Hindawi
Mobile Information Systems
Volume 2022, Article ID 2958176, 12 pages
https://doi.org/10.1155/2022/2958176

Research Article

Simulation Analysis of Artificial Intelligence in Enterprise Financial Management Based on Parallel Computing

Zhu Feng

School of Accounting, Zhongnan University of Economics and Law, Hubei, Wuhan 430073, China

Correspondence should be addressed to Zhu Feng; z0003893@zuel.edu.cn

Received 15 August 2022; Revised 18 September 2022; Accepted 30 September 2022; Published 10 October 2022

Academic Editor: Shadi Aljawarneh

Under the background of today's society, people's daily life and production are inseparable from the use of information technology. With the rapid development of Internet technology and national information industry, more and more small and medium-sized enterprises choose to apply AI financial information management system. Based on this background, this paper introduces the principle of parallel computing and applies it to the financial management of enterprises together with artificial intelligence technology. Firstly, the paper discusses the model and form of the basic technology, then optimizes it, studies the actual financial management needs of small and medium-sized enterprises, completes the design and implementation of the system structure, and conducts simulation experiments to test the effectiveness of the system's functional performance. It can be seen from the test results that the system meets the design goals and has basic functions, but some details and contents need to be improved and supplemented. Therefore, we must strive to ensure the accuracy, preciseness, and reliability of financial management data. When more data are needed, faster and better data processing based on user needs and feedback is required. This paper delves into artificial intelligence technology and the idea of parallel computing. This paper applies it to the field of enterprise financial management to create an effective management system.

1. Introduction

The development of Chinese enterprises has ushered in a new era. First of all, China has become the fastest growing region in the world today. China's optical fiber data transmission and update volume is at the forefront of the world, and Chinese enterprises have entered the information age. Secondly, China's logistics technology has developed rapidly. Today's logistics and transportation volume is at the forefront of the world, and the logistics supply chain of enterprises has undergone great changes. Finally, China's mobile payment is in a leading position. Globally, China has become the most convenient mobile payment country for online and offline transactions in the world. Therefore, in order to take advantage of China's rapidly developing technologies, Chinese enterprise management will also face many pressures and challenges. In today's more important modern financial management field, the role of financial management is self-evident, and the quality of financial management system affects the overall quality of enterprise management [1]. Accordingly, optimizing the financial management system is the main goal of development in this field [2]. The traditional form of manual bookkeeping has been unable to meet the actual needs under the background of the gradual development of the enterprise business, and the main function of the financial management system is to use excellent computerized bookkeeping to replace the traditional inefficient manual bookkeeping [3]. The Chinese finance department emphasized that computerization of accounting is the main direction for the development of accounting activities in various fields in China in the future [4]. Institutions and state-owned enterprises need to realize the transformation of computer accounting and bookkeeping as soon as possible and follow the principle of step by step in the process [5]. However, due to the large number of small and medium-sized enterprises in China, the fields and business contents involved are different. Small and medium-sized enterprises conform to the trend of information technology development and realize optimization and reform for traditional operation methods. It hopes to use network technology to realize the transformation of information management, so as to make itself develop better and faster. For businesses, the first step in managing information is financial management. Through information-based financial management, managers can plan, control, and further manage financial activities. Therefore, we need to use information technology to design a better and more practical financial management system to promote the healthy development of the enterprise and the entire enterprise financial system. In view of the current needs and development direction of small and medium-sized enterprises, this paper combines parallel computing and artificial intelligence to design a financial management system that can effectively meet the actual needs of small and medium-sized enterprises, so as to help small and medium-sized enterprises carry out more effective financial supervision and management, thus improving the development speed of enterprises and strengthening management [6].

2. Related Work

The literature believes that the ultimate goal of enterprise development is not only to pursue short-term profit maximization but also to achieve the goal of enterprise value maximization and wealth appreciation. At present, many countries are committed to improving the efficiency and level of the use of enterprise funds and improving the business activities of enterprises as much as possible, so as to maximize the output of enterprise investment [7]. After a lot of investigation and practice by scientific researchers, the modern financial management system is different from the previous accounting-based accounting management system and expands other business operations on the basis of the traditional financial management system [8]. In order to effectively combine the actual business process and operation, the management and interaction of data resources can be completed. In the context of the in-depth development of information technology, the financial management system and the business information systems of other industries are more coordinated and integrated [9]. The literature studies the relevant market and shows that computer management software usually consists of financial system, distribution system, production system, and decision support system, which is a highly integrated system. Each subsystem can work in coordination or independently [10]. When the subsystems are in cooperative operation mode, only a small amount of data needs to be input, and the entire system can exchange information. In this way, a complete scheme can be provided for the implementation of enterprise decision making [11]. The enterprise management information system is an organic combination of enterprise management, production management, accounting, and financial management [12]. The literature studies and analyzes the operation of the enterprise financial management system, discusses the problems and reasons that affect the normal operation of the financial management system, and, on this premise, designs an effective management control for the financial management system [13]. The business risk system adjusts the system operation mechanism, realizes business process reengineering, strengthens the effectiveness of the original system, and establishes a dynamic management platform for financial management that supports sustainable business development [14]. The literature shows that traditional financial processes have been unable to keep pace with the evolution of AI-powered financial management system capabilities. In response to this problem, the article conducts a preliminary study on the financial operation of artificial intelligence in business and summarizes some suggestions for guiding the reform of business financial processes [15, 16].

