimension Reduction with Principal Component Analysis
ESRF	Enhanced Scalable Random Forest
HDFS	Hadoop Distributed File System
IG	Information Gain Theory
MACD	Moving Average Convergence Divergence
MAE	Mean Absolute Error
ML	Machine Learning
MOM	Momentum
MSE	Mean Square Error
NB	Naïve Bayes
PA	Predictive Analytics
PBA	Predictive Big data Analytics
PC	Principal Component
PCA	Principal Component Analysis
RF	Random Forest
RMSE	Root Mean Square Error
RPBA	Real-time Predictive Big data Analytics
RDD	Spark's Resilient Distributed Dataset
RSI	Relative Strength Index

SMA	Simple Moving Average
SML	Scalable Machine Learning
SNB	Scalable Naïve Bayes
SRF	Scalable Random Forest
TIs	Technical Indicators
VM	Virtual Machine
YARN	Yet Another Resource Negotiator

APPENDIX A
CONFIGURATION AND SOFTWARE SET UP

The performance evaluation of the proposed PBA system is implemented with Hadoop Multi Node Cluster which consists of four computing nodes (VMs). The host machine run window 10 which has Intel® Core ™ i7-6500U 2.6 GHz processors, 8 GB physical memory and 930-GB disk. The specific details of the software components are Hadoop 2.7.1, Apache Spark 2.4.4, Scala 2.11.8 and Python 3.7. The installation and software configurations of the system are supported in the following subsection.

HADOOP 2.7 Multi Node Cluster Set Up

To develop the Hadoop cluster, SSH and java software are firstly installed on machine because it is a java -based framework for storing and processing the big data on distributed computing environment. Hadoop core appears to require SSH communicating with slave nodes and creating the process onto other slave nodes.

JAVA Installation

The java package is downloaded from oracle sites via the link: http://www.oracle.com/technetwork/java/javase/downloads/index.html. The step- by-step java installation of the system is described as follow.

*Made “java” directory under /usr/lib/jvm (The location is user option)

```
#sudo mkdir /usr/lib/jvm/java
```

* Extract java tar file and place it under /usr/local/java folder

```
# sudo tar –xzvf jdk-8u121-linux-x64.tar.gz –C /usr/lib/jvm/java/
```

*Inform the Java Home (location of java package) to Ubuntu Operating System

```
#sudo update-alternatives --install “/usr/bin/javac” “javac” “/usr/lib/jvm/java/jdk1.8.0_121/bin/javac” 1
```

```
#sudo update-alternatives --install "/usr/bin/java" "java" "/usr/lib/jvm/java/jdk1.8.0_121/bin/java" 1

#sudo update-alternatives --set "javac" "/usr/lib/jvm/java/jdk1.8.0_121/bin/javac"

#sudo update-alternatives --set "java" "/usr/lib/jvm/java/jdk1.8.0_121/bin/java"
```

* Export JAVA_HOME and PATH into ~/.bashrc using nano editor

```
# sudo nano ~/.bashrc

#export JAVA_HOME=/usr/lib/jvm/java/jdk1.8.0_121

#export PATH=$PATH: /usr/lib/jvm/java/jdk.8.0_121/bin
```

* Compile the .bashrc file and then view the installed java version

```
# source ~/.bashrc

#java -version
```

OpenSSH Configuration

SSH means secured shell which is used for the remote login. Hadoop requires SSH for communicating the nodes. Before starting the SSH set up, the existing SSH is needed to remove from the system. The configuration steps are described as follows:

*Remove OpenSSH client from the machine, update the Software state and install OpenSSH server

```
#sudo apt-get remove openssh-client

#sudo apt-get update

# sudo apt-get install openssh-server

#which ssh

#which sshd
```

*Generate RSA key pair

```
# ssh-keygen -t rsa -P ""
```

*Copy the contents of “id_rsa.pub” into the “authorized_keys” file to configure passwordless SSH

```
#cat  ~/.ssh/id_rsa.pub >>  ~/.ssh/authorized_keys
```

* Check by SSH to localhost

```
#ssh localhost
```

Hadoop 2.7.1 Installation on Ubuntu 16.0.4

Hadoop 2.7.1 software package is downloaded from the link: https://archive.apache.org/dist/hadoop/common/hadoop-2.7.1/hadoop-2.7.1.tar.gz. Then download file is extracted into the Hadoop file of the machine.

*Extract tar file into /usr/local/ and change the mood of the folder

```
# sudo tar -xvfz hadoop-2.7.1.tar.gz  -C /usr/local

#sudo chmod 777 –R /usr/local/Hadoop-2.7.1
```

* Export HADOOP_HOME and PATH into .bashrc using nano editor

```
# sudo nano ~/.bashrc

   export HADOOP_INSTALL=/usr/local/hadoop-2.7.1
   export PATH=$PATH:$HADOOP_INSTALL/bin
   export PATH=$PATH:$HADOOP_INSTALL/sbin
   export HADOOP_MAPRED_HOME=$HADOOP_INSTALL
   export HADOOP_COMMON_HOME=$HADOOP_INSTALL
   export HADOOP_HDFS_HOME=$HADOOP_INSTALL
   export YARN_HOME=$HADOOP_INSTALL
   export
```

```
HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_INSTALL/lib/native
   export HADOOP_OPTS="-Djava.library.path=$HADOOP_INSTALL/lib/native"
```

