An Implementation of Hybrid Enhanced Sentiment Analysis System using Spark ML Pipeline: A Big Data Analytics Framework

Raviya K[1]
Research Scholar, PG and Research Department of
Computer Science, Presidency College
Chennai, India

Dr. Mary Vennila S[2]
Associate Professor and Research Supervisor,
PG & Research Department of Computer Science
Presidency College, Chennai, India

***Abstract*—Today, we live in the Big Data age. Social networks, online shopping, mobile data are main sources generating huge text data by users. This "text data" will provide companies with useful insight on how customers view their brand and encourage them to make business strategies actively in order to maintain their trade. Hence, it is essential for the enterprises to analyse the sentiments of social media big data to make predictions. Because of the variety and existence of data, the study of sentiment on broad data has become difficult. However, it includes open-source Big Data platforms and machine learning techniques to process large text information in real-time. The advancement in fields including Big Data and Deep Learning technology has influenced and overcome the traditional restrictions of distributed computing. The primary aim is to perform sentiment analysis on the pipelined architecture of Apache Spark ML to speed upward the computations and improve machine efficiency in different environments. Therefore, the Hybrid CNN-SVM model is designed and developed. Here, CNN is pipeline with SVM for sentiment feature extraction and classification in ML to improve the accuracy. It is more flexible, fast and scalable. In addition, Naive Bayes, Support Vector Machines (SVM), Random Forest, Logistic Regression classifiers have been used to measure the efficiency of the proposed system on multi-node environment. The experimental results demonstrate that in terms of different evaluation metrics, the hybrid sentiment analysis model outperforms the conventional models. The proposed method makes it convenient for effective handling of big sentiment datasets. It would be more beneficial for corporations, government and individuals to improve their great value.**

Keywords—Big data; sentiment analysis; machine learning; apache spark; ML pipeline

I. Introduction

The success of Smart devices' makes people's daily lives more focused to mobile services. People use mobile devices to collect information about firms, products, deals and recommendations. Online consumer reviews for a wide variety of goods and services are widely accessible and evaluating the sentiment in customer feedback has become greatly useful for business, where companies can monitor positive and negative feedback about the brand which allow themselves to assess its over-all success and can also perform a major role in evaluating sales and optimizing business marketing approaches. Reviews from customers are one of the massive amounts of information. Since it includes millions of reviews from different websites, and the number of reviews is rising every day. This vast amount of data that increases every moment is known as big data, which involves modern technologies and architectures to capture and evaluate process to derive value from it [1]. Big data analytics is essential for the purposes of business and society. Big data requires strong machine learning methods, and environments to accurately analyses the data. A large amount of data cannot be processed by conventional methods, so to handle the huge amount of information, a new computing platform for big data, such as Apache Hadoop and Apache Spark, are intended to incorporate machine learning systems to attain high performance [2], [3]. Apache Spark, established in 2009 at University of California, is an open-source processing system. It has been one of the main frameworks in the world for large-scale data processing and analytics, achieving high efficiency for both batch and stream data. It is an API that is simple to use and run-on large datasets. For large-scale data processing, Spark is 100 times faster than Hadoop by utilizing memory computing and other optimizations. Sentiment Analysis of huge volume of data has become more and more significant and drawn many researchers. Sentiment Analysis, also referred like opinion mining, is characterized as a task to identify the views of authors on specific entities [4]. Sentimental analysis is used in various places, for example: To analyse the reviews of a product whether they are positive or negative, If a political party strategy has been successful or not, evaluate the ratings of a film and analyse information of tweets or another social media data [5]. Sentiment analysis is all about having people's real voice of a particular product, programs, organization, movies, news, events, problems and their characteristics. Social media monitoring apps in businesses rely primarily on sentiment analysis using machine learning to help them gain insights into mentions, brands and goods [6]. The machine learning is a subset of AI [7]. It trains the computers to learn and behave like human beings with the assistance of algorithms and data [8]. Machine learning is the science of preparing a system to learn and act from data [9]. Machine learning is being used by wide variety of applications, and the trend is rising every day and often denoted to as fixing the model with knowledge is the method of training the model. Fig. 1 depicts two sub-sections of machine learning algorithms.

