Scholar Publishing Group
Distributed Processing System
https://doi.org/10.38007/DPS.2022.030409
ISSN 2790-0916 Vol. 3, Issue 4: 70-77

# *Three-layer Distributed System Based on Bayesian Classifier*

**Ahthasha Khaneni***

*Commune d'Akanda, Gabon*

**corresponding author*

***Keywords:*** Bayesian Classifier, Distributed System, Semi-supervised Naive Bayesian Algorithm, System Research

***Abstract:*** Due to the rapid development of Internet technology and the rapid growth of the number of users, various data and information show an explosive growth trend. The analysis and processing of massive data is more and more important for the development of enterprises. To solve the problems of growing set and increasing system pressure, this paper proposes a distributed system with a three-tier architecture based on Bayesian classifier(BC). This paper first describes the module functions and system interface requirements of the three-layer distributed system(TLDS), followed by the design of the TLDS, and finally the system implementation. And the training time is analyzed, and it is found that the SNB algorithm improves the accuracy and shortens the training time.

## 1. Introduction

With the advent of the era of big data, classifying massive data has become a very important task. However, data classification under a single-machine architecture has been unable to utilize limited hardware resources to meet data analysis needs. Researchers try to use BC. The purpose of this research is to use the abundant hardware resources in the three-tier distributed system to improve the performance of data classification[1-2]. Due to the difference in hardware environment between traditional stand-alone systems and distributed systems, traditional data processing methods often fail to make full use of the considerable hardware resources in distributed systems, resulting in lower performance improvements and utilizing scalable hardware resources in distributed systems To improve the performance of data classification has become a research direction worth exploring[3-4].

At present, many researchers have discussed and studied Bayesian classifiers and three-layer distributed systems, and have achieved good results. For example, scholars such as Alem A proposed a solution aimed at eliminating the security problem of Hadoop clusters. The solution is

Copyright: © 2022 by the authors. This is an Open Access article distributed under the Creative Commons Attribution License (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited (https://creativecommons.org/licenses/by/4.0/).

based on secure remote cryptographic protocols, blockchain technology and threshold cryptography theory. Practical Byzantine Fault Tolerance (PBFT) is deployed as a consensus mechanism in On the blockchain, studies have shown that the proposed scheme outperforms many existing schemes in terms of computational overhead and storage requirements, and does not affect the security level of the system [5]. In order to overcome the problems of lack of accuracy and operability in data classification of existing technologies, researchers such as Seo J H have developed a new algorithm called MB Ferns and the R package to build a multi-branch decision tree and use the data obtained from the training data. The Nave Bayesian probability model, a classifier that centrally generates key features, proves that the proposed algorithm performs well for general classification problems and training data extraction operations [6]. Bayesian classifier has obvious advantages in data classification and processing. It can be seen that the study of three-layer distributed system based on Bayesian classifier has very important practical significance for data processing [7].

This paper studies the TLDS based on the BC. The structure of this paper can be divided into three parts: The first part is a brief introduction to the related concepts of the TLDS, including system modules and interfaces. The second part is the design of the three-tier distributed system. The system design part includes three parts, namely the task management module, the node load model and the block parallel scheduling strategy; the third part is the implementation of the three-tier distributed system, in this part, the accuracy and training time of SNB algorithm, INB algorithm and NB algorithm are analyzed.

## 2. Related Overview

### 2.1. System Module Function

The distributed system adopts a three-tier model with 4 modules. The functions of each module are:

(1) Logical Transaction (LT). LT is a processing center module for user requests. This module can assemble the SQL statement according to the received package sent from the Client Agent, and send the assembled SQL statement package to the next layer: CP or DA[8] through data segmentation technology. In order to improve the flexibility of system services, that is, to flexibly adjust the logical processing order of the system to user requests, the configuration file of LT uses XML format to describe the order and type of service operations [9]. In this way, it is only necessary to change the configuration file of the LT to change the logical processing sequence of user requests without changing the program code [10].

