Theory and Practice of Science and Technology
2025, VOL. 6, NO. 5, 49-53
DOI: 10.47297/taposatWSP2633-456911.20250605

Optimizing Random Forest with Apache Spark: A Survey on Distributed Machine Learning and Big Data Scalability

Zixiang Yan

York University, Faculty of Science, Toronto, Ontario, Canada, YO10 5DD

ABSTRACT
Current trends have contributed to the rapid development of Big Data Processing Technologies. As one of the first variants of machine learning algorithms, random forests have exceptional classification and elaboration characteristics. The field of application in many applications has gradually expanded, and traditional random forests have been threatened, especially in large-scale data set processing. The scope of application is limited, and high computational costs and long training time are especially important. As a distributed computing-based platform in RAM, Apache Spark can optimize the learning speed and cost of developing random noise algorithms in a big data environment. This article discusses the random noise optimization strategy based on Apache Spark.it not only deals with parallel computing, but also discusses the techniques of object selection optimization and model improvement. At the same time, it outlines the latest achievements in the field of random forest optimization based on Spark and considers possible directions of its further development.

KEYWORDS
Random forest; distributed computing; big data processing; feature selection; application implementation analysis

1 Introduction

1.1 Research Background

The rapid development of artificial intelligence and data-based algorithms has gradually made machine learning one of the important tools for data analysis. Due to its high training efficiency and significant resistance to retraining, Random Forest is widely used in classification, regression and feature selection tasks. With continuous data dissemination, traditional random forest algorithms have gradually attracted particular attention from researchers due to their low computational efficiency, high memory consumption, and long training times in processing large datasets (Yang et al.2007)., 2021). A distributed random forest has been developed to solve these problems. special attention was paid to Apache Spark-based training techniques.

As a distributed computing platform, Apache Spark can significantly improve data processing efficiency and support parallel task execution. The optimized machine learning algorithm provided by Spark MLlib enables efficient execution of random forest algorithms in a distributed environment (Bay et al. 2007)., 2021). Currently, Spark-based optimization techniques are widely used for financial risk control, medical data analysis, network security, and traffic prediction.

1.2 Research Significance

This paper discusses the Apache Spark integration mechanism to improve computational efficiency and performance of random forest algorithm models. In the process of parallel random forest training, Spark analyzes the efficiency of function selection processing in terms of task planning and memory optimization. At the same time, it studies the optimized performance of models in different applications, and the results of calculations and application characteristics in real scenarios are generalized based on the analysis of the optimization process.

2 Overview of Apache Spark and the Random Forest Algorithm

As a distributed data processing platform, Apache Spark uses a computer model in RAM as its primary function. Compared to the traditional MapReduce platform, Spark has significant advantages in terms of computational efficiency and iterative processing power. It supports multiple programming languages and contains components such as Spark SQL, Spark Streaming, and MLlib, which are quite scalable and flexible. Spark uses a highly flexible distributed data processing mechanism, error tolerance, planning and hierarchical modeling support to ensure reliability of computation while processing high performance data. It is widely used in areas such as big data analysis, machine learning, and real-time computing.

* **Corresponding Author:** W_illi_AM@163.com

Random forest, based on integrated machine learning algorithms, builds multiple decision trees and aggregates the output. Improved performance of the prediction model. The self-service sampling method takes over a subset of the training data for each decision tree and randomly selects a subset of objects in the separation node to determine the best Separation Point. The variety and reliability of the model increases. Avoiding retraining and maintaining high accuracy in complex environments, Random Forest is re-used in classification and regression problems, and its stability and extensibility, especially in Big Data Processing, allow for widespread use in financial and other areas.

3 Optimization Methods for Random Forest Based on Spark

3.1 Parallel Computation Optimization

In the context of big data, the traditional machine learning process is often limited by performance bottlenecks and limited resources, and the efficiency of model training becomes low. To combat this problem, Apache Spark implements parallel task execution using an elastic data distribution mechanism a fault-tolerant scheduling system. Random forest training tasks can be distributed for simultaneous execution to multiple computing nodes, which increases computation efficiency.

Spark effectively divides training data into subsets and distributes it to each model training node. This strategy significantly improves the efficiency of tasks, and the computational load on an individual node becomes less.

Compared to the traditional MapReduce model, in which disk I / O operations are often performed, a computer mechanism is implemented in Spark to work with memory, and intermediate results can be directly cached and reused in memory. In random forest training, this function is of great importance. The directing costs of processing objects, partitioning tree nodes and aggregation of models are reduced, unnecessary read and write operations from the disk are eliminated, and the overall learning speed and computing efficiency are increased.

