**Map Reduce-based big data classification model using feature subset selection and hyper-parameter tuned deep belief network**

**A PROJECT REPORT**

CSA1580- Cloud Computing and Big Data Analytics for Cloud API

**Submitted to**

SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES

**In partial fulfilment of the award of the degree of**

BACHELOR OF ENGINEERING IN COMPUTERSCIENCE

**By**

R. R. Ragul [192211887]

**Supervisor**

Dr. GNANA SOUNDARI



**SAVEETHA SCHOOL OF ENGINEERING**

**SIMATS CHENNAI- 602105**

**JUNE-2024**

**CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.NO.** | **Title** | **Page. No.** |
| 1 | ABSTRACT | 3 |
| 2 | INTRODUCTION | 3-4 |
| 3 | MATERIALS AND METHODS | 4-6 |
| 4 | RESULTS AND  DISCUSSIONS | 6-7 |
| 5 | CONCLUSIONS | 7 |
| 6 | REFERENCES | 7 |

**ABSTRACT:**

In this paper, we propose a MapReduce-based big data classification model that integrates feature subset selection and a hyper-parameter tuned Deep Belief Network (DBN) to efficiently handle and classify large-scale datasets. The approach leverages the scalability of the MapReduce framework to preprocess data, select relevant features, and train the DBN in a distributed manner, thereby optimizing computational resources and reducing processing time.

Firstly, we implement feature subset selection using a distributed MapReduce paradigm, which calculates the relevance scores of features across data chunks in parallel, and then aggregates these scores to identify the most significant features. This step ensures that the dimensionality of the dataset is reduced, focusing on the most informative features and improving the efficiency of subsequent modeling steps.

Next, we construct the DBN, a deep neural network consisting of multiple layers of Restricted Boltzmann Machines (RBMs). Each RBM is trained layer-wise using contrastive divergence, and the model is fine-tuned through backpropagation. The training process is distributed using MapReduce, where each mapper processes a subset of data to train the RBM layers, and reducers aggregate the learned weights to update the global model iteratively.

Hyper-parameter tuning is performed using advanced optimization techniques such as Grid Search, Random Search, or Bayesian Optimization to determine the optimal configuration of the DBN. This step is crucial for enhancing the model's performance and ensuring robust classification results.

Finally, the trained DBN is employed to extract features from the input data, which are then used to train a classifier (e.g., Softmax Regression or Support Vector Machine) for the final classification task. This multi-stage process ensures that the model is both efficient and accurate in handling and classifying large-scale datasets.

Experimental results on benchmark datasets demonstrate the efficacy of our proposed model, showcasing significant improvements in classification accuracy and computational efficiency compared to traditional methods. This approach provides a scalable and effective solution for big data classification challenges, leveraging the strengths of both MapReduce for distributed processing and DBNs for deep learning.  
  
**Key words:** Big Data Classification, MapReduce, Feature Subset Selection, Deep Belief Network (DBN), Hyper-parameter Tuning, Distributed Computing, Parallel Processing

**INTRODUCTION:**

In the era of big data, the rapid growth of data volumes presents significant challenges and opportunities for data analysis and machine learning. Traditional data processing and classification techniques often struggle to cope with the sheer scale and complexity of large datasets. This necessitates the development of scalable and efficient methods to extract meaningful patterns and insights from vast amounts of data. In this context, deep learning models, particularly Deep Belief Networks (DBNs), have shown great promise due to their ability to learn hierarchical representations and capture complex data distributions. However, training deep learning models on large-scale data can be computationally intensive and time-consuming.

To address these challenges, we propose a novel MapReduce-based big data classification model that integrates feature subset selection with a hyper-parameter tuned Deep Belief Network (DBN). The MapReduce framework, a widely-used distributed computing paradigm, allows for the efficient processing of large datasets by dividing the data into smaller chunks that can be processed in parallel across multiple nodes in a cluster. By leveraging MapReduce, we can significantly reduce the computational burden and accelerate the training process of the DBN.

Our approach begins with a distributed feature subset selection process, where the relevance of each feature is evaluated in parallel across different data partitions. This step ensures that only the most informative features are retained, thereby reducing the dimensionality of the dataset and improving the efficiency of the subsequent deep learning model. Feature selection not only enhances the computational feasibility of training the DBN but also helps in mitigating the risk of overfitting and improving model generalization.

Following feature selection, we construct the DBN, a deep neural network composed of multiple layers of Restricted Boltzmann Machines (RBMs). Each RBM is trained in a layer-wise manner using contrastive divergence, and the parameters are fine-tuned through backpropagation to optimize the network for classification tasks. The training process is distributed using MapReduce, where mappers independently train the RBM layers on different data chunks, and reducers aggregate the learned weights to update the global model iteratively. This distributed approach ensures scalability and efficient utilization of computational resources.

To further enhance the performance of the DBN, we perform hyper-parameter tuning using techniques such as Grid Search, Random Search, or Bayesian Optimization. These methods help in identifying the optimal configuration of the DBN, including the number of layers, the number of hidden units, learning rates, and other crucial parameters. Hyper-parameter tuning is essential for achieving high classification accuracy and robust model performance.

Finally, the trained DBN is used to extract features from the input data, which are then fed into a classifier, such as Softmax Regression or Support Vector Machine (SVM), to perform the final classification. The integration of feature subset selection, distributed DBN training, and hyper-parameter tuning results in a powerful and scalable big data classification model.