*Compile the .bashrc file and then view the installed hadoop version

```
#source ~/.bashrc

#hadoop version
```

* Configure the JAVA_HOME variable in the file of hadoop-env.sh

```
export JAVA_HOME=/usr/lib/jvm/java/jdk1.8.0_121
```

*Create HDFS directories to configure HDFS and YARN

```
sudo mkdir -p /usr/local/hadoop-2.7.1/hdfs/namenode

sudo mkdir -p /usr/localhadoop-2.7.1/hdfs/datanode

sudo chmod 777 -R /usr/local/hadoop-2.7.1/
```

* Modify /usr/local/hadoop-2.7.1/etc/hadoop/core-site.xml for all nodes

```
<configuration>
   <property>
   <name>fs.default.name</name>
   <value>hdfs://localhost:9000</value>
 </property>
 </configuration>
```

* Modify /usr/local/ hadoop-2.7.1/etc/hadoop/yarn-site.xml

```
<configuration>
 <property>
```

```
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
  <property>
    <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
    <value>org.apache.hadoop.mapred.ShuffleHandler</value>
  </property>
</configuration>
```

* Modify /usr/local/ hadoop-2.7.1/etc/hadoop/hdfs-site.xml

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>3</value>
  </property>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>file:/usr/local/hadoop-2.7.1/hdfs/namenode</value>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>file:/usr/local/hadoop-2.7.1/hdfs/datanode</value>
  </property>
</configuration>
```

* Create /usr/local/hadoop-2.7.1/etc/hadoop/mapred-site.xml from template

```
cp /usr/local/hadoop-2.7.1/etc/hadoop/mapred-site.xml.template /usr/local/hadoop-2.7.1/etc/hadoop/mapred-site.xml
```

*Modify /usr/local/hadoop-2.7.1/etc/hadoop/mapred-site.xml

```
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>
```

*Format the HDFS name node of the master machine

```
#  hdfs namenode -format
```

*Start all the HDFS nodes

```
# start-dfs.sh
# start-yarn.sh
```

*Check the running processes of HDFS from master node

```
#jps
NameNode
DataNode
NodeManager
SecondaryNameNode
Jps
```

```
ResourceManager
```

*Check the running processes of HDFS from each data node

```
#jps

Node Manager

Data Node

Jps
```

*Access Hadoop from browser using Web UI to view the status of cluster

```
http://localhost:8088
```

Python Installation

*****Install python 3 from terminal and add the python path

```
#sudo apt-get update

#sudo apt-get install python3.7

#sudo update-alternatives --config python3
```

Apache Spark Installation

Spark 2.4.4 software package is downloaded from the link via https://spark.apache.org/downloads.html. The installation steps are described as follow:

* Extract tar file into /usr/local/ and change the mood of the folder

```
#tar -xzvf spark-2.4.4-bin-hadoop2.7.tgz  -C  /usr/local/

#tar –xzvf scala-2.11.tgz –C /usr/local/ -C /usr/local/

#sudo chmod 777 –R /usr/local/spark-2.4.4-bin-hadoop-2.7
```

*Add SPARK_HOME and PATH into .bashrc using nano editor

```
#sudo nano ~/.bashrc

export SPARK_HOME=/usr/local/spark-2.4.4-bin-hadoop-2.7

export PATH=$PATH:$ SPARK_HOME/bin

export SPARK_LOCAL_IP=127.0.0.1

export SCALA_HOME=/usr/local/scala-2.11.8

export PATH=$PATH:$ SCALA_HOME/bin

export PYTHONPATH=$ SPARK_HOME/python

export PYSPARK_PYTHON=python3
```

*Create the spark-env.sh in the spark configuration file and add properties

```
#cd /usr/local/spark-2.4.4-bin-hadoop-2.7

#cp /local/spark-2.4.4-bin-hadoop-2.7/spark-env.sh.template /local/spark-2.4.4-bin-hadoop-2.7/spark-env.sh
```

*Create the spark-defaults.conf in the spark configuration file and add properties

```
#cp /usr/local/spark-2.4.4-bin-hadoop-2.7/spark-defaults.conf.template
/usr/local/spark-2.4.4-bin-hadoop-2.7/spark-defaults.conf
```

*Start a standalone master server by browsing (http://127.0.0.1:8080/) to view the status screen

```
# /usr/local/spark-2.4.4-bin-hadoop-2.7/sbin/start-master.sh
```

*Start a spark worker process

```
#/usr/local/spark-2.4.4-bin-hadoop-2.7/sbin/start-slave.sh
```

*Check the spark shell with python

```
# /usr/local/spark-2.4.4-bin-hadoop-2.7/bin/pyspark
```

*Stop all spark worker and master process

```
# /usr/local/spark-2.4.4-bin-hadoop-2.7/sbin/stop-slave.sh

# /usr/local/spark-2.4.4-bin-hadoop-2.7/sbin/stop-master.sh
```