(2) Cache Pool (CP). The main function of this module is to improve the data read and write rate and reduce the access pressure to the database [11].

(3) Database Agent (DA). This module is an access proxy for the database [12]. Its role is to shield the underlying details of the DC, implement the read-write separation strategy, and also allocate requests to the DC through a certain load balancing strategy, with the function of flow control [13].

(4) Database Cluster (DC). A DC is a cluster with multiple databases, storing a large amount of data [14]. The number of DC in the entire system and the number of databases in each DC can vary according to specific circumstances [15]. The role of Client Agent (CA) in the figure is to be responsible for the user's access request [16]. And convert the user request package into the package format defined by DSAS. Then requests are distributed to lower-level modules through a certain load balancing strategy. The benefit of this is that it prevents LT from blocking due to processing complex data [16].

## 2.2. Interface Requirements

In a three-layer distributed system, if a cross-layer multi-protocol association data classification task is used, the original measurement system should be integrated and new functional modules should be developed, including the data scanning program executed in the background and the page display module in the foreground [17]. It enables users to expand data scanning tasks with new logic, including: automatic scanning task delivery page, alarm threshold setting page, and alarm information display page. These pages all use a relatively single color scheme, which makes the whole interface simple and refreshing. At the same time, the execution results of the tasks are marked in green and red, which makes the results display clearer, and the contrast between different results is stronger, which is convenient for users to use [18].

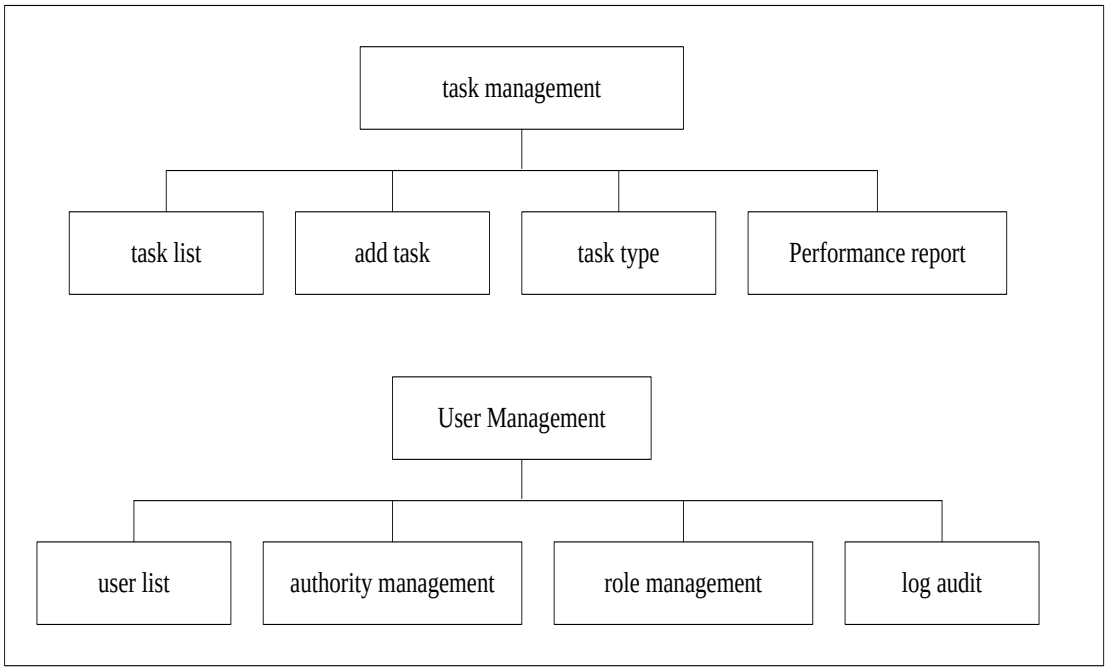


*Figure 1. System interface structure diagram*

## 3. System Design

### 3.1. Task Management Module

The task management module is used to manage the data scanning tasks issued by the user and the returned scanning task results, and mainly includes a task scheduling sub-module, a task issuing sub-module, and a task result accessing sub-module. Among them, the task scheduling sub-module is used to execute corresponding tasks regularly or periodically according to the parameters provided by the user; the task release sub-module is used to convey the data scanning task to the corresponding node; the task access sub-module is used to access task information and task results, and returns the above information according to certain filter conditions when the user needs it.