3. Algorithm Design of Parallel Computing

3.1. Basic Model of Existing Parallel Algorithms

3.1.1. DOT Model. The DOT model describes the implementation behavior of big data workloads in the form of arrays. The DOT model includes basic DOT blocks, combined DOT blocks, and DOT expressions. It can be described as

$$\begin{aligned}\overrightarrow{\mathrm{DJOT}} &= [D_1 \cdots D_n]\begin{bmatrix} o_1 \\ \vdots \\ o_n \end{bmatrix}[t] = \left[\bigcup_{i=1}^{n}(o_i(D_i))\right][t], \\ &= [t(o_1(D_1), \cdots, o_n(D_n))].\end{aligned} \tag{1}$$

Since the translation layer of the basic DOT block has only one node, the data it can handle are very limited. Therefore, several independent basic DOT blocks are combined into a unified DOT block. The formal description is

$$\begin{aligned}\overrightarrow{\mathrm{DOT}} &= [D_1 \cdots D_n] \\ &\quad \begin{bmatrix} o_{1,1} & o_{1,2} & \cdots & o_{1,m} \\ o_{2,1} & o_{2,2} & \cdots & o_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ o_{n,1} & o_{n,2} & \cdots & o_{n,m} \end{bmatrix}\begin{bmatrix} t_1 & 0 & \cdots & 0 \\ 0 & t_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & t_m \end{bmatrix}, \\ &= \left[\bigcup_{i=1}^{n}\left(o_{i,1}(D_i)\right) \cdots \bigcup_{i=1}^{n}\left(o_{i,m}(D_i)\right)\right]\begin{bmatrix} t_1 & 0 & \cdots & 0 \\ 0 & t_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & t_m \end{bmatrix}, \\ &= \left[t_1\left(o_{1,1}(D_1) \cdots o_{n,1}(D_n)\right)\right. \\ &\qquad \left. \cdots t_m\left(o_{1,m}(D_1) \cdots o_{n,m}(D_n)\right)\right].\end{aligned} \tag{2}$$

A DOT expression composed of multiple basic DOT blocks or combined DOT blocks can be used to describe the data flow of a large data payload.

It can be seen from the formal definition and description of the four layers: the operation layer and the aggregation layer only contain computing tasks, and their corresponding

0 and A matrices are both diagonal matrices, indicating that separate calculations can be performed. Independent concurrency: The communication task completed by the transport layer, the corresponding matrix T is a regular matrix, indicating that the communication must interact with each other.

In a concurrent storage system, a single miss does not necessarily cause a CPU hang, only pure errors do. C-AMAT is characterized by five parameters: C_H represents storage request access concurrency, Cm represents storage request pure miss concurrency, H represents storage request access time, pMR (PureMissRate) represents storage request pure miss rate, and pAMP (PureAverageMissPenalty) represents Pure average loss for storage requests. Through the derivation of a series of formulas, its formal description is as follows:

$$C - \mathrm{AMAT} = \frac{H}{C_H} + \mathrm{pMR} \times \frac{\mathrm{pAMP}}{C_M}. \tag{3}$$

Hit Concurrency Detector (HitConcurrency Detector, HCD) counts all hit cycles and records the status of each hit stage, calculates the CH hit concurrency of storage requests, and informs Miss. Concurrency Detector (MCD) of the current cycle if it occurs A hit occurs; MCD records the state of the pAMP cost of each pure loss cycle by counting the number of pure loss cycles, and calculates the pure loss concurrent Cm, the pure loss rate pMR, and the pure average loss storage requirement.

3.2. Parallel Computing Model Design.

The p-DOT model consists of a series of iterations, the "p-phase DOT model." In each stage q, the p-DOT model consists of three layers, as shown in Figure 1.

D layer (data layer): in a distributed system, datasets (D1 to Dn) are distributed and stored in n data nodes.

O layer (computing layer): in phase q, nodes (01 to ong) perform independent simultaneous computations, and each O node only processes the corresponding data (including input data or intermediate data) and stores intermediate results.

T layer (communication layer): in phase q $(q \neq p)$, each communication operator $t_{i,j}$ performs point-to-point message transmission, and the working node $o_i (i \in [1, n_q])$ of connection phase q is generated as an intermediary. The result is sent to the worker node o_j $(j \in [1, n_q + 1])$ in step $(q + 1)$. Note that if $t_{i,\, j} = 0$, there is no communication between nodes o_{i} and o_{j}.

Figure 1 shows the general data flow of the p-DOT model. For any stage q, if $q \neq p$, the output of this stage is the input of the next stage; otherwise, its result will be stored as the final result.

For a given big data load and a given environmental load that can be represented by the p-DOT model, the time cost of the load is

$$\Phi = \mathrm{O}\left(\frac{w}{n} + n\right) \times \mathrm{p}. \tag{4}$$

For a given big data load that can be represented by the p-DOT model, the computational complexity of the q-phase load is $\mathrm{O}(k_{\mathrm{q}})$.

Consider the computational behavior of phase q, which is described in the following form:

$$\begin{aligned} \overrightarrow{D}_q O_q &= \left[D_1 \cdots D_{n_q}\right] \begin{bmatrix} 0_1 & 0 & \cdots & 0 \\ 0 & 0_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & o_{n_q} \end{bmatrix}, \\ &= \begin{bmatrix} d_{1,1} & d_{1,2} & \cdots & d_{1,n_q} \\ d_{2,1} & d_{2,2} & \cdots & d_{2,n_q} \\ \vdots & \vdots & \ddots & \vdots \\ d_{k_q,1} & d_{k_q,2} & \cdots & d_{k_q,n_q} \end{bmatrix} \begin{bmatrix} \mathrm{o}_1 & 0 & \cdots & 0 \\ 0 & \mathrm{o}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & o_{n_q} \end{bmatrix}, \\ &\left(\overrightarrow{D}_q = \overrightarrow{D}_{q-1} O_{q-1} T_{q-1}, D_i \right. \\ &\left. = \begin{pmatrix} d_{1,i} & d_{2,i} & \cdots & d_{k_q,i} \end{pmatrix}^T, i \in \left[1, n_q\right]\right). \end{aligned} \tag{5}$$

Considering the communication behavior in phase q $(q \neq p)$, its formal description is as follows:

$$\begin{aligned} \left(\overrightarrow{D}_q O_q\right) T_q &= \left(\left[D_1 \cdots D_{n_q}\right] \begin{bmatrix} o_1 & 0 & \cdots & 0 \\ 0 & o_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & o_{n_q} \end{bmatrix} \right), \\ \begin{bmatrix} t_{1,1} & t_{1,2} & \cdots & t_{1,n_{q+1}} \\ t_{2,1} & t_{2,2} & \cdots & t_{2,n_{q+1}} \\ \vdots & \vdots & \ddots & \vdots \\ t_{n_q,1} & t_{n_q,2} & \cdots & t_{n_q,n_{q+1}} \end{bmatrix} &= \left[o_1 (D_1) \cdots o_1 \left(D_{n_q}\right)\right], \\ &\begin{bmatrix} t_{1,1} & t_{1,2} & \cdots & t_{1,n_{q+1}} \\ t_{2,1} & t_{2,2} & \cdots & t_{2,n_{q+1}} \\ \vdots & \vdots & \ddots & \vdots \\ t_{n_q,1} & t_{n_q,2} & \cdots & t_{n_q,n_{q+1}} \end{bmatrix}. \end{aligned} \tag{6}$$

Calculation result; each communication operator $t_{i,\, j}$ distributes the intermediate result $o_{\mathrm{i}}(D_{\mathrm{i}})$ generated by the worker node O_{i} in step q to the step by means of point-to-point message passing (including file transfer, TCP protocol and shared memory FIFO strategy, etc.) Worker nodes in $(q + 1)$.