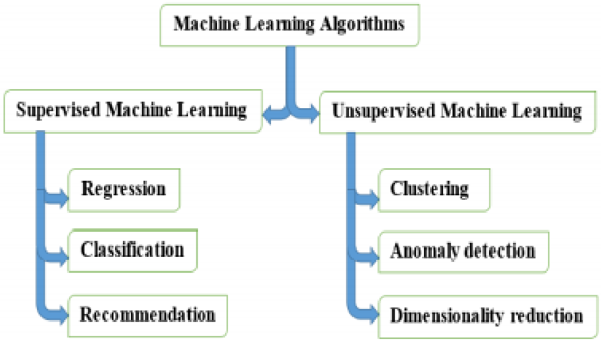

Fig. 1. Machine Learning Algorithms.

Supervised machine learning refers to working with a set of labelled training data to learn [7]. Every observation has a collection of features and label in the training dataset. Algorithms for supervised machine learning can be classified into regression, classification and recommendation engines. Unsupervised machine learning has been used when a dataset is un-labelled, means when a model does not require labelled data that is referred to unsupervised learning. These types of models try to learn or discovering hidden structures in un-labelled data or reduce the data down to its most important features [9]. With unsupervised learning there is no right or wrong answer [7]. They are commonly used during clustering, detection of anomalies and reduction of dimensionality. Sentiment analysis is recognized as a problem of classification and it can also be solved by the method of machine learning techniques [10]. In this work, MLlib by Spark, that is the machine learning libraries developed on top most layer of Spark to provide great-quality and high-speed machines. MLlib utilized Java, Scala, and Python, so that this may incorporate it into full workflows [8] that can be used for data analysis of large scale and since Spark's MLlib is a recent library developed in 2014. According the limited awareness of researchers, A small number of researches have been carried out to measure the sentiment of large-volume data using Spark's MLlib, so more analytical work is necessary for this field. The purpose of this study is to have new sentiment classification experiments on large volume of data by the Spark ML and DL with TensorFlow by implementing deep-learning models and evaluating their performance with existing algorithm. The rest of this paper is laid out as follows. In Section 2, we begin with a related work by Apache Spark ML. The core components of Apache Spark architecture are then introduced in Section 3. Section 4 introduces Apache Spark's machine learning library and computation. The ML pipelines for machine learning in Spark are discussed in Section 5. Then, for the suggested solution in Section 6, we move on to custom DL pipelines. Following that, Section 7 discusses some of the ML classifiers. The proposed methodology for sentiment analysis through big data analytics using hybrid CNN-SVM with spark DL is implemented in Section 8. Dataset, pro-processing, and word embedding are all discussed in Sections 9, 10, and 11. Section 12 and 13 describes the experimental setup, results, and discussions. Finally, summary and conclusions of this paper is presented.

II. Related Work

There is also a large amount of research on distributed systems for solving big data problems, particularly using Apache Spark. Sentiment analysis is the most ongoing research field in recent years that researchers have concentrated on, and several researchers have used various methods to perform sentiment analysis. The enormous interest in the field of sentiment analysis is largely dependent on availability of information and the developments in the internet. Advances in new techniques and algorithms have shown that combining sentiment analysis with machine learning can provide greater potential to predict the success of newly released products. Baltas et al. [8] implemented a Twitter data sentiment analysis system that used Spark MLlib for classification. On real Twitter info, three algorithms were used: decision tree, Naïve Bayes and logistic regression with binary and ternary classifications were evaluated. The Pre-processing of data is handled to maximize performance. The framework was tested on various sizes of datasets and various features. The authors suggested Naïve Bayes is superior than other classifiers and then the size of the dataset could influence the output of the classifiers. Sayed and et al [11] discovered in their paper that the Spark ML has an attraction over Spark MLlib in the performance and accuracy of big data analytics problems. Al-Saqqa and et al [12] examined about sentiment classification of big data using Spark's MLlib. In terms of efficiency, they find that the SVM is higher than other classifiers. AL-barznji and et al [13] addressed sentiment analysis using algorithms such as Naïve Bayes and SVM to evaluate the text with Apache Spark's aids. They discovered that in all situations, the SVM is more specific. Some recent works on deep learning models using Apache Spark as follows. The authors in [16] used a deep learning approach to detect the sentiment of Arabic tweets. Their approach relies on the utilization of pre-trained word vector representations. In [14], The ensemble model that combines both Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) models. LSTM is a special type of RNN which can learn long-term dependencies. LSTM was developed to prevent the long-term dependency problem that exists in standard RNNs. The results show good improvements in the accuracy and f1-score measurements over some other flat approaches. Alsheikh et al. [17] proposed a deep learning model for mobile big data analytics using Apache Spark. This model suggested the deep model parallelization by slicing the MBD into several partitions in spark RDD. The results illustrated the achievement of a better implementation using deep learning models through the Spark system compared to traditional lighter models. Various methods discussed in this paper for sentiment analysis are mainly based on results regardless of time complexity. If the size of the information is small, the analysis will be finished within the time limit. But time is a significant constraint when analyzing broad corpus, so the proposed structure would decrease the time required for sentiment analysis. This approach is much simpler and fully utilizes the capabilities of Spark ML framework. It is the Spark-based large-scale approach to sentiment analysis of product review data set without the need to build a sentiment lexicon or proceed with any manual data transcription.