3.2 Feature Selection Optimization

In the process of training random forest models, the choice of characteristics has a significant impact on the performance of the model and the efficiency of computation. The space features of high dimensionality increases the complexity of computation, and can also lead to noise or irrelevant information, which will weaken the model's ability to generalize. Optimization of feature selection can reduce computational redundancy and improve Model generalization.

Practical applications to assess the contribution of characteristics to the target variable typically use the information gain and Gini index (Hu, 2019). As one of the methods for optimizing the choice of characteristics, the learning process calculates the gain of information or Gini index for each trait in order to detect the most representative traits and eliminate unnecessary ones. Principal Component Analysis (PCA) is a classical method of linear dimensionality reduction that, by decomposing a Covariant matrix of variance by eigenvalues, converts the original feature space into a low-dimensionality subspace, records the main direction of data change, and reduces the redundancy of features.

3.3 Deep Random Forest Optimization

The rapid development of deep learning has contributed to the study of the combination of traditional machine learning algorithms and deep architecture. It aims to improve the characteristics and possibilities of generalization of the model. A hybrid model named Deep Random Forest appeared. It combines the advantages of traditional random forest in the field of classification and regression with the possibilities of deep feature extraction. The model uses a hierarchical training structure, extracts multi-layered data features through hierarchical learning, and integrates the output of multiple high-level decision trees to produce more accurate predictions.

The bottom layers fixes coarse-grained features such as boundaries and frequency, and the deeper layers learn more abstract semantic representations by merging features. On this basis, Generative Adversarial Networks (GANs) can be applied to synthesize new feature data, and the training sampling space becomes more extensive so that the model can improve learning ability based on limited or unbalanced data sets, and the use of discriminators increases the model's ability to distinguish between real and synthetic samples not only improves classification accuracy, but also further improves robustness.

3.4 Computational Complexity Optimization

In conditions of rapid growth of data volumes, management of computational complexity is especially important to ensure the performance of the model and its practical application. In environments with multidimensional data used in traditional random forest models, training bottlenecks are often obvious. Problems such as excessive training time and

memory consumption are solved in parallel. To solve these problems, Spark platform provides optimization strategies that can conditionally be divided into three categories: parallel decision tree construction, incremental learning, and efficient storage optimization. Parallel decision tree structure allocates learning tasks between multiple computing nodes, thus significantly speeding up the overall model training process, especially in scenarios where large aggregated data is processed. This is greatly improved. Incremental learning adapts to new data by dynamically updating the model, eliminating the need to over qualify the entire model from zero, reducing the cost of overqualification, and optimizing the adaptability of the data or data in real time. In ultra - high-dimensional data scenarios, storage optimization strategies are particularly important.

4 Application Case Analysis

After the integration of the Random Forest and spark algorithms, it gained widespread application in many industries, such as the detection of telecommunications scams. Each application scenario will be detailed below. The telecommunication fraud detection, the traffic flow prediction, analysis of medical data, and flood monitoring.

4.1 Telecom Fraud Detection

After the popularization of telecommunications networks, the problem of telecommunications fraud becomes more serious. Traditional methods of detecting fake SMS messages make it difficult to combat such phenomena under conditions of rapidly increasing number and complexity of SMS messages, especially with regard to classification of characteristics and accuracy. The problem of detection is gradually becoming more serious. Young et al. (2019) proposed a random forest model based on Spark. The spatial subspace separation method is used to detect SMS scams, and the parallel computation of Spark is used as the basis for classifying SMS messages. Combined with feature selection techniques to improve detection accuracy, this technique effectively reduces the severity of detection problems.

Spark computing framework handles all SMS data in parallel, reduces computational costs, improves detection speed. By applying a spatial partitioning strategy, key user behavior characteristics in spatial separation strategies, improves model accuracy, builds each decision tree in a separate section, and improves Model generalization and reliability.

The Model classifies custom SMS data at multiple characteristic levels, such as communication frequency and content mode, estimates their impact on prediction, assigns training tasks using Spark'S RDD mechanism, dynamically adjusts the weight of decision trees to improve classification efficiency, and studies the use of operator data sets containing millions of SMS records. The data contains call behavior and user attributes and uses PCAS to reduce dimensionality and remove redundant functions, thus improving classification efficiency. Optimized classification accuracy in the spark Random Forest model has been significantly improved. Compared to the traditional model, learning time in Big Data Processing decreases by 50%, and the recognition rate of high-risk users increases by 20% after weighted optimization (Yang et al.2007)., 2021). Optimization of the training model not only makes the time-sharing characteristics of the classification process readable, but weighted recognition also increases the level of accuracy of the high-risk data distribution across the data series.