According to formula (6), the total complexity of the computing task is

$$\Phi_{\sum \mathrm{comp}} = \sum_{q=1}^{p} \Phi_{\mathrm{comp}} = \sum_{q=1}^{p} O\left(k_q\right), \tag{7}$$

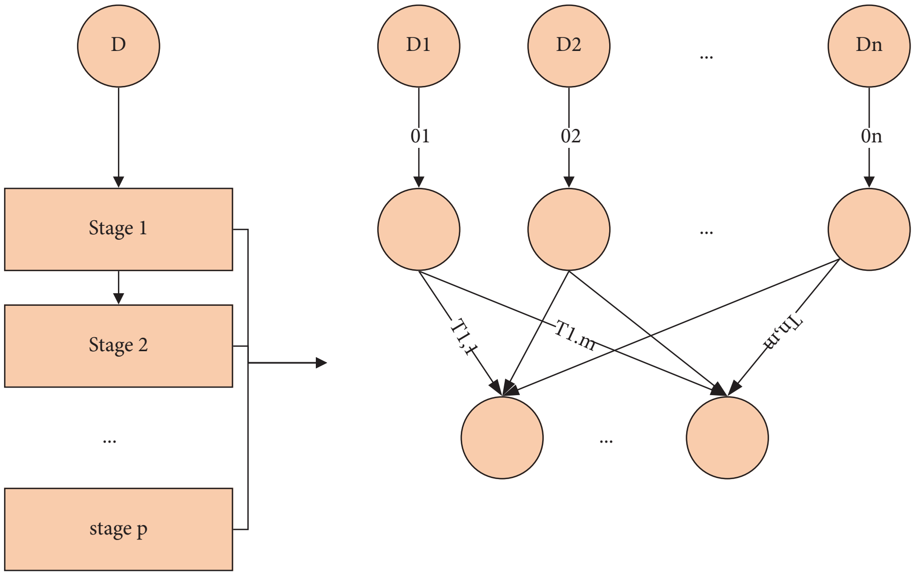

Figure 1: Data flow of the p-DOT model.

because

$$k_q = \left[\frac{w_q}{(n_q \times m)}\right] + 1. \tag{8}$$

Also, m is constant for a given ambient load. But,

$$O(k_q) = O\left(\frac{w_q}{n_q}\right) \Rightarrow \Phi_{\sum \text{comp}} = \sum_{q=1}^{p} O\left(\frac{w_q}{n_q}\right). \tag{9}$$

It can be seen from the foregoing that

$$\begin{gathered} O\left(\frac{w_q}{n_q}\right) = O\left(\frac{w_{q-1}}{n_{q-1}}\right) (\forall q \in [2, p]), \\ O(n_1) = O(n), w_1 = w, \end{gathered} \tag{10}$$

but

$$\Phi_{\sum \text{comp}} = O\left(\frac{w_1}{n_1}\right) \times p = O\left(\frac{w}{n}\right) \times p. \tag{11}$$

According to formula (5), the total computational complexity of the task is

$$\Phi_{\sum \text{comm}} = \sum_{q=1}^{p} \Phi_{\text{comm}} = \sum_{q=1}^{p} O\left(\max\left(n_q, n_{q+1}\right)\right), \tag{12}$$

because

$$n = \max(n_q), \tag{13}$$

but

$$\begin{gathered} O\left(\max\left(n_q, n_{q+1}\right)\right) = O(n), \\ \Phi_{\sum \text{comm}} = \sum_{q=1}^{p} O(n) \times p = O(n) \times p. \end{gathered} \tag{14}$$

To sum up, for a given big data load that can be represented by the p-DOT model and a given environmental load, the overall complexity of load communication is $O(n) \times p$ and the proof is completed.

3.3. *Parallel Computing Algorithm Optimization.*

The p-DOT model can be selected to represent the data load and environmental load, where $O(w/(cn) + n + c)xp$ is the time cost function of the load, and based on the period p, its form is further described as the following formula.

Consider the period q, whose form is described as follows:

$$\begin{aligned} \overrightarrow{D}_q O_q T_q &= \overrightarrow{D}_q O_q T_{\text{thread}} T_{\text{process}} \\ &= \left[(d_1' \cdots d_c') \cdots \left(d_{c\times(n_q-1)+1}{}' \cdots d_{c\times n_q}{}'\right)\right], \\ &\quad \begin{bmatrix} o_1 & 0 & \cdots & 0 \\ 0 & 0_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & o_{c\times n_q} \end{bmatrix}, \\ &\begin{bmatrix} t_{1,1}' & \cdots & t_{1,c}' & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & t_{2,1}' & \cdots & t_{2,c}' & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & t_{n_q,1}' & \cdots & t_{n_q,c}' \end{bmatrix}, \end{aligned}$$

$$\begin{bmatrix} t_{1,1}'' & t_{1,2}'' & \cdots & t_{1,n_{q+1}}'' \\ t_{2,1}'' & t_{2,2}'' & \cdots & t_{2,n_{q+1}}'' \\ \vdots & \vdots & \ddots & \vdots \\ t_{n_q,1}'' & t_{n_q,2}'' & \cdots & t_{n_q,n_{q+1}}'' \end{bmatrix},$$
$$\left(D_i = \left(d_{c\times(i-1)+1}' \cdots d_{c\times i}'\right), i \in \left[1, n_q\right]\right)$$
$$\left[\sum_{j=1}^{c} t_{1,j}'\left(o_j\left(d_j'\right)\right) \cdots \sum_{j=1}^{c} t_{n_q,j}'\left(o_{c\times\left(n_q-1\right)+j}\left(d_{c\times\left(n_q-1\right)+j}'\right)\right)\right],$$
$$\begin{bmatrix} t_{1,1}'' & t_{1,2}'' & \cdots & t_{1,n_{q+1}}'' \\ t_{2,1}'' & t_{2,2}'' & \cdots & t_{2,n_{q+1}}'' \\ \vdots & \vdots & \ddots & \vdots \\ t_{n_q,1}'' & t_{n_q,2}'' & \cdots & t_{n_q,n_{q+1}}'' \end{bmatrix}. \tag{15}$$

