

Received May 11, 2020, accepted May 19, 2020, date of publication May 25, 2020, date of current version June 19, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2997079

Overview of Intelligent Online Banking System Based on HERCULES Architecture

GUANGSHENG LUO¹, (Graduate Student Member, IEEE),

WENWEI LI², (Graduate Student Member, IEEE),

AND YUZHONG PENG³, (Student Member, IEEE)

¹Intelligent Information Processing Laboratory, School of Computer, Fudan University, Shanghai 200433, China

²School of Computer, Fudan University, Shanghai 200433, China

³Intelligent Information Processing Laboratory, School of Computer, Guangxi Teachers Education University, Nanning 530001, China

Corresponding author: Guangsheng Luo (luoguangsheng03@126.com)

This work was supported in part by the Intelligent Information Processing Laboratory, School of Computer, Fudan University, Shanghai, China, and in part by the company in Shanghai.

ABSTRACT The online banking transaction system is the application system with the most complex business, the most demanded, and frequent version updates in the software engineering application system. The existing online banking business sub-module is intelligent and faces major challenges in security. Traditional online banking systems cannot meet this capability. This article combines machine learning and online banking business module design to implement a business agent online banking system based on a new architecture. The article first proposes new features and new challenges of the online banking system, discusses the technical problems solved by the intelligent online banking system, and analyzes HERCULES Architecture business intelligent machine learning algorithm model, smart deposits and white-collar loans and other core processes, and then designed and implemented an intelligent online banking system business model, focusing on the issues of the intelligent online banking system, soft load balancing implementation and transaction security, through business implementation. The effectiveness of the proposed business agent is verified. The actual use results show that the intelligent online banking of the HERCULES architecture has greatly improved the intelligence and security of the traditional online banking system. Finally, we summarize and analyze the value and innovation of the intelligent online banking system, and look forward to the shortcomings of the system.

INDEX TERMS HERCULES architecture, machine learning, hot release, smart deposits, white-collar loans, load balancing.

I. INTRODUCTION

Financial activities affect the economic development of countries around the world, and financial markets play an important role in the economic and social organization of modern society [1]. In financial markets, data are priceless assets. The success of investors depends on The quality of information used by investors when making trading decisions, and how quickly investors can make investment decisions. However, with the modernization of financial transactions and information systems, a large amount of information has emerged, facing massive financial Data, many traders do not know where to start analyzing financial assets. In the past few decades, both the financial industry

The associate editor coordinating the review of this manuscript and approving it for publication was Guanjun Liu¹.

and the academic world have conducted extensive research on the movement of financial markets [2]. The intelligent online banking system is a new type of bank operation model built using artificial intelligence technology under such a background. Customers can enjoy a more intimate and personalized service experience. Online banking has the obvious characteristics of fewer institutions, fine staff, and low cost. Provide customers with more favorable interest rates than traditional banks, launch financial products and services with lower fees and more convenient channels, reduce operating costs for commercial banks, Winning a competitive advantage, and expanding the space for differences that bring unlimited possibilities. The emergence of an intelligent investment service model has irreplaceable advantages for traditional banks, but in the network era, millions of data processing per second, tens of millions of daily transactions,

real-time acquisition. As a result, there are many problems to be solved, and how to support the performance, stability, and scalability of the online banking system is one of them. Before the emergence of applications such as online banking and intelligent customer service, machine learning has been widely used in the financial field, especially with the improvement of computing power and the popularization of machine learning technology, machine learning has become an indispensable component of the financial ecosystem. Part. Such as loan approval, automatic asset management, risk assessment, etc.

1) The application status of machine learning algorithms in the field of financial banking.

The application of machine learning in the financial field can be roughly divided into several aspects. One is the customer-oriented application, that is, the online banking system in this article, which has intelligent algorithm transactions and intelligent customer service robots. Electronic channel system; second is back-end applications, including risk management models, capital optimization portfolio, etc.; the third is financial asset portfolio management and compliance management of financial institutions, etc.

Algorithmic trading is also called automated trading. In the 1970s, many hedge funds and financial institutions have started to use artificial intelligence technology in trading to implement rapid trading decisions on the system. Algorithmic trading systems [3] usually operate thousands or millions of transactions per day. Portfolio management, the most popular smart investment advisor is to search and match in various assets and financial instruments according to the goals entered by the user. Technically through charts, moving averages, relative strength index, momentum, conversion rate, and price. Other mathematical transformations to analyze the direction of movement of financial instruments [4]. Research the size, price-earnings ratio, and assets and liabilities of the company before investing [5]. Loan insurance, machine learning algorithms in bank online banking systems can be based on millions of consumer case data (Age, work, market conditions, etc.) and the results of borrowing or insurance for development and training (whether this person defaults, whether the loan is returned on time, has a car accident occurred?), Etc.

2) Research Motivation of Intelligent Online Banking System.

The online banking system currently uses a lot of funds in artificial intelligence technology, such as funds and financial management. For example, the machine will automatically place an order transaction based on the algorithm set above. A kind of technical analysis method such as fundamental analysis. It divides into two categories from the realization of technical architecture. One is based on the traditional J2EE architecture. J2EE uses JAVA language to develop various large-scale enterprise-level applications; the other is based on Struts. Architecture to build an online banking system, Struts is a free and open-source WEB layer application framework, developed by the Apache Software Fund, Struts is highly

configurable, contains front-end control components and a series of action classes, action mapping, and practical tools for processing XML Class, so there is an urgent need for a new set of architectures that can support intelligent implementation.

From the perspective of intelligence, it is currently mainly reflected on intelligent investment consultants [6], intelligent customer service [7], intelligent venture capital, and speech recognition [8]. In terms of banking channels, the bank funds system and client service system are still very different from the online banking system. The online banking system pays more attention to transactions, rules, risks, etc., and focuses on specific sub-businesses. Credit consumption, transfers, deposits, and loans are the motivation of this paper. This paper attempts to combine machine learning neural networks [9], decision tree algorithm and specific online banking sub-business to solve business intelligence.

Section 1 of this article presents the new features and challenges of the intelligent online banking system. Section 2 analyzes the technical problems and security frameworks solved by the intelligent system of the HERCULES architecture. Section 3 be focused on the analysis of machine learning algorithms for smart deposits and white-collar loans. And business transaction security. Section 4 is designed and implemented based on Section 3 and is compared with similar systems. Section 5 summarizes the full text of the innovation and shortcomings of the research on intelligent online banking systems. The research direction of the online banking system that deserves attention is preliminaries discussed.

A. NEW FEATURES OF BUSINESS INTELLIGENCE AND SECURITY

From the perspective of the development of financial technology, the development technology of artificial intelligence is gradually maturing. Artificial intelligence technology can promote the financial service level of the banking industry and the efficiency of operation and management to be greatly improved. On the one hand, artificial intelligence uses user data analysis and management to achieve intelligence while Sum up the corresponding countermeasures for the bank's operation and management in time, to promote the intelligentization of banking development and management [10].