III. Apache Spark Architecture for Bigdata

Apache Spark is a platform for bigdata analytics that can offer an enhanced substitution to the Map Reduce model. In contrast to map reduce model, Spark does not push the data to a disk for every step. The data is gathered in the memory till it is fully stored. If the memory is full then the data flows over to the hard drive. Therefore, the advantage of in-memory processing applied in Spark can make its processing very fast. The architectural layout of Spark is described in Fig. 2.

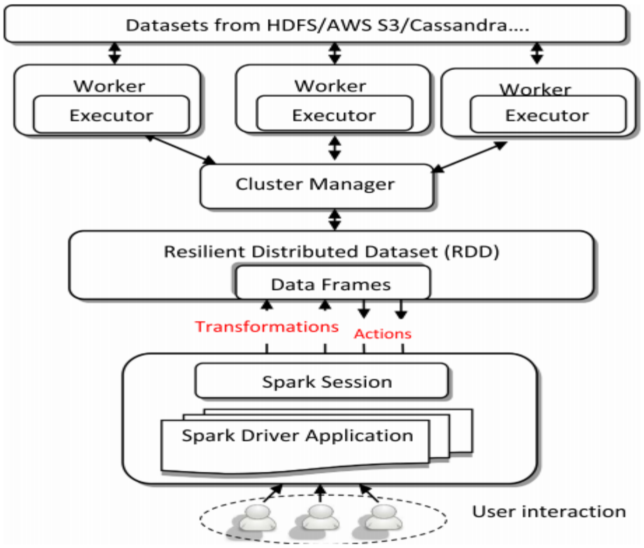

Fig. 2. Spark Architecture.

The Spark System utilizes a master/worker layout in which the master node handles the worker nodes. If the spark is mounted on the cluster, the executors mostly in worker nodes are automatically constructed and then the task is executed according to the instructions provided by the cluster manager. In the context of the Spark Session, the driver serves as the user interface from which the user transmits and receive the instructions. The Spark Session serves as the primary gateway for all communication. Spark operates at the centre on the basis of RDD idea. RDD's include applications like distributed data processing, the ability to use multiple data sources, fault tolerance and parallelism. Spark conducts two fundamental activities, i.e., transformations and actions. The transformations perform the RDD work, which transforms the input data using activities such as mapping, joining and key reduction to return the final output to RDD's and actions received the data from RDD's.

IV. Spark Machine Learning Library

Apache Spark is a very powerful tool for analytics of big data and presents excellent performance in terms with running time. MLlib is the first library supplied with Spark for machine learning shown in Fig. 3. Unlike single node machine learning frameworks, it is much efficient and scalable. MLlib also offers distributed processing options through parallel processing as well as facilitates the use of distributed architectures for big data analytics. This criterion would reduce the processing time needed but, at same time, control the duration to evaluate analytical results. If the role of machine learning has several predictions to measure and it is highly important. It provides some of the big data tool attempts to break down each machine learning section can also be used by the distributed architecture to decrease the total running time.

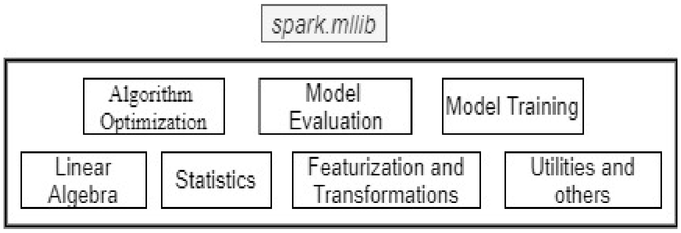

Fig. 3. Spark MLlib.