From a computational perspective, consider computational complexity

$$\Phi_{\sum \text{comp}} = O\left(\frac{w}{n}\right) \times p. \tag{16}$$

Since the computing layer 0 of DOT_{top} is expanded by a factor of c, then

$$\Phi_{\sum \text{comp}}'\left(p - \text{DOT}_{\text{top}}\right) = O\left(\frac{w}{(cn)}\right) \times p. \tag{17}$$

From a communication perspective, consider communication complexity

$$\Phi_{\sum \text{comm}} = O(n) \times p. \tag{18}$$

Since the communication layer T of DOT_{top} is divided into process-level communication $\text{T}_{\text{process}}$ and thread-level communication T_{thread},

$$\begin{aligned} \Phi_{\sum \text{comm}}'\left(p - \text{DOT}_{\text{top}}\right) &= \Phi_{\sum \text{process}} + \Phi_{\sum \text{thread}} \\ &= O(n) \times p + O(c) \times p = O(n + c) \times p. \end{aligned} \tag{19}$$

It can be known by derivation that

$$\begin{aligned} \Phi'\left(p - \text{DOT}_{\text{top}}\right) &= \Phi_{\sum \text{comp}}' + \Phi_{\sum \text{comm}}' \\ &= O\left(\frac{w}{(cn)} + n + c\right) \times p. \end{aligned} \tag{20}$$

To sum up, for a given big data load that can be represented by the p-DOT model and a given environmental load, when using multi-core technology, the time cost of the load is $\varphi' = O\ (w/(cn) + n + c)xp$, and the proof is complete.

If the task has partial synchronization conditions, the time cost function of its partial synchronization is $\varphi = 0$ $(w/n + s)\ xp$, where S is the number of machines where the communication behavior occurs, and p is the number of stages.

Consider period q. Assuming the first sq machine ($s_{\text{q}} < n_{\text{g}}$) where the communication behavior occurs, the formal description of the normal task execution process is as follows:

$$\begin{aligned} &\overrightarrow{\text{D}}_q\left(O_q T_q^s\right)_{\text{exec}} \\ &= \left[\, D_1 \ \cdots \ D_{n_q} \right] \begin{bmatrix} o_1 & 0 & \cdots & 0 \\ 0 & 0_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & o_{n_q} \end{bmatrix} \\ &\begin{bmatrix} t_{1,1} & \cdots & t_{1,s_{q+1}} & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ t_{s_q,1} & \cdots & t_{s_q,s_{q+1}} & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 \end{bmatrix}. \end{aligned} \tag{21}$$

From a computing point of view, considering the computational complexity, then the computing layer O of $D\ (OT^S)_{\text{exec}}$ remains unchanged, and then

$$\Phi_{\sum \text{comp}}''\left(p - D\left(OT^s\right)_{\text{exec}}\right) = O\left(\frac{w}{n}\right) \times p. \tag{22}$$

From the communication point of view, considering the communication complexity, the number of machines that have communication behaviors in the communication layer T of $D\ (OT^S)_{\text{exec}}$ is

$$s = \max\left\{s_q \mid q \in [1, p]\right\}, \tag{23}$$

and

$$\Phi_{\sum \text{comm}}''\left(p - D\left(OT^s\right)_{\text{exec}}\right) = O(s) \times p. \tag{24}$$

It can be known by derivation that

$$\begin{aligned} \Phi''\left(p - D\left(O^s\right)_{\text{exec}}\right) &= \Phi_{\sum \text{comp}}'' + \Phi_{\sum \text{comm}}'' \\ &= O\left(\frac{w}{n + s}\right) \times p. \end{aligned} \tag{25}$$

Under normal circumstances, the time complexity of judging whether the task currently meets the convergence conditions will not exceed the normal task execution process, that is,

$$\Phi''\left(p - D(OT)_{\text{judge}}\right) < \Phi''\left(p - D\left(O^s\right)_{\text{exec}}\right), \tag{26}$$

and therefore

$$\begin{aligned} &\Phi''\left(p - D\left(OT^s\right)_{\text{exec}}(OT)_{\text{judge}}\right), \\ &= \Phi''\left(p - D\left(O^s\right)_{\text{exec}}\right) = O\left(\frac{w}{n + s}\right) \times p. \end{aligned} \tag{27}$$

To sum up, for a big data iterative task that can be represented by the p-DOT model and a given environmental load, if the task has partial synchronization conditions, then the time cost function of its partial synchronization is $\varphi'' = O(w/n + s)v$, and the proof is complete.

3.4. Algorithm Detection. This paper tests the optimal number of machines n^* corresponding to the input data w of different scales and verifies the correctness of the time cost function of the p-DOT model and its inference. When testing the first 4 datasets, in order to avoid I/O acquisition conflicts between processes, only one process is running on each machine, but when testing the 5th dataset, due to the number of machines in the MPI cluster Limited, so run two processes on each machine. The number of machines in the experiment is the number of processes actually participating in the work, as shown in Table 1.

As can be seen from Figures 2 and 3, although there is a certain deviation, $\sqrt{e}\ (w)$ and $e\ (n^*)$ are obviously linearly related, that is, $e\ (n^*) = O\ (\sqrt{e}\ (w))$. Therefore, it can be seen that the optimal machine n^* is proportional to the square root of the data size w.

Combining the curves in Figures 2 and 3, we can see that for a given big data load that can be represented by the p-DOT model and a given environmental load, the time cost of the load is $\varphi = O\ (w/n + n)\ xp$ which is also correct.

The reasons for the deviation are as follows. (1) There are many factors that affect the performance of big data applications. The p-DOT model only selects the scale w of the input data and the number of machines n as the first two parameters, which makes the model parameters imperfect in terms of accuracy. (2) The experimental platform has interference from other network loads, resulting in the above communication measurement errors.