Artificial intelligence is considered to be the next generation of technological commanding heights. Two U.S. companies Equ Bot LLC and ETF Managers Group jointly launched the world's first ETF-AIEQ that uses artificial intelligence to invest in October 2017; the fund is very important in the online banking system. A product represented by the online banking system after nearly ten years of developments, its products have become more and more diversified. In addition to funds, there are also credit scores, etc., that is, the use of new types of unstructured and semi-structured data sources (such as Social media, mobile phones, and text messages) to capture the borrower credit and use artificial intelligence to assess qualitative factors such as consumer behavior and willingness to pay. The characteristics of the intelligent

online banking system in this article are relying on a portal-based, personal online banking, corporate online banking, and other “channel-centric” HERCULES architecture. At present, the banking system business faces the main characteristics and challenges of the development of artificial intelligence.

Business security is a core issue in the development of the banking system. It can be said that it is related to the survival of the banking system. Poor information security due to technical reasons has become a major factor restricting the development of China’s online banking. Besides, network security and confidentiality measures have invaded the financial system network. There are occasions when the Bank of Shanghai’s direct online banking registration, login, and user name registration need to install the corresponding security controls. Nanjing Bank’s direct online banking is controlled by a password plus a mobile phone verification code. China Resources Bank’s direct online banking is through password security controls that are used according to the user’s choice. The intelligent online banking systems of these three banks all use traditional security mechanisms, which are relatively weak and have a low degree of security protection, which cannot meet the depth, breadth, and accuracy requirements of risk management and control of the core system of the bank. Especially credit cards The rapid development of business and the need to improve risk control capabilities have been thoroughly researched and applied. Personal credit risk measurement is achieved through credit scoring. Credit scores are based on customer basic information and credit history data, and use corresponding credit scoring models to obtain different levels of credit score.

In this paper, the intelligent online banking based on the HERCULES architecture will integrate corresponding algorithms such as classification algorithms, factor analysis, individual user level credit analysis, and decision trees to process related business. For example, integrated credit scores are based on the BP algorithm, The motivation of the back propagation algorithm is to adjust the connection weight between each neuron adaptively in the training process of the neural network to find the best mapping function between input and output so that the objective function or loss function is minimized. Complete tasks such as classification and regression [11], model as follows 1:

$$y = g \left(\sum_{j=1}^4 v_j g \left(\sum_{i=1}^7 w_{ji} x_i + b_j \right) + b \right) \quad (1)$$

The expression using vectors is:

$$y = g \left(Vg \left(W^T X + B_1 \right) + B_2 \right) \quad (2)$$

Among them: $X = (x_1, x_2, \dots, x_{32})^T$ is the input vector representing the user’s information, Y is the output vector representing the rating level, the derivation of the BP algorithm can be Refer to chapter 3.1.

B. NEW CHALLENGES FOR BUSINESS INTELLIGENCE

The bank’s business development is carried out based on traditional e-banking, product research and development and online are subject to traditional banks, and it is not targeted. It cannot achieve cross and combine transactions for multi-dimensional regions, customer groups, customer levels, products, channels, services, etc.

- 1) Challenge 1: The challenge of intelligitizing the traditional online banking architecture.

The online banking system is essentially a transaction system. The traditional online banking is just an information management system, which lacks intelligent related content. The online banking system in this paper uses a deep learning algorithm on the HERCULES architecture. The direct proposed by Moody *et al.* Is introduced. Reinforcement learning (Direct Reinforcement Learning) trading system [12], [13].

If the price sequence of a wealth management product in the online banking system is $p_1, p_2, \dots, p_t, \dots$, and the price return at time t is a trading signal, a differentiable nonlinear function can be used to approximate the trading decision at each time point, namely:

$$F_t = \tanh (\langle w, x_t \rangle + b + uF_{t-1}) \quad (3)$$

Among them x_t is the feature vector being the current market environment. In machine learning transactions, $x_t = [r_1, \dots, T_{t-M+1}] \in \mathbb{R}^M$ the nearest M return values are directly, used to $\langle \cdot, \cdot \rangle$ represent the inner product operation. The parameters of the trading system are w, b, u . After the decision, the transaction returns at time t are:

$$R_t = \mu (F_{t-1} r_t - \delta |F_t - F_{t-1}|) \quad (4)$$

where u is a constant and the term $F_{t-1} T_t$ is the profit or loss caused by market fluctuations. The term $\delta |F_t - F_{t-1}|$ represents the transaction cost caused by the change in the fund position at time t . The characteristics of intelligent online banking transactions based on the HERCULES architecture are as follows:

$$\begin{aligned} & \min_{\{\theta, d(\cdot)\}} -U(R_1, \dots, R_t, \dots, R_T) + \alpha \|x_t - A(x_t)\|^2 \\ & \text{where } R_t = \mu (F_{t-1} r_t - \delta |F_t - F_{t-1}|) \\ & F_t = \tanh (\langle w, f_t \rangle + b + uF_{t-1}) \\ & f_t = d(x_t) \end{aligned} \quad (5)$$

where $U(R_1, \dots, R_t, \dots, R_T)$ represents the value function of the trading system, and $A(x_t)$ being the final reconstructed vector of the original input vector

- 2) Challenge 2: The challenge of business security.

The business online banking system is required to meet the user’s multi-dimensional, targeted, and personalized needs. It also requires the user to be close to the user’s behavior and habits. It is a fool operation for users to complete the deposit and loan business.

User security prevention, client software recognition of phishing websites, transaction risk evaluation, abnormal transaction behavior of customers, for example, the customer has made a deposit business in the current city, but less than an hour. But withdrew money in another city, which is not in line with common sense. Secondly, users automatically purchased wealth management fund products, In addition to how the password is guaranteed to be personally safe for personal identity authentication, and adaptive transaction risk processing strategies are all challenges.

- 3) Challenge 3: Challenges of intelligent service soft load. The transaction concurrency of the online banking system is tens of millions. At present, the common practice is to load through high-cost hardware, and the price of a hardware device can be hundreds of thousands to millions. It can be carried out according to demand during transactions Load balancing processing, no need to use hardware equipment for loading. Complex transactions can be completed through a flexible interface configuration, load balancing weight distribution, and communication link health status check, and routing parameter configuration are all completed in the configuration file. When various intelligent services are concurrent, they no longer use the hardware.

Summary of This Section: From the analysis of the above characteristics and challenges, this article introduces a new technical architecture and machine learning combined solution. Through the design of the HERCULES architecture in this article, it can provide flexible management, control, and operation of business rules, which can be effectively solved. These inherent problems in the traditional online banking system, while incorporating new machine learning algorithms into specific business modules, effectively solve the problem of intelligence and automation. On the other hand, when intelligent and automated transactions are realized concurrently, Independently design a set of soft load algorithms to better support the entire intelligent system.

II. TECHNICAL ANALYSIS OF HERCULES ARCHITECTURE INTELLIGENT ONLINE BANKING SYSTEM

The reduced Context design of the HERCULES architecture avoids the definition of too many POJOs, that is, simple java object definitions. There is a standard business process template, data verification style, and data access SqlMap and security controls. HERCULES implements channels through the channel access layer The separation of the processing layer and the channel-independent layer. This makes the transaction logic development independent of the channel, and truly achieves the design goal of “one-time development, multi-channel reuse”. The following introduces the HERCULES architecture component model and load balancing, and The architecture engine template is analyzed and elaborated around the intelligent online banking system.