4.2 Traffic Flow Prediction

In modern traffic data analysis and processing, computational efficiency has a significant impact on prediction results. China's economy is developing rapidly, the number of vehicle owners in the family is growing every year, and the problem of urban traffic jams has become quite serious. Traditional prediction methods, such as time series and Markov models, cannot meet current needs. The combination of Random Forest and Spark became an effective solution (Bai et al., 2021).

In this study, RDD technology is used for distributed data storage and computation to speed up the task planning process. Spark's MapReduce platform is used for parallel construction of decision tree models. Training tasks can be distributed to multiple computer nodes to be performed, which increases the overall learning rate. Load balancing strategies are used to achieve a balanced distribution of tasks, optimize computational efficiency, and traffic congestion is affected by many complexities. This study presents Yin-Yang Pairwise Optimization (YYPO), which uses metaheuristic search to adjust hyperparameters to optimize the classification model, and at the same time, techniques of selecting features such as information retrieval and Gini index, reduce the complexity of the model and computational costs, which allows achieving right is significantly improved total computational efficiency.

This study processed traffic data in the Hefei demonstration district. The data included 60,000 microwave records and 360,000 GPS records. The study compared four models: K-nearest neighbors, LSTM networks, traditional random forest, and an optimized version of random forest (YYPORF). The experiments showed that the results of the YYPORF test were the best in all indicators: accuracy was 95.58%, reliability was 95.78% and memory speed was 95.60%. F1 reached 0.956

(Bai et al.2007)., 2021).

4.3 Medical Data Analysis

Diabetes mellitus is a common chronic disease worldwide. Early diagnosis is closely related to the patient's long-term health condition. With improved health, the need for accurate prediction of the disease gradually increases. Traditional statistical models face difficulties in processing large amounts of encrypted and unbalanced data, so model optimization has become a relevant task. Yang et al. (2021) proposed a method that combines random forest and spark, resulting in increased computational efficiency and prediction accuracy. The characteristics of medical data are usually expressed in high capacity, dimension and diversity. The RDD mechanism in Spark allows for greater efficiency of data storage and management. The calculation process is optimized based on Dag-base task planning, which improves the performance of the model in big data analysis. For disease prediction and classification tasks, the MapReduce model can split data into multiple parts, providing parallel learning of computational nodes to improve efficiency, and Spark Streaming supports real-time data processing to ensure timely disease prediction and abnormality detection. This study used the diabetes dataset of the National Institute of Diabetes and Digestive and Kidney Diseases (NIDDK) as the subject of the study. The data set contains various health indicators, such as glucose concentration and insulin level. After the normalization of the data, the missing values were processed and the distribution by categories was balanced. Methods such as the K-neighbor algorithm and spark optimized random forest model have been compared. The results show that the average prediction accuracy of random forest optimized by Spark reached 81.13%, which is better than other models. Training time in the traditional single-threaded method was greatly extended, and parallel spark computing reduced training time by more than 40%. (Yang et al., 2021).

4.4 Flood Disaster Detection

Natural disasters seriously affect people's economies and living standards, and frequent floods pose a serious threat to public safety. Improving the speed and accuracy of flood detection is a significant contribution to improving the management of the city's Disaster Relief Services. Meng et al. (2024) proposed a solution that combines a random forest algorithm with a spatial analysis model. While the accuracy of the prediction increases, the calculation time also decreases significantly. The RDD spark mechanism reduces frequent read and write operations from disk, improves resource usage, and makes memory storage and data management efficient. In the training of the random forest model, load balancing under the parallel spark execution strategy evenly distributes tasks among computing nodes, reduces computational costs, and allows to obtain many types of data associated with floods, such as precipitation. This study combines optimized spark Random Forest and spatiotemporal analysis to construct a dynamic estimation model that can accurately analyze the impact of floods in the spatiotemporal dimension, and dynamically adjusting the number and depth of Decision Trees improves the ability to generalize models in different urban environments. The subject of this study is long-term data on floods and disasters in 11 parts of the city. The report includes data such as rainfall, altitude, information on land use and infrastructure. To evaluate the efficiency of the model, this study compared the support vector machines, long short-term memory (LSTM) networks, traditional random forest, and Spark-optimized random forest. The results show that the optimized model has the highest accuracy rate reaching 88.74%, and training time is reduced by more than 50%. As for short-term flood prediction, the F1 values for random forests optimized with LSTM and spark in the work of, they reached 92.1% and 91.3% respectively, while F1 values for the traditional model were only 35.07% and 32.3% respectively (Meng et al,. 2024).