Integration is an additional benefit of Spark MLlib, this means that MLlib benefits from many software components accessible in the Spark ecosystem. It includes Spark GraphX, SQL, Spark Streaming and large spectrum of highly organized documents are freely accessible to the machine learning community. The basic machine learning utilities offered by spark MLlib are shown in Fig. 3. Spark Machine Learning Libraries provide executions of many algorithms and it is used for basic machine learning methods includes classification, regression, clustering, reduction of dimensionality, extraction and transformation of features, regular mining of patterns, and recommendations.

V. Spark ML Pipelines

Apache Spark ML is open-source platform for fast processing of large-scale data. Here Spark ML offers high-level APIs made on top of Data Frames for scalable data processing. In order to extract and predict the features, Spark uses the ML Pipeline to pass the data via transformers and estimator to implement the model.

Transformer: It is an algorithm to converts one Data Frame into next Data Format. Transformers are used here to convert text data with a feature vector.

Estimator: It is an algorithm to fit for the Data Frame to generate transformer. Estimators are used to train the model, which can convert input data to do the predictions.

Pipeline: The pipeline connects several transformers with estimators together to describe the ML workflow and it provides the mechanisms to build, evaluate and fine tune pipelines.

ML pipeline consisting of a set of pipeline stages to make it simpler using multiple algorithms to be merged into a single pipeline or workflow to be operated in a particular order. Spark is completely consistent with applications based on Java, as it uses Scala which runs on Java Virtual Machine. Using PySpark, it can work with RDD in Python programming language also. Further, Spark MLlib contain RDD format-based implementation of machine learning algorithms. Spark ML is based on datasets and enables us to use Spark SQL with it.

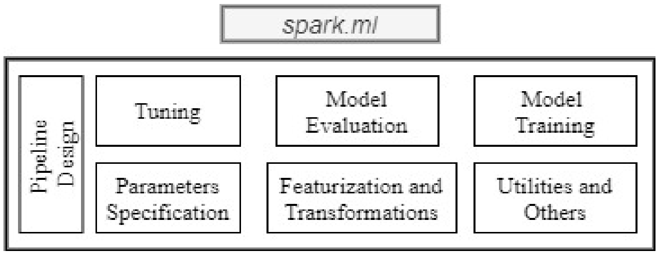

Fig. 4. Spark ML.

Feature extraction and manipulation tasks are very simple, as Spark SQL queries can now handle a lot. Typically, Dataset is in an extremely raw text, and this data usually goes through a loop or workflow where it is pre-processed, transmuted, and transformed until it is consumed for model training. The complete work flow of data transformation and its stages are fully encapsulated with the concept of ML Pipeline shown in Fig. 4. The ML libraries provide high-level API for implementation and fine-tuning of various machine learning pipeline models. It also allows us to save and load machine learning algorithms, trained models and pipelines so that it could reuse the results of previous steps to generate new results without the need to start the whole process from scratch.

VI. Apache Spark Custom DL Pipelines

The Deep Learning Pipelines is a high-level framework, which is applicable through Apache Spark MLlib. It facilitates the implementation of popular deep learning workflows. Furthermore, Databricks is an establishment started by the originators of Apache Spark provided the open-source library called DL pipelines and it support to develop deep learning models using python. It is a large-scale API integrate with the power of deep learning libraries such as TensorFlow and Keras. Both ML and DL provide a consistent collection of high-level APIs assembled on top of Data Frames to help users to build and tune functional pipelines for machine learning and deep learning models. To achieve the best outcomes in deep learning, experimenting with different training parameter values is an essential step called hyperparameter optimization. Since deep learning pipelines allow the exposure of deep learning training to Spark machine learning pipelines as a step, users can also rely on the integrated tuning work for hyperparameters.

VII. Spark ML Classifier

A. *Naive Bayes (NB)*

Naive Bayes algorithms are mostly used in sentiment analysis, spam filtering, and recommendation systems. They are quick and simple to implement. This is a basic multiclass classification algorithm based on Bayes' theorem. The conditional probability from each class function is calculated first and then the theorem of Bayes is employed to predict an instance's class label. Naïve Bayes is highly suitable for large dataset. Usually, the accuracy of NBC is increased when the data size increase.