A. MULTI-CHANNEL ACCESS DEVELOPMENT COMPONENT OBJECT MODEL

The HERCULES platform architecture includes the following six major types of component modules: HERCULES Runtime, HERCULES Dynamic Module Repository, HERCULES Web Console, HERCULES IDE, and HERCULES IDE HERCULES SmartWeb and Multi-Channels Support, HERCULES ToolBox. HERCULES Architecture Logical Structure See figure 1.

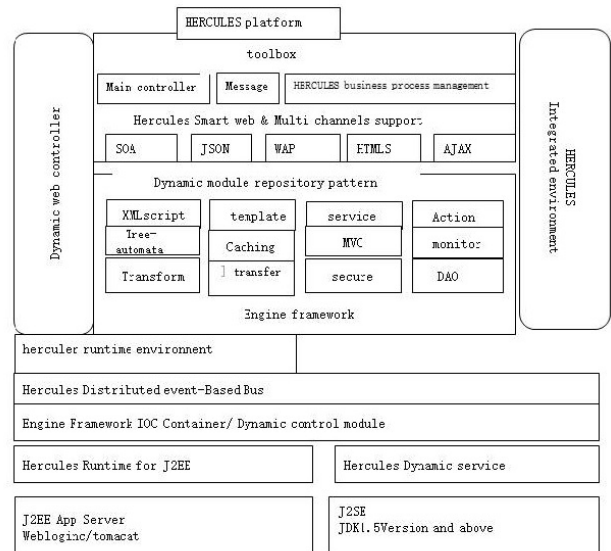


FIGURE 1. HERCULES architecture.

The component object model is an object-oriented programming model used for inter-process communication and as a platform for component software development. The HERCULES platform supports multi-channel access, and its channel layer supports HTTP, TCP / IP, SOAP, MQ, HTTPS, UDP / IP-Multicast, SNA LU6.2, and other communication methods support automatic analysis and formatting of different types of messages such as custom messages, ISO8583 messages, XML, serialized streams, etc. The only Java platform that truly supports multi-channel application development. When running in a J2SE environment, the HERCULES platform directly provides key features that the J2SE platform does not have, such as thread management, session management, transaction management, and cluster technology. Enterprise-level computing (such as TCP / IP, MQ, IP-MultiCast, etc.) is crucial.

HERCULES ToolBox provides a large number of practical tools and systems to fully support SOA-based, dynamic module-based, component-based system development. The main tools are HERCULES BPM: business workflow management products, follow WFCM specifications, including business process definition tools, Workflow engine, workflow API, workflow management monitoring, etc. HERCULES HostSimulator: host simulation system. It can greatly accelerate the progress of system development,

integration, and testing. HERCULES Report: provides report customization tools and APIs to achieve WYSIWYG reports definition.

The design of HERCULES architecture must ensure application security through platform-level XML configuration and rule engine-based configuration guarantees, thereby reducing the requirements for application developers. The system-level application security control is achieved through the control of the dynamic chain of responsibility. HERCULES provides Mature application security components, such as user and certificate binding components, digital verification components, etc. HERCULES architecture can support both J2SE and J2EE environments. For different Java environments, HERCULES architecture provides corresponding operating environment support. For J2EE environments The HERCULES platform provides the HERCULES Runtime for J2EE, which supports the current mainstream J2EE application servers, such as IBM WebSphere, Oracle WebLogic, JBoss, TomCat, Abuse, HERCULES Runtime for J2EE. The main features of the HERCULES architecture are shown in Table 1.

TABLE 1. Main features of the HERCULES architecture.

name	versions	Main Features
PowerEngineDynamic Web Edition Runtime	V6.1	Lightweight component container (IoC)
		Lightweight dynamic module container
		Lightweight dynamic Web module container
		Lightweight dynamic service containers
		Distributed event bus
		Support dynamic module hot deployment
		Support Tree Automata
		Provides a comprehensive application security policy
		Supports rich message processing formats
		PowerEngineDynamic Edition Runtime
Lightweight dynamic module container		
Lightweight dynamic service containers		
Distributed event bus		
Support dynamic module hot deployment		
Support Tree Automata		
Provides a comprehensive application security policy		
Supports rich message processing formats		

The HERCULES dynamic module library contains the following characteristics and more comprehensive functions: it provides richer cross-domain components; it provides richer domain-oriented components; module isolation: the HERCULES platform manages reusable components by dynamic modules, and uses dynamic modules. Configuration, you can set the API firewall between the modules to effectively protect or expose the necessary interface (API).

The module firewall technology ensures the system's robustness, stability, and security; high-efficiency, loosely coupled lightweight components Assembly mechanism: Many existing basic platforms use the so-called XML data bus to assemble components. This coupling mode not only does not mention loose coupling but also has high coupling performance costs. HERCULES uses IoC dependency injection technology and interface-based dynamic services Technology for assembly of components not only guarantees true loose coupling but also realizes a lightweight coupling mechanism.

B. AOP DYNAMIC MODULE PROCESS TEMPLATE AND BUSINESS TRANSACTION SECURITY FRAMEWORK

1) AOP DYNAMIC MODULE PROCESS TEMPLATE

HERCULES adopts AOP technology, which is aspect-oriented programming. It can implement a technology to dynamically add functions to the program without modifying the source code through pre-compilation and dynamic proxy at runtime. At the same time, it uses Custom Tag technology to enable platform development In a Web-oriented system, no Java code has to be written in all dynamic pages. This greatly improves development efficiency and simplifies maintenance. IoC technology (dependent injection technology) is used to control the relationship between IoC feature container control components rather than traditional direct control in code. As early as 2003, Crane Company launched a multi-channel application development platform based on IoC technology. To date, HERCULES is still the only platform in the Java basic platform that uses IoC technology and supports deployment in a J2SE environment. A tree-like finite state machine (Tree Automata) that can be persisted is implemented. Workflow engine based on tree-like finite state machine (Tree-Automata) technology. Realized a breakthrough in business process customization development technology. The introduction of this technology makes it easier to realize business requirements that were previously difficult to achieve, and has truly achieved the goal of "demand-driven development". At present, there is no similar product in the industry that supports Tree Automata technology.

The HERCULES architecture provides a process template based on the Tree Automata workflow engine to implement complex business process processing. Some business requirements in online banking development (such as batch enterprise data maintenance, etc.) are very similar to the user's operating habits, but it is difficult to achieve through conventional Java development or conventional process configuration engines. The tree state machine provided by HERCULES can easily realize such business functions.

HERCULES introduced persistent workflow engine support. The HERCULES architecture business process control is implemented with Template. In enterprise-level development, Template is reflected on three levels and is based on programmatic process control. HERCULES provides a large number of mature business process abstract

components of this type, such as entry process templates, authorization process templates, and dynamic sequence process templates. Reference to these templates through the transaction configuration can meet the process control needs of more than 90 % of transactions; the process is based on XML Script. The definition of XML can simplify the complexity of the extended interface and standardize the process implementation. The sequence process, conditional process, loop process, calling sub-process XML Script and calling other templates can be realized through XML Script; based on the tree-based multi-level state machine workflow engine. Use the principle of Tree Automata to process State and Transition of complex processes; thus standardizing complex workflow control and realizing workflow in applications

HERCULES supports the configuration of data item inspection. By defining the data dictionary, HERCULES can immediately know the meaning of a certain data item, improve the readability of the application code, and facilitate continuous and rolling application development and maintenance. At the same time, the data dictionary provides the end-user with a uniform and international way of displaying error messages; Style is introduced. The data items in the data dictionary can correspond to one or more Styles (usually one-to-one) as needed. Style determines the validity checks rules for data items. In addition to the conventional effective checking methods, HERCULES introduces regular expression-based validity checking, which makes very complex validity checking very simple. The style also supports dynamic data checking, such as account Style checking will automatically check with the current user. The combination of Style technology and Rule technology enables developers to achieve a single data item inspection and cross-domain inspection requirements only through configuration. Not only improves development efficiency, but also ensures development quality, especially guarantees the application security of various key business systems.