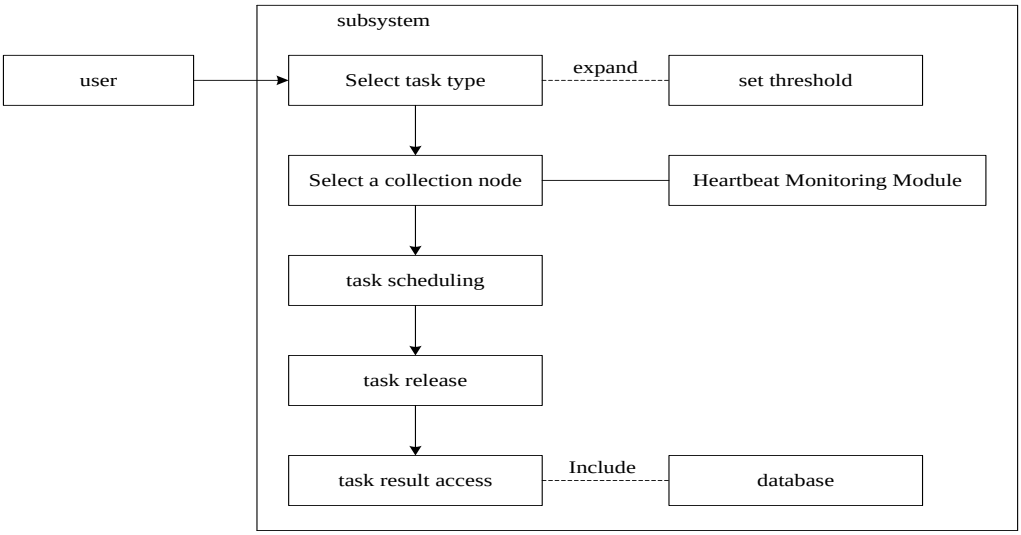


*Figure 2. Task module structure diagram*

## 3.2. Node Load Model

This section will build a reasonable data scan load model for each data storage node based on the total scan workload for that node. Suppose a distributed system uses the $K_i$ replica mechanism to store all data shards on n nodes ($k \leqslant n$). Then for the scan subtask for the data slice Pi, it can be further divided into scan subtasks for k replicas. Among them, the division ratio of scanning subtasks for k replicas should satisfy the following constraint formula:

$$\sum_{h=0}^{n-1} Y_{i,h} = 1, \quad w.r.t \begin{cases} 0 < Y_{i,h} \leq 1; \text{Shard } P_i \text{has a replica on node } R_h \\ Y_{i,h} = 0; \text{Shard } P_i \text{ has no replica on node } R_h \end{cases} \tag{1}$$

Specifically, $Y_{i,h}$ indicates the proportion of the scanning workload that the replica of the data shard $P_i$ on the storage node $R_h$ should undertake. For a storage node $S_h$ with $m_h$ replicas, the overall scanning workload $S_h$ can be expressed by the following formula:

$$S_h = \sum_{i=0}^{m_h - 1} Y_{i,h} \times T_i \tag{2}$$

## 3.3. Block Parallel Scheduling Strategy

The block parallel scheduling strategy fully considers the load balancing between data classification and multi-threading and the subtask switching overhead within a single thread, and improves the parallel processing capability of multiple scanning subtasks in a single storage node. The core idea is that in the block parallel scheduling strategy, each scan subtask will be divided into multiple block subtasks of the same size as independent scheduling units for each data classification task. For example, the scan subtasks for data slices P5 and P4 in FIG. 3 are divided into 4 and 2 block subtasks of the same size, respectively. From the perspective of subtask switching overhead, each data classification thread in the block parallel scheduling strategy should schedule continuous block subtasks as much as possible, so as to avoid the impact of subtask switching overhead as

much as possible. On the other hand, when there is load imbalance among data classification threads, the data classification with higher load in the block parallel scheduling strategy will migrate some block subtasks to the data classification with lower load, so as to ensure that different data Classification threads have similar scan loads. In general, the core idea of the block parallel scheduling strategy is to ensure a balanced scan load and low subtask switching overhead in the multi-threaded parallel scanning process.