4. Enterprise Financial Management System Design

4.1. System Requirement Analysis. With the continuous development of economy and technology, people's life is more and more inseparable from computers and the Internet, which promote people's modern life with convenient, fast, and intelligent systems. But to realize good network management, it must have the support of powerful computer system. This paper designs a kind of special financial management system after actually investigating the financial management needs of small and medium-sized enterprises to meet their business operations. Based on the principle of practicality, the system uses information technology to manage financial information and data and inputs data and financial information into standard computers in actual work, giving full play to the computer's rapid processing capabilities and standardized management. Through the analysis and investigation of the actual situation, the functions of master data management, voucher management, and data security problems such as user authentication and user authority in system management have been solved.

Table 1: Input data scale w and corresponding optimal number of machines n^*.

w/n^*	0.5 GB	7.5 GB	50 GB	500 GB	1 TB
Wordcount (MPI)	128	320	1280	4000	8000
Terasort (MPI)	128	640	1600	5000	10000
Wordcount (Hadoop)	64	256	800	4096	—
Terasort (Hadoop)	64	128	640	2048	—

The main system requirements are as follows.

Simple Operation and Friendly Interface. It adopts Windows operating habits, which is beautiful and elegant. After simple training, administrators can easily operate the interface.

Permission Control, Safe and Reliable. Different permissions are assigned to different categories of administrators. Users can change the permissions of each operator. After the operator logs in to the system and enters a password, the system will automatically grant permissions to prevent unauthorized operations, which is safe and reliable and can prevent the unclear division of responsibilities.

Data Query, Fast and Convenient. According to the basic information system, it provides a powerful daily processing query function, which can realize simple query and fuzzy query, and users can also print reports.

The report is reasonable and easy to use. According to the system requirements, the system can meet the statistical requirements of financial managers.

System performance requirements refer to the requirements for system reliability and functional scalability in addition to system functional requirements, which have a greater impact on the use environment and business specifications of the system.

4.1.1. Quick Response Capability. Although the management system designed in this paper involves a small range of management, it also includes a large number of commodity types. When multiple users access the system, higher requirements are placed on the database system and server that can quickly respond to query requests sent to users' needs. In addition, the frequent exchange of system business data requires the system to control the response time within an acceptable range.

4.1.2. Load Capacity. The system load depends on the number of users used and the frequency of services. After running for a period of time, the number of users will stabilize at a certain number, and the system load requirements can be set at this time. In addition, although the system-related data storage is relatively large, the storage capacity of the database is not a problem.

4.1.3. Security. System security is achieved through user authentication. The user enters the correct user name and password to log in to the system. If the input is incorrect, you

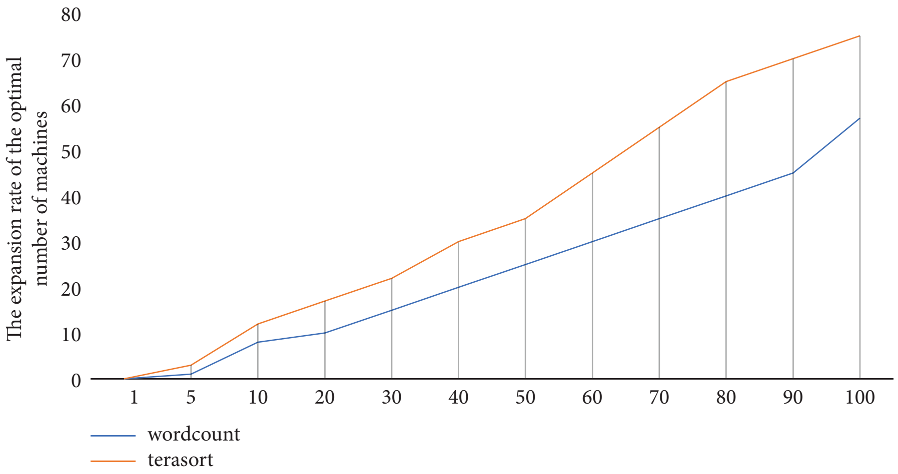

Figure 2: The square root of the expansion rate of the input data size in the MPI test.

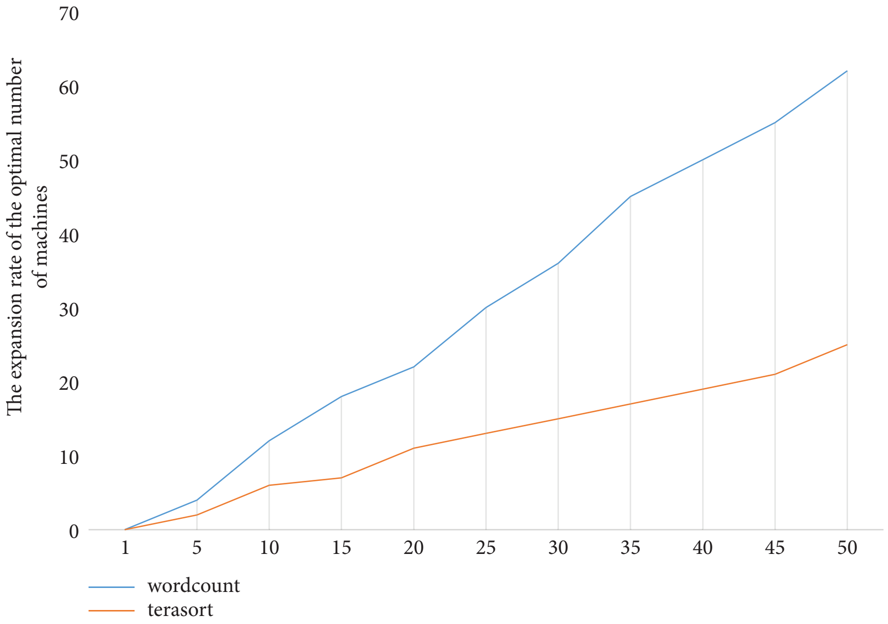

Figure 3: The square root of the expansion rate of the input data size in the Hadoop test.

will be returned to the login window. Also, the system needs to be equipped with data recovery and backup functions to avoid the risk of data loss due to crashes.