2) BUSINESS TRANSACTION SECURITY FRAMEWORK

HERCULES architecture online payment user security software supports signature verification in PKCS1, PKCS7 [14], and other formats, integration with applications such as ASP, PHP, and operating systems such as Windows and Linux, with wide merchant adaptability. Block phishing websites to prevent users from entering by mistake. The bottom layer of the HERCULES architecture provides comprehensive support for security services and provides technologies in PKI / CA security systems such as SSL and CA. Through the PKI / CA security system, a series of security functions such as user authentication, server authentication, and data encryption transmission is realized. To ensure that the application system has design considerations for security. The application server not only prevents illegal visitors from accessing critical business applications (authentication) but also prohibits users with insufficient permission from performing operations. For sensitive data, such as operator login password, encryption protection, to prevent the information stored in

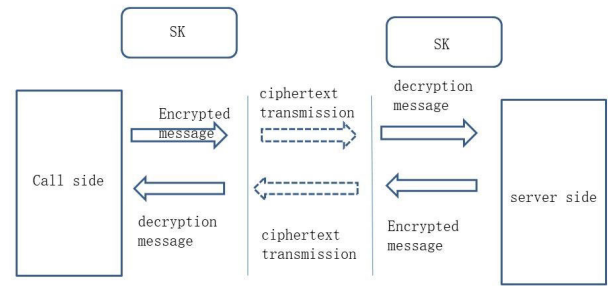


FIGURE 2. HERCULES architecture security design.

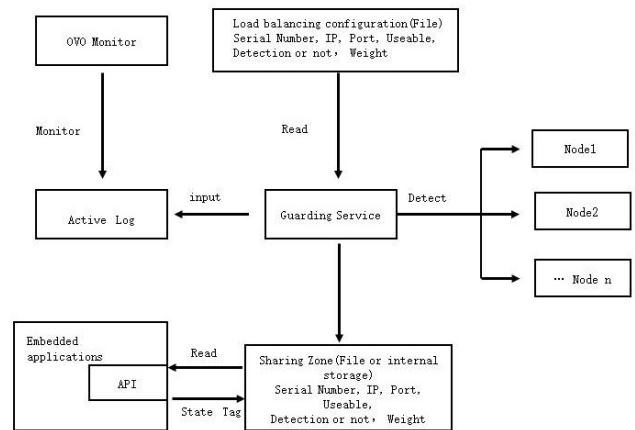


FIGURE 3. The load balancing of the HERCULES architecture.

clear text leakage. The use of encryption algorithms mainly includes SHA-1, MD5, SM3, and other hash algorithms and DES, AES, 3DES, and other symmetric encryption algorithms [15]. The security design is illustrated in figure 2.

C. SOFT LOAD BALANCING ALGORITHM

HERCULES architecture of load balance based on the above five features, refers to one or more corresponding application server, one or more additional software installation or provided to the application of common API calls to achieve access to multiple different nodes of the same services or interfaces. It is based on the specific environment, the advantages of simple configuration, flexible use, low cost, and can meet the demand of the majority of the load balancing, The current strategy is polling strategy and weight strategy, evenly distributed load to different servers and each server set a weight value, is an integer value of 1 to 65535, the greater the weight value, the more times to be distributed. The load balancing of the HERCULES architecture is illustrated in figure 3 function description of load balancing as Table2.

Load balancing for payment and business transactions refers to software load balancing, in one or more corresponding application servers, one or more additional software installation or provided to the application of common API calls to achieve access to multiple different nodes of the same services or interfaces. Each business through a weight value

TABLE 2. Function description of load balancing.

Application name	function
Guard Service	Load balancing profiles can be read and parsed periodically (such as once a day) to get information about each node of the load According to the node information, the connectivity of each node is probed.
	Based on the result of the probe, the node state is written to the Shared area.
	Periodic connectivity detection is performed according to the preset polling cycle
Embedded API	Read and parse the available nodes applied in the Shared area
	An available node is selected to be returned to the embedded application according to the load balancing strategy (equalization or weight), and the embedded application accesses the interface or service according to the returned node information.
	If the embedded application has an exception (disconnected connection, failed routing, timeout, etc.) when accessing the interface or service, the Shared area node is marked as unavailable through the API.
Embedded applications	Call the API to get the available node information
	To encapsulate communication and access services or interfaces based on node information
	If there is an exception when accessing the Service or interface, and the API is called to set the state of the node, this call is optional. If the Service polling time is long, it is recommended to join this call.

TABLE 3. Sample load balancing application configuration file.

```
[GLOBAL]
G_TIMEOUT=10
CStatInfoFile=$LDBLANCEHOME/CStatus.info
[GRPNAME_A] #Group name, this value needs to be specified
when calling GeDtnAddr function in API
BAKGRP = GRPNAME_B #Backup group name. If there is no
backup group, fill in the blank. When all NODEs in this group
are unavailable, it will try to obtain NODE from the backup
group.
TYPE = SOCKET #type, there are two types of SOCKET and
MQ, their corresponding NODE value format is different
TIMEOUT = 5 #connection timeout seconds
NODE = 11.1.3.56: 3456: 1: Y #The meaning of its value is IP:
PORT: WEIGHT: Whether to detect
NODE = 11.1.3.57: 3456: 1: Y #Where WEIGHT is the weight
and the value is a non-negative integer, set to 0 will not use the
NODE
NODE = 11.1.3.58: 3456: 1: Y #API selects the probability of
each available NODE according to their weight ratio, whether
it needs to detect whether it is represented by Y or N, when
there are multiple available NODEs listed side by side, there
are multiple When NODE is available side by side.
[GRPNAME_B]
BAKGRP = GRPNAME_A
TYPE = SOCKET
TIMEOUT = 5
NODE = 11.1.5.56: 3456: 1: Y
NODE = 11.1.5.57: 3456: 1: Y
NODE = 11.1.5.58: 3456: 1: Y
[GRPNAME_M] #Group name, this value needs to be speci-
fied when calling GeDtnAddr function in API
BAKGRP = GRPNAME_N #Name of the backup group. If
there is no backup group, fill in the blank. When all NODEs
of this group are unavailable, it will try to obtain NODE from
the backup group.
TYPE = MQ #type, there are two types of SOCKET and MQ,
their corresponding NODE value format is different
TIMEOUT = 5 #connection timeout seconds
NODE = QM1: CHL1: TCP: 11.2.68.16: 40209: 1: Y #mq type
NODE values ??are divided into 7 parts, which are:
NODE = QM1: CHL2: TCP: 11.2.68.16: 40219: 1: Y #Queue
manager name, channel name, connection type,
IP, PORT, weight, whether detection is needed, where
WEIGHT is the weight and the value is a non-negative integer,
set to 0 will not use the NODE, #API selects the probability of
each available NODE, and allocates according to their weight
ratio. Or N indicates that they are listed side by side when there
are multiple NODEs available, and side by side when there are
multiple NODEs available.
[GRPNAME_N]
BAKGRP = GRPNAME_M
TYPE = MQ
TIMEOUT = 5
NODE = QM2: CHL1: TCP: 11.2.68.26: 40209: 1: Y
NODE = QM2: CHL2: TCP: 11.2.68.26: 40219: 1: Y
[GRPNAME_T]
BAKGRP =
TYPE = TUXEDO
TIMEOUT = 5
NODE = 11.1.6.56: 14561: TtestServ: 1: Y # Relative to the
SOCKET type, the parameter "service name" was added
NODE = 11.1.7.56: 14562: TtestServ: 1: Y #Communication
detection program for health check
[GRPNAME_P]
BAKGRP =
TYPE = WTC
TIMEOUT = 5
NODE = 11.1.8.56: 3456: WtcTestServ: 1: Y # Relative to the
SOCKET type, the parameter "service name" has been added
NODE = 11.1.9.56: 3456: WtcTestServ: 1Y #Communication
detection program for health check
```