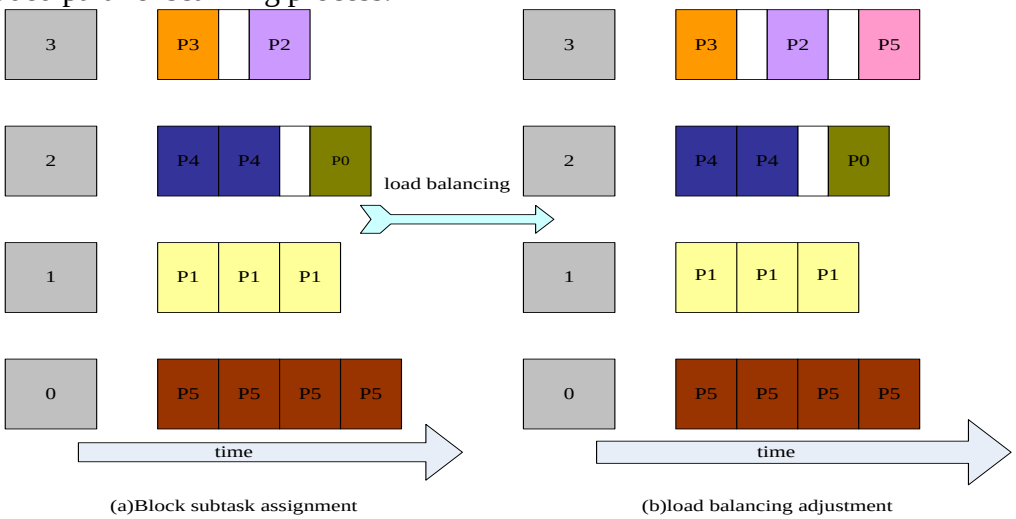


*Figure 3. Block parallel scheduling scheme*

## 4. System Implementation

### 4.1. Accuracy Analysis

The advantage of the semi-supervised naive Bayes algorithm is that it is very simple to apply, has no particularly complicated logic and has good classification performance.

The experimental part first describes the experimental data set, and then designs the experiment according to the algorithm steps. In the experiment, 8 data sets are selected from the database to detect the Naive Bayes algorithm (NB) and the semi-supervised Naive Bayes algorithm (SNB). and the improved Naive Bayes classification algorithm (INB) for the classification performance of these three classifiers.

Select data sets 1 to 4 to analyze the accuracy, and the obtained accuracy results are shown in Table 1.

*Table 1. Correctness table of various algorithms*

| Data set | NB(%) | INB(%) | SNB(%) |
|---|---|---|---|
| Data set 1 | 70.27 | 73.24 | 79.89 |
| Data set 2 | 81.32 | 83.45 | 86.78 |
| Data set 3 | 82.48 | 85.62 | 90.67 |
| Data set 4 | 90.23 | 94.83 | 98.71 |

Analysis of the data in Table 1 shows that the three algorithms of NB, INB and SNB have the highest accuracy in data set 4, which are 90.23%, 94.83% and 98.71% respectively, and the

accuracy of the three algorithms varies with The increase of the data set increases, which shows that the classification performance of the NB, INB and SNB algorithms are improved and depend on the labeled training data. The more labeled training data, the better the performance of these classifiers; from this experiment It can be seen that the accuracy of the SNB algorithm and the INB algorithm is higher than that of the NB algorithm in most cases, indicating that the use of unlabeled data can improve the performance of the classifier, but among the three algorithms The accuracy of SNB is always the highest, which shows that SNB improves the accuracy of classification.