4.1.4. Reliability. System reliability refers to the ability of management system software to maintain its own performance after practical application, involving hardware environment, network environment, system platform characteristics, system development platform, and so on. System reliability is a global, system-specific requirement that includes fault tolerance, fault resilience, and system maturity. Fault tolerance refers to the ability of system software to withstand and prevent such errors when the management system fails. Failure is unavoidable, but if the system can avoid the collapse of the whole system, it also meets the design conditions. System stability refers to whether the normal state of the system can be restored by re-entering the system if there is a serious failure that affects the normal operation of the system during the operation of the system, such as the protection mechanism during data input.

4.1.5. Ease of Use. Ease of use is a system feature demanded by every customer. After completing the development of the management system, in addition to ensuring complete functions and normal use, it is also essential for the company's existing operators to quickly use the system. Operators can fully understand the functional use and maintenance operations of the system in a short time or after simple training, indicating that the system is easy to use. Enterprise software users need little logical thinking to understand the full capabilities of the software. Users can master the use of the software according to the instructions contained in the software, which greatly shortens the software training process. The navigation of the software interface is concise and clear, and the user can achieve the desired operation function through fewer pages during the operation.

4.1.6. Maintainability. According to the principle of software development, a software should be handed over to the customer for testing and actual use after the design is completed. No matter how much manpower and material resources are spent, there must be various defects in software design, which can be reflected in the failure of operation. The complexity of the maintenance process and the level of maintenance costs in the event of a failure indicate the sustainability of the system. After a software failure occurs, the system needs to provide a log-like function, so that maintenance personnel can find the cause of the failure in a short period of time based on the log and experience. Once software administrators have discovered the cause of the failure, they can fix the failure with minimal cost and time.

4.2. System Architecture Design. Based on actual needs, after opening the browser software of the operating system and entering a fixed domain name, the user can enter the system login page under the condition of networking. After the user completes the login and authentication, he can access various functions provided by the system.

MVC is usually divided into view presentation layer, model layer, control layer, and other components. Indicates the interaction with the user interface and is responsible for the realization of the system UI functions. The model layer mainly processes the data entering the business. The control layer receives and processes all the user's requests and calls the processing interface of the system model layer to respond to the user's request.

The main functional structure of the business financial management system developed in this paper is shown in Figure 4.

As shown in Figure 4, in the system architecture design, Spring is used to coordinate various processing operations and business logic layers. The Struts presentation layer describes the technical details of the implementation of the functional modules related to the system and then separates the modules through Spring. Hibernate architecture uses session factory for integrated database operations. Hibernate's transaction processing mechanism is used to handle complex data interactions and data manipulations.

Based on the analysis of system requirements, this paper develops and designs a financial management system that meets the needs of enterprises. The system function structure is shown in Figure 5.

4.3. System Interface Design. According to the principle of object-oriented design and the relevant guiding ideology of SOA framework, it is planned to execute and complete various operations of the financial management system of small and medium-sized enterprises through the services in web services. The service is defined as follows:

```
public interface SourceManage {
@Profiled (tag = “SourceManage”)
public int uploadSource (String sourcePath);
@Profiled (tag = “SourceManage”)
public int updateSource (int sourceId, SourceInfo sourceInfo);
@Profiled (tag = “SourceManage”)
public SourceInfo downloadSource (int sourceId);
@Profiled (tag = “SourceManage”)
public SourceInfo getSourceById (int sourceId);
@Profiled (tag = “SourceManage”)
public List<SourceInfo> getSourceBySourceName (String sourceName);
@Profiled (tag = “SourceManage”)
public int deleteSource (int sourceId);
@Profiled (tag = “SourceManage”)
public int isSourceExit (int sourceId);
}
```

As shown in the code snippet above, the SourceManage interface of financial management mainly includes operations such as uploading (uploadSource), downloading (downloadSource), querying (getBySourceId, getBySourceName), deleting (deleteSource), querying whether the financial resource file exists (isSourceExit) information data, and so on to fully realize the full authority management of access to related resources.

4.4. Analysis of System Test Results. The test environment of the client is shown in Table 2. The test process is similar to the development process, and it is also carried out in stages and steps. It is impossible to test the entire system as a separate entity from the beginning. To test this system, start with each functional module.

First, start with each functional module of the system and test it as a unit. Jump from one form to another, examine different situations, and use single-step tracing, set breakpoints, and output intermediate variables in between. Finally, the different processes are combined and tested in a relatively complete compilation part. Because it is not clear whether the results obtained are correct, we first output the results to the file, then evaluate the accuracy of the results according to different situations, and gradually follow up to determine the aspects that need to be changed or improved.

Second, while making sure the main part is working properly, try to change other non-main parts of the module and make improvements with reference to the related functions of WordPad and Notepad in Windows.

Finally, the system as a whole has been fully tested in many aspects, and many errors and imperfections have been modified to ensure that the system functions meet the design requirements and can work normally. Test each module of the system. After testing each module, assemble all the