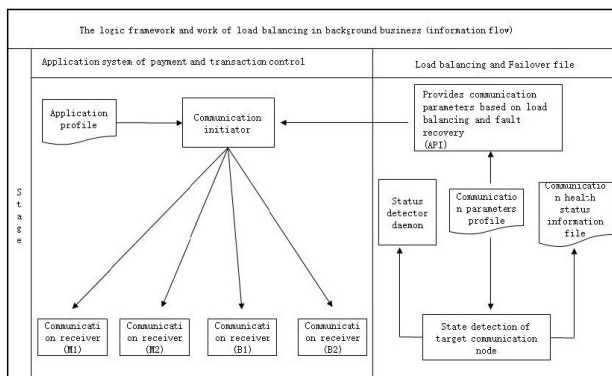


FIGURE 4. Working principle of load balancing.

between an API for the load of the algorithm. The working principle of load balancing is illustrated in figure 4.

Table 3 Sample Load Balancing Application Configuration File. The following is a core algorithm code snippet with a lot of loads. Its main function is to randomly calculate and compare its weight value according to different communication types, such as a socket, HTTP, etc., and return the maximum weight value that falls within the given interval of random numbers.

Algorithm 1 Load Balancing Core Algorithm Code Snippet**Require:** *NodeBean*, *String*, *Stringtype*.**Ensure:** *String[]* *returnNode*

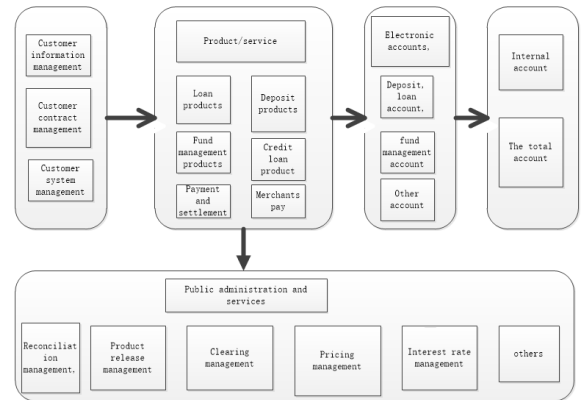
```

1: List < Map > nodes = nodeBean.getNodes(); All
   nodes within a single group
2: double gAllWeight = 0; The sum of the type weight
   values
3: Boolean isScanSucc = false; whether the node is avail-
   able
4: Double random = Math.Thersandom(); random number
   used in the random algorithm
5: if type.equalsIgnoreCase("socket") then
6:   SocketBeans sb = null
7:   List < SocketBean usable > = new ArrayList <
   SocketBean > () calculates the available nodes
8:   List < Double > intervalWeight = new ArrayList <
   Double > () the interval range value of the weight
9:   IntervalWeight.Add(0.0) the initial value of the weight
   interval
10:  Double count = 0
11:  for Map<String, Object> map: nodes do
12:    for Map.Entry<String, Object> entry:
   map.entrySet() do
13:      sb = (SocketBean)entry.getValue()
14:      isScanSucc = IsOrUseNode(sb.getNameNode())
15:      if isScanSucc then
16:        gAllWeight = gAllWeight +
   Double.parseDouble(sb.getWeight())
17:        usable.add(sb)
18:        count = count +
   Double.parseDouble(sb.getWeight())
19:        intervalWeight.add(count)
20:      end if
21:    end for
22:  end for
23: end if
24: Random algorithm: random number = random number
   * weight sum, compare the random number in which the
   interval, then return the node of this interval
25: Random = random * gAllWeight random number =
   random number less than 1 * sum of available node
   weights
26: for int i = 0; i < usable.size(); i++ do
27:   sb = usable.get(i)
28:   if intervalWeight.get(i) <= random && random <=
   intervalWeight.get(i + 1) then
29:     returnNode = new String[3]
30:     returnNode[0] = sb.getIp()
31:     returnNode[1] = sb.getPort()
32:     returnNode[2] = sb.getWeight()
33:     return returnNode
34:   end if
35: end for

```

III. INTELLIGENT ANALYSIS OF HERCULES ARCHITECTURE INTELLIGENT ONLINE BANKING SYSTEM BUSINESS

In the previous chapter, the HERCULES architecture component model, software load balancing, and architecture engine templates have been explained. In this section, based on the HERCULES architecture, intelligent online banking deposits and loans and business transaction security will be discussed. The overall business logic architecture is illustrated in figure 5.

**FIGURE 5.** The total business logic diagram of intelligent online banking.**A. INTELLIGENT ANALYSIS OF DEPOSIT BUSINESS**

Machine learning is a branch of artificial intelligence. It uses algorithms to the learning experience to realize automatic deposit and loan business without human intervention. Innovative smart deposits can help banks reduce costs and improve service capabilities. At present, smart deposits in banks are mainly divided into three kinds. The first kind is to deposit and withdraw, similar to the current period. The second kind is to pay interest regularly, that is, monthly interest payment, principal renewal. The third is tantamount to calculate interest on a file, the longer the term, the higher the interest rate. Each interest rate is not analyzed, and you can refer to the interest rate of each bank for specifics. This section focuses on analyzing the combination optimization decision of the deposit business combined with machine learning neural network technology.

The background of the HERCULES architecture divides the customer information sample into two types of normal customers and risk customers. Then according to the customer's information indicators $X_i = (x_1, x_{i2}, \dots, x_{i11})$. Here are 11 tentative sample samples. The normalized values based on these indexes are illustrated in Table 4.

Establish a BP neural network model [16] based on user information

$$Y = g \left(V \left[f \left(W^T X \right) + B_1 \right] + B_2 \right) \quad (6)$$

where Y is a smart deposit interest rate obtained based on the user information according to the model, and the activation

TABLE 4. User information data.