B. *Support Vector Machine (SVM)*

It is a supervised algorithm for classification. The SVM is an efficient classifier that has been successfully used in all aspects of text classification. Documents of text are denoted by a vector. The classification is achieved via defining the hyperplane that raises the margin among two categories, and the support vectors are the vectors that describe the hyperplane. Based on the mined features of the training dataset, the model seeks to find the hyperplane defined by vectors that divide the positive and negative training vectors with largest probability.

C. *Logistic Regression (LR)*

Logistic regression is a model of regression where one value out of specific number of values can be taken by the dependent variable. It determines the relationship between the instance class and the extracted input features using the logistic function, while it is commonly used for binary classification, it can also be used to solve problems with multiple classes.

D. *Random Forest (RF)*

This supervised algorithm is also known as random decision forest and is commonly used for classification, regression, and other tasks. A forest is described as a grouping of trees. It consists of a large number of separate decision trees that work together to form an ensemble. All tree in the random forest is given a class prediction, and the model prediction is made using the class with the most votes. Random decision forest has been considered a robust and reliable classifier due to the principle of bagging and bootstrapping.

VIII. Proposed CNN-SVM using Spark DL

CNN is an artificial neural network that shares their weight. This creates CNN more analogous to the Network of biological neurons and decreases the complexity of both the weight and the network model. A fully connected soft-max layer is utilized as the classification layer for sentence level sentiment classification in CNN proposed by Kim [11]. This classification layer, however, has become too simple for the task of classifying sentiments. Fortunately, the CNN pooling layer output values are considered as function vectors of the input word. They may use as the input of some other classifiers. In this article, we propose an SVM classifier based on CNN that considers CNN as an automated feature learner and SVM as the classifier of sentiment. CNN outputs, the distributed function illustrations of the input terms, are considered to be features of SVM shown in Fig. 5.

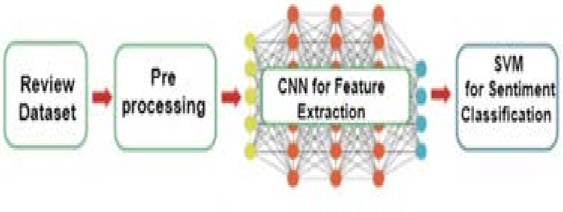

Fig. 5. Proposed Hybrid CNN_SVM Pipeline.

The CNN consists of four layers, namely the input, convolution, pooling and fully connected layer. To create feature map in the convolutional layer, all the available windows of words in the sentence are included. A max-over-time pooling operation is introduced to the function map following the convolution operation. This operation represents the pooling layer and gets a function vector of the m-dimension where the filter size is m. This operation establishes the layer of pooling and gets a vector of m-dimensional feature, where n is the total number of filters. Multiple filters of various window sizes are used in the CNN model. Then these features are moved to the last sheet, such as the fully connected layer, the output of which is the distribution of probability between labels. The pre-trained word embedding is fine-tuned throughout the training processing of the CNN by back propagation. Fine-tuning helps them to learn quite detailed word representations. If the terms may not exist in the embedding of pre-trained words, they are arbitrarily prepared. The vectors of the completely linked CNN layer are considered to be representations of the distributed sentence function, and then these representations of sentences are considered to be feature vectors of SVM classifier. The SVM classifier is trained using the feature vectors labelled with this sentiment. When this method is established for sentiment classification, the incoming sentences are converted to distributed feature representations, which are then fed into the SVM classifier for classification. It is predicted that such a pipeline model will join the benefits of CNN and SVM.

IX. Dataset

The Amazon online product review of about 100,000 reviews is used as data set for this study. This dataset contains products reviews of various domains like electronics, home appliances and books collected from amazon.com websites. To estimate the generalization error, the datasets is divided into two parts: training and testing data. After loading the datasets into the system, before applying and evaluating models the datasets are split into 80 % train dataset and 20 % test dataset randomly. In which, the training dataset is used to build a model, that is used for training the models to get predictions or recommendations. But testing dataset is independent of the training dataset, which is not used in the process of the building a model. The test dataset is to determine the efficiency of the proposed model.

X. Pre-Processing

The pre-processing of the data is the most key step. The purpose of the steps is to make data more machine-readable. Hence, uncertainty is reduced in feature extraction. In addition, to convert the streaming input to a data frame in order to run a pre-processing pipeline that includes the following steps:

Removing null reviews: This involves deleting any reviews with a null value.