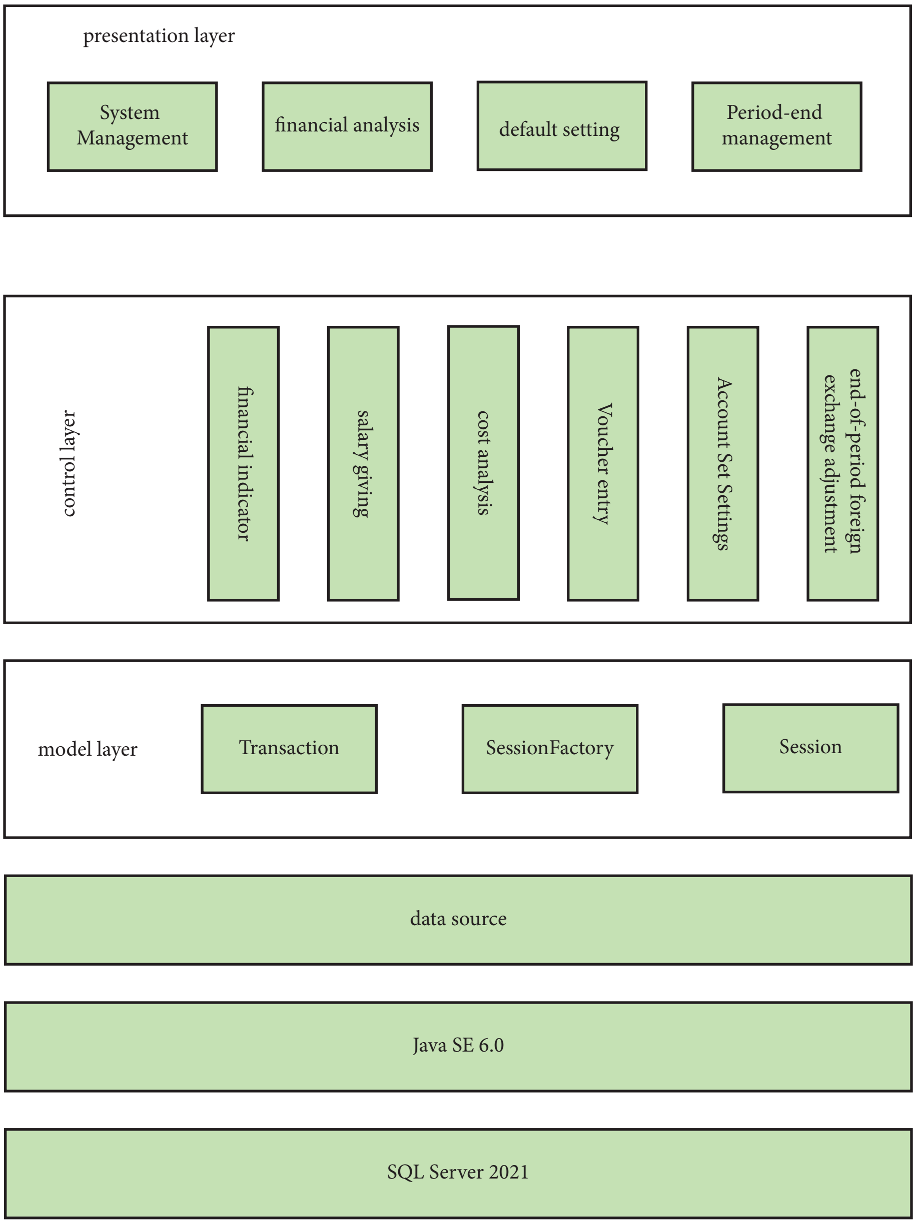

Figure 4: Architecture diagram of enterprise financial management system.

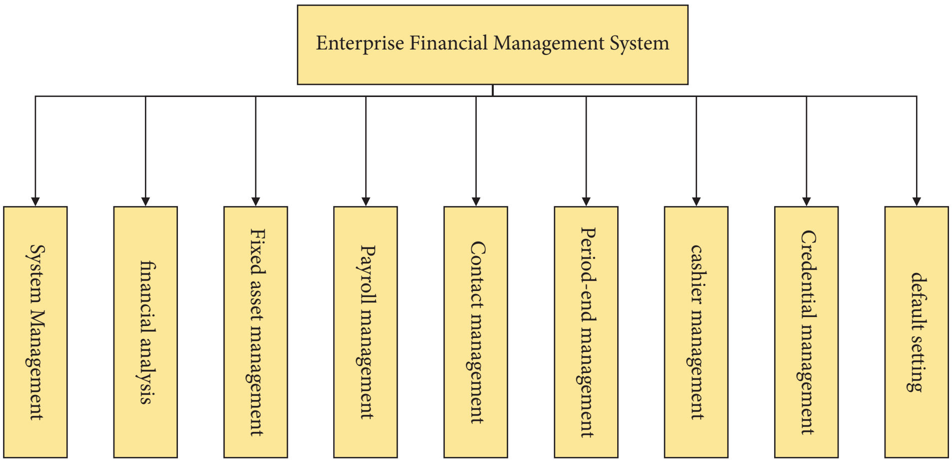

Figure 5: Functional structure diagram of enterprise financial management system.

Table 2: Client test environment.

-	OS type		Browser tools	Hardware information
Client environment 1		Windows XP	IE10 Chrome Firefox Opera V24	— — — —
Client environment 2	Windows	Win 7	IE10 Chrome Firefox Opera V24	CPU intel (R) i5-12400f@ 2.4 GHz Memory: 12 GB Screen resolution: 1920*1080
Client environment 3		Win 10	IE10 Chrome Firefox Opera V24	— — — —

Table 3: System test.

No.	Test content	Testing method	Test results
1	Security system	Software deployment based on design criteria	System can be successfully installed
2	Uninstall the system	Software deployment based on design criteria	Can successfully uninstall the system
3	Basic data	System, administrator, and user	The system automatically updates the data within 24 hours and accurately enters the data
4	Manage business	Different administrators can perform corresponding business management operations	Accurately manage system data
5	Inquiry business	Corresponding information can be released, and data query can be performed after entering the operation module	Complete target data query based on conditions
6	Business functions	Statistics, management, analysis of input data	Display target data and analysis results
7	Maintenance system	Perform system data operations and user rights management	Set for user role permissions
8	Software fault tolerance	Judgment at test time	When the system fails, report an error and recover

Table 4: Financial management system performance test results.

Test indicators	Test results
Delay test	The delay time is 1.34 seconds The delay time is 2.11 seconds The delay time is 3.46 seconds
Reliability test	Success rate: 98.32%
Concurrency testing	The number of concurrent users is greater than or equal to 250

modules, and then test the interface as a whole. The operation stability of this system is tested through the following aspects, as shown in Table 3.

The main goals of the test are system reliability, system page latency, and system performance. Set system uptime to 48 hours. During the test, we observe the operation of various system indicators within 48 hours and calculate the average value of the test through multiple tests, as shown in Table 4. The test results show that the software design can meet the expected performance requirements.

Through the test of the system, the results show that the system is simple, easy to operate, and practical, and each interface meets the requirements of system safety.

5. Application Directions of Artificial Intelligence in Enterprise Financial Management

5.1. Reducing Process Time and Facilitating Real-Time Management. In terms of financial analysis, it is gradually shifting from traditional financial analysis to analysis supported by real-time system data. With the popularization and implementation of enterprise financial management systems, financial activities are becoming more and more concise, which can give enterprises more time and space to optimize and operate financial function management. In the later stage of financial system optimization, procurement projects can consider further developing the functions of the financial statement module to facilitate the acquisition of data and ultimately provide financial evaluation, analysis, and decision making for enterprises.