Index	value range	meaning
X1	0,1	Gender (0: female; 1: male)
X2	0,1,2	Marital status (0 unmarried; 1 married; 2 other)
X3	0,12,3,4	Education (0 junior high school or below; 1 high school or secondary school; 2 junior college; 3 university; 4 postgraduate or above)
X4	0,1,2,3,4	The nature of the housing (0 has property rights; 1 lives with parents; 2 units are allocated; 3 mortgages; 4 leases; 5 others)
X5	0,1	With or without children (0 none; 1 yes)
X6	0,1,23	Position (0 senior management staff; 1 intermediate management staff; 2-general staff; 3 other)
X7	0,1,23	Title (0 advanced; 1 intermediate; 2 junior; 3 others)
X8	0,1,2,3,4J	The nature of the unit (0 joint-stock enterprises; 1 state-owned enterprises; 2 foreign-funded enterprises; 3 private enterprises; 4 institutions and institutions; 5 other)
X9	0,1	Account in our bank (0 none; 1 yes)
X10	0,1	Loan history (0 none; 1 yes)
X11	0,1,23	Card status (0 with dragon card 1 with other bank credit card 2 without card 3 and others)

function of the model is

$$f(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \quad (7)$$

The model uses the square error function $E_p = \frac{1}{2}(y - \bar{Y})^2$, \bar{Y} For the desired output, $B_1 = (b_1, b_2, \dots, b_{10})^T$ and $B_2 = (b)$ These are the threshold vectors from the first layer neural network to the second layer and the threshold vectors from the second layer to the third layer. If $E_p = 0$, it means that the error between the model output and the actual is very small, assuming that the model output is

$$\hat{y}_p = (\hat{y}_1^p, \hat{y}_2^p, \dots, \hat{y}_k^p) \quad (8)$$

$\hat{y}_p \in \bar{Y}$, the actual output is:

$$y_p = \text{left}(y_1^p, y_2^p, \dots, y_k^p \text{ right}) \quad (9)$$

$y_p \in \bar{Y}$, then the derivative of E_p is as follows

$$\begin{aligned} \frac{\partial E_p}{\partial \hat{y}_j^p} &= \frac{\partial \left(\frac{1}{2} \sum_{i=1}^k (\hat{y}_i^p - y_i^p)^2 \right)}{\partial \hat{y}_j^p} \\ &= \frac{\partial \left(\frac{1}{2} (\hat{y}_j^p - y_j^p)^2 \right)}{\partial \hat{y}_j^p} \\ &= \hat{y}_j^p - y_j^p \end{aligned} \quad (10)$$

The specific training of the model will not be discussed in this section.

B. INTELLIGENT ANALYSIS OF LOAN BUSINESS

In the consumer credit market, in terms of small loans, p2p loans are complementary to bank loans [17], [18]. The smart loan business is somewhat similar to the p2p business. The main attributes considered by the loan business are the borrower repayment ability, The willingness to repay, historical repayment records, credit support, legal liability for repayment, etc., depending on the amount of the user's loan, whether the bank can bear the risk reasonably, bank users are divided into normal, concerned, subordinate, and suspicious. Normal, the financial state is very good, the risk is generally, that is, the normal state; if the credit is not good, the financial state is insufficient, the risk is high, that is, the suspicious state. This article is mainly based on the analysis and modeling of white-collar financial factors and credit payment factors. User's risk level adopts the method of decision tree classification, as shown in Figure 6.

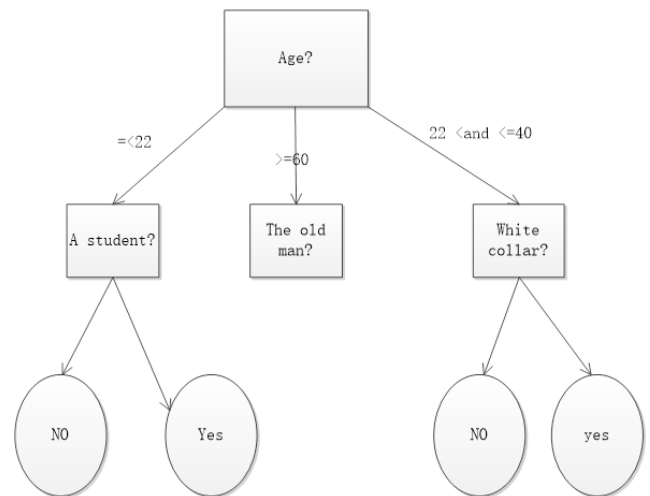


FIGURE 6. Loan business decision tree algorithm.

Decision tree algorithm such as

$$I(s_1, s_2, \dots, s_m) = - \sum_{i=1}^m p_i \log_2(p_i) \quad (11)$$

The sample set of the data set S records the attributes of the user, I expects the information output, and p_i is the probability

of outputting for each attribute label. The indicator to measure the expected output is expressed by entropy:

$$E(A) = \sum_{j=1}^v \frac{s_{1j} + s_{2j} + \dots + s_{mj}}{s} I(s_{1j}, s_{2j}, \dots, s_{mj}) \quad (12)$$

The item $\frac{s_{1j} + s_{2j} + \dots + s_{mj}}{s}$ is the weight of the j th subset, which is equal to the sample in the subset s_j . The number is divided by the total number of attribute samples in S . Given a subset s_j , the probability of expected information is

$$I(s_{1j}, s_{2j}, \dots, s_{mj}) = - \sum_i^m p_{ij} \log_2(p_{ij}) \quad (13)$$

p_{ij} is the probability that the sample in s_j belongs to the parent class. Entropy is a concept of physics and is used to reflect the degree of hybridization of a system. Shannon introduced entropy into information theory in 1948. Specific analysis of entropy will not be discussed in this section.

C. BUSINESS TRANSACTION SECURITY ANALYSIS

The security functions of business transactions based on the HERCULES architecture are illustrated in Table 5. Background transaction security processing is illustrated in Figure 7 and Figure 8.

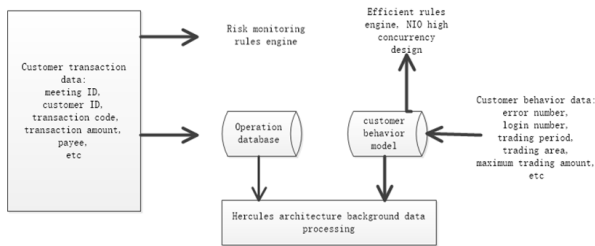


FIGURE 7. HERCULES architecture handles security behind the scenes.

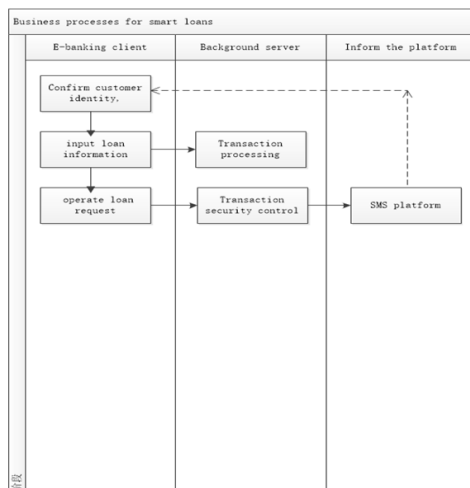


FIGURE 8. Safe processing of loans.

TABLE 5. Business transaction security function analysis.