Tokenization: The text is subdivided in to smaller tokens based on separator characters like white space, comma, tab, and so on in this phase.

Noise removal: This step involves removing any irrelevant information from the text that could affect the classifier's performance, including such numbers, punctuation marks, URL links, and special characters.

Stop-words removal: Non-descriptive words that can be displaced within the bag-of-words approach are known as stop words. Articles, prepositions, conjunctions, and pronouns are removed because they are not semantically necessary to characterize the viewpoint.

XI. Word Embedding

The interesting properties inside the data which you can use to make predictions are known as features. The process of converting raw data into inputs for a machine learning algorithm is known as feature engineering. For use in Spark machine learning algorithms, Features must be translated into feature vectors, which are numerical values that represent each feature's value. But for deep learning model, a neural network is a collection of neuron layers with the output of one layer being fed into the next layer. Each layer passes on the modified version of data to the next layer to promote more informative features further. Neural networks can't process direct terms; instead, they operate with word embeddings, or more precisely, feature vectors that represent certain words [15]. Neural networks can apply to any domain while learning features from the task at hand. Word2Vec is a predictive model for learning word embeddings from unstructured text that is computationally efficient. The first layer of CNN is the embedding layer converts words into real-valued feature vectors (embeddings) that take morphological, syntactical and semantic information of the words. the CNN use word embeddings feature as an input for the system. Each and every word was thus encoded as a 300-dimensional word vector that was supplied to the network. Word2vec is for word level embedding. Word level embedding is expected to obtain syntactic and semantic details, and character level embedding is projected to grab type and morphologic details. Data source on Google News (approximately words of 100 billion) is used to train the vectors in the proposed technique.

XII. Experimental Setup

Spark ML and DL have been used (open-source Bigdata tool) as development environment for performing the experiments. SparkFlow will take advantage of Spark ML's most important machine learning feature, which is the ability to combine deep learning pipelines with TensorFlow. SparkFlow allows users to train deep learning models in Spark and then link the trained model to a pipeline for smooth raw data predictions. The CNN models have been developed using TFLearn (a deep learning library) and it is pipeline with SVM using Spark ML pipeline.

Fig. 6 represents the major stages of this approach. The proposed method starts with data pre-processing and feature extraction, followed by the use of machine learning classifiers, Naïve Bayes, Support vector machine and logistic regression separately under Spark ML and proposed CNN-SVM using Spark DL environment. Finally, different metrics are used to measure the results. The PySpark library is built with the necessary Python API to run applications on top of the Spark. In order to estimate the efficiency of the proposed model, a series of tests were carried out, specifically, in terms of running time and classification results. There are five models are utilized for classification such as Naive Bayes, Logistic Regression, SVM and Random Forest and proposed hybrid CNN-SVM. Hyper parameter tuning is a technique for deciding the best parameters for achieving the highest degree of precision for the proposed model. Grid-Search with 5-fold cross validation has been applied on training data. The scalability and speed of the method is investigated in this experiment. The experiment is run four times: 1000, 2000, 4000, and 8000 reviews to see how easily Apache Spark can process data using the algorithm.

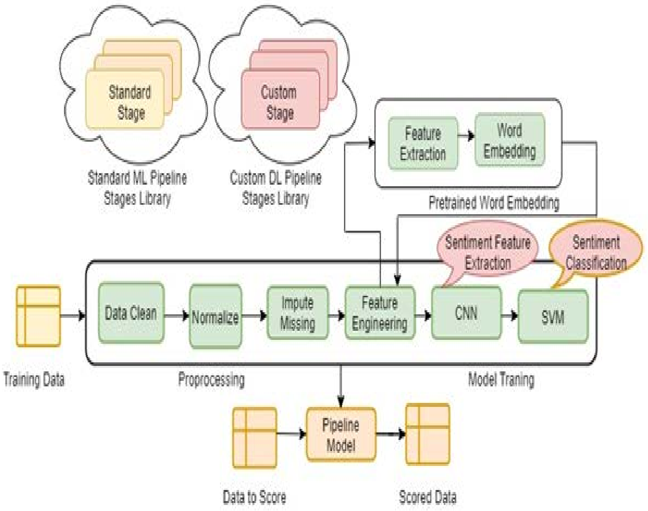

Fig. 6. Hybrid CNN_SVM using Spark DL Pipeline.