5.2. Information Disclosure and Sharing to Improve Management Efficiency. With the development of society and technology, in order to optimize the financial management process, promote the centralization of decentralized financial management, and improve the efficiency of financial activities, enterprises must establish a center that can be used for internal information exchange and communication. Financial information sharing can effectively improve the efficiency of enterprise system information interaction, thereby promoting information sharing among various departments, and sharing with other departments, and clearly establish a special financial information platform within the enterprise. The system includes customer information, business information, decision-making information, and so on. This will shape the future of business management.

5.3. Strengthening Risk Management and Improving Decision-Making Ability. Financial risk management is one of the core contents of an enterprise's financial management system, and its main function is that the decision-making system acts directly through risk monitoring and feedback. However, because the company lacks a risk management system, the lack of financial risk management directly leads to a negative impact on business operations. Therefore, it is necessary to improve the risk management and control system.

In terms of effectively building a financial risk monitoring system, enterprises can start from the following three aspects. First, enterprises can form relevant standardized data indicators according to their own conditions through comparative analysis of other enterprises. Second, use real-time relevant data analysis to obtain suitable data and compare it with real-time indicators and standardized risk indicators. Furthermore, link processing is performed for the appropriate business. For example, taking the changes in the company's accounts receivable recovery rate and budget cost allocation as an example, we combine these indicators with the company's initial alert threshold for real-time monitoring and adjustment.

6. Conclusion

By using the financial management system, enterprises can improve the efficiency of financial management and achieve twice the result with half the effort, especially in the fierce market competition, which can give enterprises a development advantage. In the case of relatively low production level and production efficiency of small and medium-sized enterprises, their market share is very low. At this time, the financial management system is not only a simple application software system but also an important part of production and operation. Practice shows that the system has the following advantages: friendly interface and simple operation. Operators with limited computer access can also operate the menu item prompts; detailed information management, including adding, deleting, and other specific operations, provides powerful navigation, query, and statistical functions; the system supports multi-identity user operations. Users are effectively connected to facilitate the comprehensive operation and management of basic financial information and enterprise information; the business process is arranged reasonably, and the division of labor in the voucher verification stage and the posting stage is clear, which is in line with expectations; the report statistics are detailed, and users can print statistical reports based on their needs. The data are accurate and clear, which is convenient for data analysis. Due to the limited level, the system is not perfect in some aspects, and more research is needed. (1) The security and reliability of the financial management system need to be improved and optimized. (2) The functional module design is not detailed enough, and the data analysis integration function still needs to be perfected.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The author declares that there are no conflicts of interest.

References

[1] L. Wang and Y. Wang, "Supply chain financial service management system based on block chain IoT data sharing and edge computing," *Alexandria Engineering Journal*, vol. 61, no. 1, pp. 147–158, 2022.

[2] C. Pollitt, "Integrating financial management and performance management," *OECD Journal on Budgeting*, vol. 1, no. 2, pp. 7–37, 2001.

[3] H. Karadag, "Financial management challenges in small and medium-sized enterprises: a strategic management approach," *Emerging Markets Journal*, vol. 5, no. 1, pp. 26–40, 2015.

[4] K. Lu, Y. Fu, C. Gu, and L. Zhang, "Problems and solutions of popularization of accounting computerization," *Physics Procedia*, vol. 33, pp. 1155–1159, 2012.

[5] L. Zhou, "The research on issue and countermeasures of accounting information of SMEs," *International Journal of Business and Management*, vol. 5, no. 3, p. 223, 2010.

[6] S. Murungi and C. Kayigamba, "The impact of computerized accounting system on financial reporting in the ministry of local government of Rwanda," *Journal of Emerging Trends in Economics and Management Sciences*, vol. 6, no. 4, pp. 261–265, 2015.

[7] M. M. Gadzhiev and Y. G. Buchaev, "Efficiency assessment of enterprise innovation activities," *Life Science Journal*, vol. 11, no. 10, pp. 581–586, 2014.

[8] A. Chornovol, J. Tabenska, T. Tomniuk, and L. Prostebi, "Public finance management system in modern conditions," *Investment Management and Financial Innovations*, vol. 17, no. 4, pp. 402–410, 2020.

[9] H. He, Y. Hong, W. Liu, and S. A. Kim, "Data mining model for multimedia financial time series using information

entropy," *Journal of Intelligent and Fuzzy Systems*, vol. 39, no. 4, pp. 5339–5345, 2020.

[10] O. Bulkot, "Modern system of international financial management in multinational companies," Економічний часоШис-ЧЧІ, vol. 1-2, no. 2, pp. 8–11, 2015.

[11] S. Selvanayaki, S. D. Sivakumar, A. Rohini, and K. Mani, "Financial management practices and profitability of modern rice milling firms in Kangayam Cluster, Tamil Nadu," *Agricultural Economics Research Review*, vol. 29, no. 2, pp. 297–306, Article ID 17251, 2016.

[12] R. P. Dameri, "Improving the benefits of IT compliance using enterprise management information systems," *Electronic Journal of Information Systems Evaluation*, vol. 12, no. 1, pp. 27–38, 2009.

[13] Z. Zhang, Y. Liu, J. Zhang, and X. Song, "Research on the influence of new generation of information technology on contemporary enterprise logistics management information system," *Journal of Physics: Conference Series*, vol. 1648, no. 4, Article ID 042039, 2020.

[14] X. Huang and F. Guo, "A kernel fuzzy twin SVM model for early warning systems of extreme financial risks," *International Journal of Finance & Economics*, vol. 26, no. 1, pp. 1459–1468, 2021.

[15] D. Mhlanga, "Industry 4.0 in finance: the impact of artificial intelligence (ai) on digital financial inclusion," *International Journal of Financial Studies*, vol. 8, no. 3, p. 45, 2020.

[16] J. De Weerdt, A. Schupp, A. Vanderloock, and B. Baesens, "Process Mining for the multi-faceted analysis of business processes—a case study in a financial services organization," *Computers in Industry*, vol. 64, no. 1, pp. 57–67, 2013.