Risk model	Risk model library	Manage anti-fraud models and credit scoring models; Calculate logic maintenance and parameter configuration for each model through the rule engine.
Risk strategy	Fraud strategy	Classify fraud strategies, provide addition, deletion, modification, and investigation operations; fraud strategies are mainly divided into 9 categories: monitoring category, account list, IP list, account list, and document list
Risk list	Account list management	Risk attributes such as list of risk accounts, accounts, whether or not they are blacklisted;
	IP list management	Risk IP list, mainly including data types such as list type (black, white, gray), risk factors;
	Phone list management	Risk phone list, mainly including phone number, phone type, list type (black, white, gray), risk factors and other data attributes.
	Document list management	Risk document list, mainly including document type, document number, list type (black, white, gray), Data attributes such as risk factors;
	Customer number list management	Risk customer number list, mainly including customer number, list type (black, white, gray), risk factors and other data attributes;
	Device list management	List of risk client devices, mainly including unique information of the device, list type (black, white, gray), risk factors and other data attributes
	MAC list management	Risk client MAC address, mainly including device MAC address, list type (black, white, gray), risk factors and other data attributes;

The risk model refers to the random forest algorithm in the literature [19] and the neural network model of transaction fraud detection, as well as behavior detection algorithm [20]. The specific algorithm part will not be analyzed in detail in this article. Some scholars have adopted Deep Representation

Learning [21] and cross-feature learning hybrid models [22] in transaction fraud learning. The future research and discussion of such algorithms when doing credit card business in online banking systems will be conducted.

IV. THE BUSINESS IMPLEMENTATION OF THE INTELLIGENT ONLINE BANKING SYSTEM IS COMPARED WITH SIMILAR SYSTEMS

In the HERCULES architecture, OSGI's WEB and JAR methods are adopted. OSGi has become one of the most popular system architecture methods on the Internet, desktop programs, mobile devices, and enterprise-level applications. The HERCULES platform can be accessed through HERCULES Dynamic WEB The Console performs real-time management of dynamic module release, update, and start/stop. HERCULES can also provide open APIs for management and control of the HERCULES operating environment, such as JVM memory, threads, communication, etc. The HERCULES platform can use the HERCULES WEB Console to parameterize dynamic modules During run-time modification, dynamic parameters can be determined by configuration to determine parameter types and read and write permissions. The HERCULES platform can publish applications to multiple servers at the same time through the HERCULES WEB Console, which greatly saves operation and maintenance time.

In the previous chapter, the business and research methods of the HERCULES architecture have been analyzed. In this section, the design and implementation of the smart deposit business and white-collar loan business will be discussed in detail in conjunction with the research and analysis of the previous section, and comparison with similar systems.

A. SMART DEPOSIT BUSINESS MODEL DESIGN

For the deposit business, the bank identified the customer based on the transaction data previously collected, and the customer accepts the monetary funds to be deposited into the bank entity's account system [23]. The starting amount of smart deposit is temporarily set at RMB 100, but the bank has the right Regulations, regulatory policies, and business requirements adjust the starting point amount and inform customers that the adjustment will take effect on the date determined by the Bank's notice; smart deposits collect funds, and the effective balance in the current settlement account meets the agreed conditions of the service agreement and is transferred When entering the value-added service account for management, a collection of funds is formed; in the holding period of the collected funds, the day of transfer of each collected funds to the value-added service account is the value date, and the day of transfer out of the value-added service account As the termination date, the period between the two is the holding period of the collected funds. For banks, the profitability is defined by the return on equity and the net interest margin, etc. [24], [25].

The expiration of the collected funds, the early withdrawal of the collected funds, and the termination of the smart

deposit service are all situations that result in the transfer of the collected funds from the value-added service account; value-added income, the system pays the value-added income of the collected funds to the customer. Clearance; early withdrawal / early withdrawal of collected funds, when the effective balance of the customer's current settlement account cannot meet the customer's demand for funds such as withdrawals, transfers, and consumption payments, within the limit of the total amount of collected funds in the value-added service account, The collected funds will be transferred from the value-added service account to the current settlement account in accordance with certain allocation rules; the normal termination of the single collected funds, that is, the collection of the collected funds to Termination of the period. The Bank will accrue value-added income to the customer for the single fund in accordance with the rules stipulated in the service agreement, and other collected funds continue to obtain value-added income; abnormal termination of all collected funds, including customer applications and the Bank agreement to terminate, Unilateral termination due to customer's breach of contract (the customer was deducted for part or all of the collected funds due to judicial or other competent authorities taking coercive measures If the customer defaults on all funds) and other circumstances that cause the abnormal termination of all collected funds, the system will treat all the collected funds of the customer as termination.

The description of the core business process. The user logs in direct online banking and clicks on the "smart deposit" function. If the contract is not signed, the smart deposit homepage is displayed. The customer clicks "I always want to deposit". After the customer agrees to the service agreement, the transaction password can be checked and the contract can be signed successfully If the contract has been signed, My Smart Deposit page is displayed. Click "Cancel". After the system verifies the transaction password of the contracted account, you can cancel the contract of the smart deposit product signing agreement on the current account. The customer clicks the smart deposit balance query link, the page Display the detailed balance information of each pooled funds in the smart deposit value-added account under the current electronic account and can try to calculate the current earnings of each pooled funds that are not due or transferred in advance (the pool The current income of the fund = the principal of the collected funds * the highest interest rate that is closest to the actual holding period * the number of days of the actual holding period ÷ 360, the calculation result is provided by the Internet account system).

In the online banking system [26], my smart deposit page displays "cumulative historical income", click the smart deposit details query link, the page displays the transaction details of all smart deposit accounts, the query time displays the latest month by default, and the transaction type defaults to All, when the electronic account is in the state of being verified, the customer clicks on the contracted smart deposit to prompt the customer: "Your electronic account is in the state of being verified and cannot sign the smart deposit,

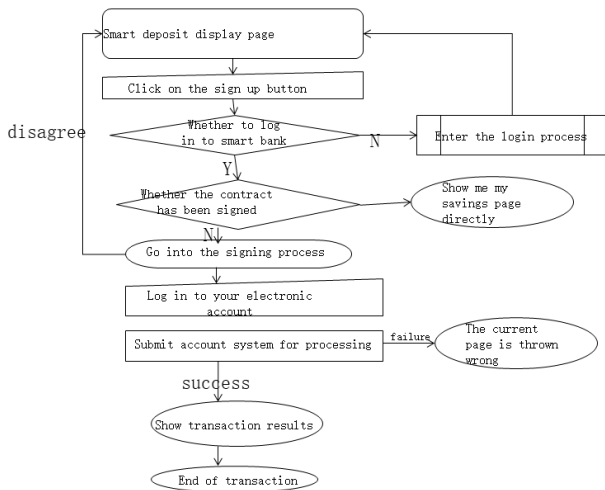


FIGURE 9. Smart deposit business process.

please deposit as soon as possible to verify and activate” The customer has not signed a smart deposit. When the deposit is to be realized, the signing process is shown in Figure 9. Banks provide accounting and accounting systems to pay customers. These systems are point-to-point transactions [27].