### 4.2. Analysis of Training Time

Data sets 5 to 8 are used as test data sets in the training time analysis, because these four data sets reflect the characteristics of massive data. First we will test how long it takes for SNB to run on the cluster. In the experiment, the 16 nodes are divided into eight groups, and then the data processing capabilities of the clusters composed of 2, 4, 6, 8, 10, 12, 14 and 16 nodes are tested respectively. From the running time, the speedup ratio is calculated. Next we will test the time it takes for SNB to train on these four datasets. The training time of SNB and NB is shown in Table 2, and the schematic diagram of the acceleration ratio of SNB algorithm is shown in Figure 4.

*Table 2. Algorithm training time table*

| Data set | Data set size (G) | NB training time(s) | SNB training time(s) | Number of nodes |
|---|---|---|---|---|
| Data set 5 | 1.2 | 51278 | 48630 | 6 |
| Data set 6 | 5.4 | 246319 | 217694 | 2 |
| Data set 7 | 10.6 | 476581 | 347068 | 2 |
| Data set 8 | 20.4 | 816472 | 706549 | 2 |

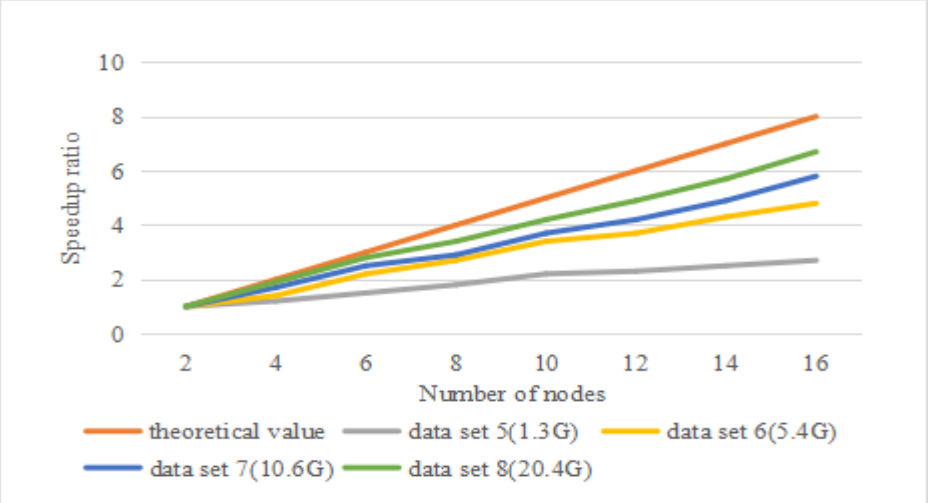


*Figure 4. Speedup versus number of nodes*

It can be seen from Table 2 that the training time of the SNB algorithm in the four data sets is shorter than that of the NB algorithm, which shows that the SNB algorithm can shorten the data training time. It can be seen from Figure 4 that the speedup ratio of the data set increases with the increase of the number of nodes, which shows that the SNB algorithm has good scalability and is suitable for large-scale data sets. Through the experimental analysis of the SNB algorithm, it is found that the algorithm improves the accuracy and shortens the training time.

## 5. Conclusion

BC is widely used in data classification, so this paper designs and studies a TLDS based on BC. This paper firstly introduces the relevant concepts, then designs the system, and finally the system implementation part. In this paper, through the experimental comparison and analysis of the three algorithms of SNB, INB and NB, it is found that the SNB algorithm has the highest accuracy and the shortest training time among the three algorithms, so the SNB algorithm has good performance and good scalability. Large scale datasets. This paper studies the TLDS based on BC and draws conclusions, but there are still many shortcomings that need to be improved. The TLDS based on BC is a direction worthy of in-depth research, which has a positive effect on data classification and processing.

## Funding

This article is not supported by any foundation.

## Data Availability

Data sharing is not applicable to this article as no new data were created or analysed in this study.

## Conflict of Interest

The author states that this article has no conflict of interest.