5 Future Research Directions

While Apache Spark's integration has allowed significant progress in modeling and training random forests based on large-scale data, current research continues to face certain limitations. The following directions of future research can be highlighted:

5.1 Integrating Reinforcement Learning to Optimize Random Forest Training Strategies

Currently, random forest optimization is mainly based on traditional parameter tuning and data partitioning strategies. By incorporating reinforcement training as an intelligent optimization method, key learning parameters can be dynamically adjusted, including the number and maximum depth of trees and feature selection methods. The ability to generalize the model is improved. For intelligent search, supporting technologies such as customizable parameter setting and dynamic hyperparameter setting, deep X networks, and online learning models are used to support self-optimization

and improve Model Reliability.

5.2 Applying Spark-Based Random Forest to Multimodal Data Fusion

Multimodal data, such as text, images, audio, etc., are different from traditional structured data. They are more often used in practical applications and more difficult to process. Using cross-modal feature extraction techniques combined with deep feature selection and distributed computing capabilities of Spark, multimodal information can be distributed by levels and integrated into unified modeling to achieve efficient integration of heterogeneous data, and the precision and computational efficiency of analysis tasks large scale is in the process of improvement.

5.3 Combining Graph Computing with Random Forests for Network Analysis

In the field of social networks, detection of financial fraud and modeling of the behavior of telecommunications systems, the universality of data with a graphical structure is gradually increasing. Spark's GraphX module plays an important role in distributed computing support of big graphic data. Future studies may combine graphical computation with random forest models to create a system that will have both local decision-making capabilities and structural-topological features. Graph neural networks (GNNs) can be processed in node representation learning, and random forest can also be applied to the process of building predictive and classification problems such as node classification, link prediction, and community detection.

5.4 Addressing the Increased Complexity Introduced by Spark

Although spark significantly improved the extensibility and efficiency of random forest model training, the introduction of distributed architecture gradually increased the new complexity, increased the cost of communication and memory use, and unbalanced resource planning. Future studies should focus on ultra-scaled data processing environments to control and optimize these computational costs.

Specific directions include the development of lightweight Random Forest architecture, the development of adaptive resource allocation and task planning strategies, the construction of automated learning systems and logical justification to reduce manual intervention, inference to reduce manual intervention; and integrating edge computing and federated learning to alleviate centralized deployment bottlenecks. These methods improve and optimize the actual availability of random forest. models and engineering adaptability.

6 Conclusion

The integration of Apache Spark and the random forest algorithm provides a more efficient path of optimization for various industry applications. By combining distributed computing, feature selection, and deep learning technology, the Random Forest model optimized with Spark demonstrated high competitiveness in the rapidly expanding Big Data Processing Environment.

The amount of data and computing needs continues to grow. Reducing directing costs, simplifying model structure, and improving data interpretability remain key research tasks. Multimodal data fusion and further integration with graph computing are also expected to be focal points in future studies. With continued optimization of the spark computing platform, the range of Random Forest will expand, and more efficient solutions can be found for large-scale data analysis.

References

[1] Bai, X., Feng, Y., & Li, L. (2021). Optimized random forest prediction model for urban road congestion. *Science Technology and Engineering*, 21 (26), 11205–11211.

[2] Duan, W., & Tong, M. (2021). An improved random forest algorithm based on Spark. *Computer Applications and Software*, 38(8), 41–50.

[3] Hu, T. (2019). *Research on the optimization and parallelization of random forest algorithm based on Spark* (Master's thesis). Qilu University of Technology.

[4] Lei, C. (2021). *Research on parallel random forest algorithm based on Spark* (Master's thesis). Jiangxi University of Science and Technology.

[5] Meng, X., Liu, C., & Cao, Y. (2024). Research on spatiotemporal evolution of urban agglomeration flood disaster resilience based on optimized random forest algorithm. *Journal of Disaster Prevention and Mitigation Engineering*, 44(4), 784–792.

[6] Yang, J., Xu, J., Yue, Q., & Zeng, D. (2021). Research on SMS fraud user identification based on Spark and random forest. *Computer Engineering and Science*, 47(5), 23–31.

[7] Yang, Y., & Guo, J. (2021). Optimization of diabetes prediction based on Spark and random forest. *Electronics World*, 17, 200–201.