XIII. Result and Discussion

The results obtained based on processing time in Table I illustrate that in comparison to other sentiment analysis models, the proposed CNN-SVM has the fastest speed. Each algorithm's execution time is registered. There is a positive relationship between the amount of review data and processing time as it increases.

The classification accuracy of the models' performances is assessed using the random split method. Assuming that accuracy is influenced by a variety of factors, Table II displays average results of 5 algorithms based on assessment parameters such as average. For single-node systems, the values are taken into account. The model created with the CNN-SVM classifiers outshines the other classifiers. The outputs produced through this hybrid CNN_SVM Pipelined technique show higher rates of accuracy. Spark MLlib is a versatile method for analysing big data as evidenced by the findings of this research.

It presents spectacular performance in terms of running time and sentiment analysis of domain independent datasets shown in Table III. It is predictable that much greater performance is achieved in multi-node start-up configurations, as it is evaluated in ten node environments with much larger data sets. It also compared running time on growing number of nodes with varying size of data.

Table IV and Fig. 7 display the experimental outcomes. First, the computational efficiency of Spark is rising as the number of nodes in the computing cluster increases, and the subsequent experimental results indicate that the running time decreases. Secondly, the improvement of computational performance is stronger when this method is adopted to larger data. The results indicate that our proposed system performed well both in accuracy and running time.

TABLE I. Processing Time vs No of Reviews on Single Node

Algorithm	Time taken for no. of Reviews (seconds)			
	1000	2000	4000	8000
Naive Bayes	21s	29s	36s	50s
Logistic Regression	17s	18s	20s	22s
Random Forest	16s	16s	17s	19s
SVM	15s	15s	15s	16s
Proposed CNN-SVM	10s	10s	11s	11s

TABLE II. Accuracy of Various Models on Single Node

Algorithm	Accuracy for no. of Reviews				Average Accuracy
	1000	2000	4000	8000	
Naive Bayes	0.73	0.76	0.77	0.76	0.75
Logistic Regression	0.68	0.69	0.70	0.71	0.69
Random Forest	0.75	0.76	0.77	0.77	0.76
SVM	0.89	0.88	0.90	0.91	0.89
Proposed CNN-SVM	0.94	0.95	0.96	0.96	0.95

TABLE III. Evaluation Metrics for Various Domains

Domain	Performance Metrics			
	Precision	Recall	F-Score	Accuracy
Electronics	0.95	0.94	0.96	0.95
Kitchen Appliances	0.94	0.95	0.96	0.96
Books	0.94	0.93	0.94	0.94

TABLE IV. Number of Nodes vs Running Time

Nodes	Running time in seconds (50,000 Reviews)	Running time in seconds (1,00,000 Reviews)
(1)	240	461
(2)	148	266
(3)	110	180
(4)	89	154
(5)	61	102
(6)	52	84
(7)	45	72
(8)	42	63
(9)	40	55
(10)	35	50

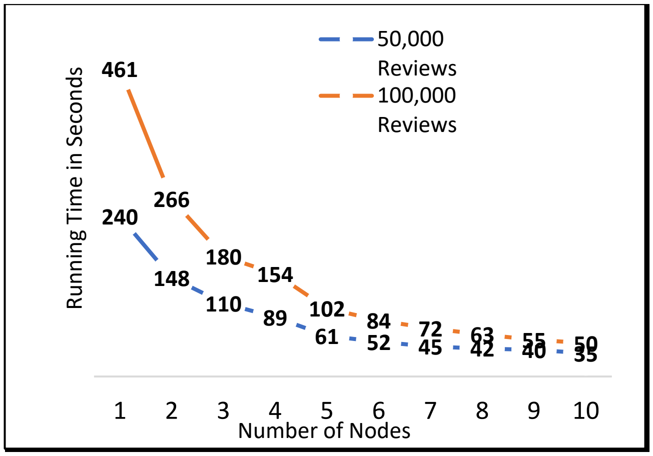

Fig. 7. Number of Nodes vs Running Time.