**Materials and methods:**

1. Data Collection and Preprocessing

Datasets: Large-scale benchmark datasets from domains such as image recognition, text classification, or biomedical data.

Data Cleaning: Removing noise, handling missing values, and normalizing data.

Data Splitting: Dividing the dataset into training, validation, and test sets.

2. Feature Subset Selection

Feature subset selection is crucial for reducing the dimensionality of the data and focusing on the most informative features.

MapReduce Implementation:

Map Function:

- Input: Data chunk (a subset of the dataset).

- Process: Calculate relevance scores for each feature using techniques such as Chi-Square, Mutual Information, or Recursive Feature Elimination (RFE).

- Output: Key-value pairs (feature, relevance score).

Reduce Function:

- Input: Key-value pairs from mappers.

- Process: Aggregate the relevance scores for each feature.

- Output: Top N features based on aggregated scores.

Python code:

def feature\_selection\_mapper(data\_chunk):

for feature in data\_chunk:

relevance\_score = compute\_relevance\_score(feature)

emit(feature, relevance\_score)

def feature\_selection\_reducer(key, values):

aggregate\_score = sum(values)

if aggregate\_score > threshold:

emit(key, aggregate\_score)

3. Deep Belief Network (DBN) Construction

A DBN is a type of deep neural network with multiple layers of Restricted Boltzmann Machines (RBMs).

Steps:

RBM Training:

- Train each RBM layer using contrastive divergence.

- Initialize weights and biases for each layer.

Fine-Tuning:

- After pretraining the RBM layers, fine-tune the entire network using backpropagation.

Hyper-Parameter Tuning:

Parameters to Tune:

- Number of layers

- Number of hidden units per layer

- Learning rate

- Batch size

- Number of epochs

Optimization Techniques:

- Grid Search, Random Search, or Bayesian Optimization to find the optimal hyper-parameters.

4. MapReduce Implementation for DBN Training

Layer-wise Training:

Map Function:

- Input: Data chunk and RBM layer.

- Process: Train the RBM layer on the data chunk.

- Output: Learned weights for the RBM layer.

Reduce Function:

- Input: Learned weights from mappers.

- Process: Aggregate weights to update the global model.

- Output: Updated global weights for the RBM layer.

Python Code:

def rbm\_training\_mapper(data\_chunk, rbm\_layer):

local\_weights = train\_rbm\_layer(data\_chunk, rbm\_layer)

emit(rbm\_layer\_id, local\_weights)

def rbm\_training\_reducer(rbm\_layer\_id, local\_weights\_list):

global\_weights = aggregate\_weights(local\_weights\_list)

emit(rbm\_layer\_id, global\_weights)

**RESULTS AND DISCUSSION:**

Our proposed MapReduce-based big data classification model integrates feature subset selection and hyper-parameter tuned Deep Belief Networks (DBNs) to efficiently handle and classify large-scale datasets. By leveraging the MapReduce framework, we achieved significant feature dimensionality reduction, which improved computational efficiency and model performance. Hyper-parameter tuning via Grid Search and Bayesian Optimization further optimized the DBN, resulting in high classification accuracy. Experiments on benchmark datasets like MNIST and KDD Cup '99 demonstrated superior accuracy and reduced training time compared to traditional methods, validating the model's effectiveness in big data contexts. This approach offers a scalable and robust solution for complex classification tasks.

**CONCLUSION:**

In this study, we developed a MapReduce-based big data classification model that combines feature subset selection with a hyper-parameter tuned Deep Belief Network (DBN) to efficiently process and classify large-scale datasets. By leveraging the distributed computing power of the MapReduce framework, we achieved scalable and efficient data handling. Our model significantly reduces dataset dimensionality through feature selection, improving both efficiency and accuracy. The DBN, trained and fine-tuned in a distributed manner, further enhances classification performance. Experimental results demonstrate that our approach delivers superior accuracy and computational efficiency, providing a robust solution for big data classification challenges.

**REFERENCES:**

1. Gupta R, Gupta H, Mohania M. Cloud computing and big data analytics: what is new from databases perspective? In: *1st International Conference on Big Data Analytics* (*BDA*), New Delhi, India, 2012, 42–61.
2. Agrawal D, Das S, Abbadi AE. Big data and cloud computing: current state and future opportunities. In: *14th International Conference on Extending Database Technology* (*EDBT*), Uppsala, Sweden, 2011, 530–533.
3. Madden S. From databases to big data. *IEEE Internet Comput* 2012, **16**: 4–6.
4. Kraska T. Finding the needle in the big data systems haystack. *IEEE Internet Comput* 2013, **17**: 84–86.
5. Alpaydin E. *Introduction to Machine Learning*. 2nd ed. Cambridge, MA: MIT Press; 2010.
6. Witten IH, Frank E, Hall MA. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Series in Data Management Systems. Amsterdam: Morgan Kaufmann; 2011.
7. March ST, Hevner AR. Integrated decision support systems: a data warehousing perspective. *Decis Support Syst* 2007, **43**: 1031–1043.
8. Watson HJ, Wixom BH. The current state of business intelligence. *Computer* 2007, **40**: 96–99.
9. Provost F, Fawcett T. *Data Science for Business. What You Need to Know about Data Mining and Data-Analytic Thinking*. 1st ed. Sebastopol, CA: O'Reilly; 2013.
10. Mell P, Grance T. The NIST definition of cloud computing (draft) recommendations of the national institute of standards and technology. *NIST Spec Publ* 2011, **145**.