## References

*[1] Alotaibi A, Hamdaoui R. Improvement of Semi-Supervised Document Classification based on Fine Tuning Naive Bayesian Classifier. European Journal of Scientific Research, 2021, 158(3):181-190.*

*[2] A. A. Yahya, Levin V M. Bayesian classifier is the tool of increasing the efficiency of defects recognition in power transformers. Power engineering research equipment technology, 2020, 21(6):11-18.*

*[3] Park D H, Chen S, Kim T W. Probabilistic assessment of drought states using a dynamic naive Bayesian classifier. Terrestrial Atmospheric and Oceanic Sciences, 2020, 31(3):359-368.*

*[4] Hadiwandra T Y. Perbandingan Kinerja Model Klasifikasi Decission Tree, Bayesian Classifier, Instance Base, Linear Function Base, Rule Base pada 4 Dataset Berbeda. SATIN - Sains dan Teknologi Informasi, 2019, 5(1):70-78.*

*[5] Alem A, Dahmani Y, Bendaoud B. Skyline Computation for Improving Nave Bayesian Classifier in Intrusion Detection System. Ing énierie des Syst èmes D Information, 2019, 24(5):513-518.*

*[6] Seo J H, Lee J Y. Novel nomogram based on risk factors of chronic obstructive pulmonary disease (COPD) using a nave Bayesian classifier model. Journal of the Korean Statistical Society, 2019, 48( 2):278-286.*

*[7] Sudeep, Tanwar, Jayneel, et al. Human Arthritis Analysis in Fog Computing Environment Using Bayesian Network Classifier and Thread Protocol. IEEE Consumer Electronics Magazine, 2020, 9(1):88-94.*

*[8] Akhtar N, Mian A. Nonparametric Coupled Bayesian Dictionary and Classifier Learning for Hyperspectral Classification. Neural Networks & Learning Systems IEEE Transactions on, 2018, 29(9):4038-4050.*

*[9] Obi M, Nwauzi K K, Akhetuamen S. Tuberculosis Diagnosis Using Nave Bayes Classifier. International Journal of Scientific and Engineering Research, 2021, 10(11):1367-1372.*

*[10] Rao D M, Higiro J D. Managing Pending Events in Sequential and Parallel Simulations Using Three-tier Heap and Two-tier Ladder Queue. ACM Transactions on Modeling and Computer Simulation, 2019, 29(2):1-28.*

*[11] Andrew, Bate. Bayesian confidence propagation neural network. Drug safety, 2018, 30(7):623-625.*

*[12] Merkle E C, Rosseel Y. blavaan: Bayesian structural equation models via parameter expansion. Stats, 2018, 58(6):129-138.*

*[13] Cottet V, Alquier P. 1-Bit matrix completion: PAC-Bayesian analysis of a variational approximation. Machine Learning, 2018, 107(3):579-603.*

*[14] Karbalayghareh A, Qian X, Dougherty E R. Optimal Bayesian Transfer Learning. IEEE Transactions on Signal Processing, 2018, 66(14):3724-3739.*

*[15] Hu G X, Kuipers D R, Zeng Y. Bayesian Inference via Filtering Equations for Ultrahigh Frequency Data (I): Model and Estimation. Siam/asa Journal on Uncertainty Quantification, 2018, 6(1):34-60.*

*[16] Mckinley T J, Vernon I, Andrianakis I, et al. Approximate Bayesian Computation and simulation-based inference for complex stochastic epidemic models. Statistical science, 2018, 33(1):4-18.*

*[17] Burke D L, Bujkiewicz S, Riley R D. Bayesian bivariate meta-analysis of correlated effects: Impact of the prior distributions on the between-study correlation, borrowing of strength, and joint inferences. Statistical Methods in Medical Research, 2018, 27(2):428-450.*

*[18] Ranacher P, Stéphane Joost, Bachmann S, et al. Linguistic traits as heritable units? Spatial Bayesian clustering reveals Swiss German dialect regions. Journal of Linguistic Geography, 2021, 10(1):11-22.*