XIV. Conclusion

The main focus in this study was on rapidly implementing sentiment analysis on the Big Data sets. Spark MLlib has been used for handling a large volume of data as it is scalable. This paper offered new studies to classify sentiment on large amounts of data using Spark's MLlib with TensorFlow by implemented the proposed deep learning CNN-SVM model and the performance of this model is compared with different machine learning classification algorithms. Four classifiers were compared with our proposed model in terms of accuracy. The evaluation result shows that the proposed model has improved performance over the other classifiers. This work was implemented on multi node configuration with larger dataset. As part with our role in the future, we are working to perform an experimental evaluation of Spark MLlib in a number of programming languages (e.g., Python and R), and Software configurations that use a collection of large datasets with a range of data characteristics. In addition, we will develop a better deep learning model to extract optimized features in order to boost performance against other classification methods at a faster rate under large data volumes. The accuracy could be pointed for further development in future.

References

[1] S. Lenka Venkata, “*A Survey on Challenges and Advantages in Big Data*,” vol. 8491, pp. 115–119, 2015

[2] Ramesh R, Divya G, Divya D, Merin K Kurian, and Vishnuprabha V, “*Big Data Sentiment Analysis using Hadoop*”, IJIRST, Volume 1, Issue 11, pp. 92-98, 2015.

[3] Mohammed Guller, “*Big Data Analytics with Spark*”, ISBN13 (pbk): 978-1-4842-0965-3, 2015.

[4] Nurulhuda Zainuddin, Ali Selamat,” *Sentiment Analysis Using Support Vector Machine*”, IEEE International Conference on Computer, Communication, and Control Technology (I4CT 2014), Kedah, Malaysia,pp.333-337, 2014.

[5] Rajat Mehta,”*Big Data Analytics with Java*”, Published by Packt Publishing Ltd, ISBN 978-78728-898-0, UK, 2017.

[6] Kamal Al-Barznji, Atanas Atanassov, “*A Framework for Cloud Based Hybrid Recommender System for Big Data Mining*”, a journal of "Science, Engineering & Education", Volume 2, Issue 1, UCTM, Sofia, Bulgaria, pp. 58-65, 2017.

[7] Jason Bell, “Machine Learning: Hands-On for Developers and Technical Professionals”, Published by John Wiley & Sons, Inc., Indianapolis, Indiana, 2015.

[8] Baltas, A., Kanavos, A., & Tsakalidis, A. K. , “ *An apache spark implementation for sentiment analysis on twitter data.*“ In International Workshop of Algorithmic Aspects of Cloud Computing (pp. 15-25). Springer, Cham.

[9] Boštjan Kaluža, “*Machine Learning in Java*”, first published: Published by Packt Publishing Ltd, UK, 2016.

[10] Nick Pentreath, “*Machine Learning with Spark*”, Published by Packt Publishing Ltd. BIRMINGHAM – MUMBAI, 2015.

[11] Hend Sayed, Manal A. Abdel-Fattah, Sherif Kholief, "*Predicting Potential Banking Customer Churn using Apache Spark ML and MLlib Packages*“, A Comparative Study," (IJACSA) International Journal of Advanced Computer Science and Applications, vol. 9, pp. 674-677, Nov 2018.

[12] Samar Al-Saqqaa, b, Ghazi Al-Naymata, Arafat Awajan, "*A Large-Scale Sentiment Data Classification for Online Reviews Under Apache Spark*," in The 9th International Conference on Emerging Ubiquitous Systems and Pervasive Networks, EUSPN Belgium, 2018.

[13] Kamal Al-Barznji, Atanas Atanassov, "*Big Data Sentiment Analysis Using Machine Learning Algorithms*," in Proceedings of 26th International Symposium "Control of Energy, Industrial and Ecological Systems, Bankia, Bulgaria, May 2018.

[14] S. Hochreiter and J. Schmidhuber, "*Long short-term memory*", Neural computation, Volume 9 Issue 8, November 1997, pp. 1735–1780.

[15] L. Almuqren and A. Cristea, "*Framework for sentiment analysis of Arabic text*", Proceedings of the 27th ACM Conference on Hypertext and Social Media, Halifax, Nova Scotia, Canada, July 10-13 2016, pp. 315-317.

[16] L. Al-Horaibi and M. Khan, "*Sentiment Analysis of Arabic Tweets Using Semantic Resources*", International Journal of Computing and Information Sciences, Volume 13 Issue 1, January 2017, pp. 9-18.

[17] Alsheikh, M.A., Niyato, D., Lin, S., Tan, H.-P., Han, Z., 2016. , “*Mobile Big Data Analytics Using Deep Learning and ApacheSpark*“, 22–29. https://doi.org/10.1109/ MNET.2016.7474340.