B. BUSINESS MODEL DESIGN OF WHITE COLLAR LOAN

From the perspective of the loan market, high-quality customers will become the focus of competition for commercial banks. White-collar personal loans of the intelligent online banking system in this paper refer to the borrowers who meet the loan requirements of commercial banks to solve the borrower’s consumption needs or temporary capital needs. Personal unsecured pure credit loan. The user’s credit rating is divided into multiple levels according to traditional standards, such as excellent, good, poor, etc. The correlation between rating levels refers to the literature [28]. The qualified staff of various departments of the government in the previous year’s assessment, the staff of institutions, electric power, tobacco, petroleum, banking, insurance, and securities industry officially qualified for the previous year’s assessment, teachers of large, middle and primary schools, doctors and nurses in hospitals Long, military platoon leader (inclusive) or above officers, married, and spouse with permanent residence in the bank’s business area, stable working income and other groups. Loan amount, white-collar personal loan amount The minimum limit for a single-household loan is 100,000 yuan (including), The maximum should not exceed 1 million yuan (inclusive). According to the applicant’s circumstances (administrative level and family annual total income and other comprehensive conditions) to determine the loan amount of loan. The term of the loan is a one-time withdrawal and repayment, and the loan with monthly interest payment does not exceed 12 months.

Loans with one-time withdrawals and monthly repayments of principal and interest at a maximum of 36 months are allowed. For high-quality customers with stable positions

or positions or payroll clients of our bank, a three-year credit period can be granted. The term of a single loan does not exceed 12 months. The white-collar credit loan can be used for withdrawals and repayment parties’ one-time withdrawals, and monthly payments of interest are due. Proof of relationship; proof of academic qualifications, professional titles, qualifications for practicing; proof of work and income; proof of assets, or other financial resources. The design process of the core business of white-collar loans:

1) Customers enter the online banking system, register and log in. Create an electronic account, and bind the bank or other bank card number (non-account).

2) Customers choose white-collar loan products. Do simple questionnaire surveys, and conduct scoring tests. Those who pass can enter the next step. Those who fail to meet the standards are eliminated.

3) The customer enters the following contents in order: credit information and loan details. After the information is entered, the system will automatically calculate the loanable range and the executive interest rate range, and submit the loan application.

4) The client manager logs in to the background management system and conducts the preliminary review: if the information provided by the client is more authentic, the client manager can choose to conduct a return visit.

5) The follow-up credit review and loan process will remain unchanged. Customers can go to the counter to repay, transfer online to the repayment account, or transfer to the bound card through the electronic account, and the bound card is the repayment account.

C. MACHINE LEARNING ALGORITHM K-MEANS IMPLEMENTATION

The algorithm of the smart deposit is based on the k-means algorithm, which divides customers into different dimensions according to the basic data of bank customers, such as customer ID, name, registration date, age, gender, phone number, mailing address, and some key feature information, such as age, education, and income.

Age is divided into different stages, and less than 20, 20 to 30 years old, 30 to 40 years old, 40 years of age or older, according to the customer’s education program is divided into higher education, general education, and low education. Through the mailing address to determine whether first-tier cities or second-tier cities, etc. When K-means algorithm in clustering according to different dimensions to certain interest rates on deposits and loans. The algorithm steps are as follows:

1) The root selects the initialized k sample as the initial clustering center $a = a_1, a_2, \dots, a_k$

2) Calculate the distance from each sample x_i in the data set to K clustering centers and divide it into the class corresponding to the cluster center with the smallest distance;

3) For each category a_j , recalculate its clustering center $a_j = \frac{1}{|c_j|} \sum_{x \in c_j} x$ (that is, the center of mass of all samples belonging to this category);

4) Repeat steps 2 and 3 above until a certain abort condition is reached (number of iterations, minimum error change, etc.). The essence of the K-means algorithm is to calculate the Euclidean distance. $D = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$.

Algorithm 2 K-Means Algorithm

Require: $x, rain, k, max_iter$

Ensure: $cluster_center$ Return the final cluster center point

```

1:  $num\_iter = 0$ 
2: Initial cluster center
3:  $cluster\_center = x, rain[: k]$ 
4:  $pre\_cluster\_center = copy.deep\_copy(cluster\_center)$  Last cluster center point
5: Start iteration
6: while  $num\_iter < max\_iter$  : do
7:   Temporary variables
8:    $clusters\_data = DictionaryClustersubscript : coordinates$ 
9:   for  $iinx, rain$  : do
10:     $cluster\_dists = []$ 
11:    for  $index, clusterin enumerate(cluster\_center)$  : do
12:      $distance = get\_distance(i, cluster)$ 
13:      $cluster\_dists.append((index, distance))$  distance from each sample to the center point
14:      $cluster\_dists.sort(key = lambda x : x [1])$  ascending
15:      $min\_index, min\_dist = cluster\_dists [0]$  Take the closest
16:   end for
17:   if  $min\_index \text{ not in } clusters\_data$  : then
18:      $clusters\_data[min\_index] = []$ 
19:   end if
20:    $clusters\_data[min\_index].append(i)$  Add data to temporary variables
21: end for
22: Update cluster center point
23: for  $index \text{ in } clusters\_data$  : do
24:    $cluster\_center[index] = calc\_mean(clusters\_data[index])$ 
25: end for
26: if  $pre\_cluster\_center == cluster\_center$  : then
27:   break If the cluster center point no longer changes, then end
28: else
29:    $pre\_cluster\_center = copy.deep\_copy(cluster\_center)$  Copy it
30: end if
31: end while

```

D. COMPARISON OF THE REALIZATION OF INTELLIGENT ONLINE BANKING AND SIMILAR SYSTEMS

The intelligent online banking system based on HERCULES architecture has obvious advantages compared with similar systems developed by other architecture on the market, such as SPRING MVC or Struts, as shown in table 6.

Algorithm 3 K-Means Algorithm $get_distance$

Require: X, Y

Ensure: $np.sum((X - Y) ** 2) * 0.5$

Algorithm 4 K-Means Algorithm $calc_mean$

Require: X

Ensure: $list_mean$

```

1:  $l = len(X[0])$ 
2:  $list\_mean = []$ 
3: for  $iinrange(l)$  : do
4:    $s = 0$ 
5: end for
6: for  $jinX$  : do
7:    $s+ = j[i]$ 
8:    $m = s / len(X)$ 
9:    $list\_mean.append(m)$ 
10: end for

```

TABLE 6. The realization comparison between intelligent online banking and similar systems.

Features	HERCULES	other products
The IOC technology	IOC based development	Development based on traditional XML configuration method
Multi-channel support	To fully support	Lack of support layer outside the J2EE container
Message handling	A very powerful	Support limited types
J2SE support	Full support for J2SE sessions, transactions, clusters, and more	Does not support
SOA	Support and practical application	Part of the support
Distributed event mechanism	There are	There is no
Heat release	There are	There is no
Standard flow template	Provides a number of mature process templates	Basically not
XMLScript process	Support rich flow control elements	Over-reliance on XML customization processes is difficult to debug and maintain
Tree-automata workflow engine	To fully support	Not supported or supported only
System-level process	Dynamic chain of responsibility	Does not support
Intelligent business model	Fusion of machine learning algorithms	Does not support

V. CONCLUSION

A. THE INNOVATION POINT OF INTELLIGENT NET SILVER

The intelligent online banking system based on the HERCULES architecture is implemented to meet the flexibility and friendliness of business users for the formulation and maintenance of business rules and implements a series of complete functions such as template definition, update, version control management, and fixed-point hot release. At the same time combined with the technology of artificial

intelligence machine learning when designing the business module, this system has the following characteristics:

(1) “Lego building blocks” software development method. Through the HERCULES framework, software development engineers can pay more attention to business integration without having to consider the underlying logic and implementation, completely liberating system designers, and facilitate the flexible development of multi-domain customized products. Developing software products (projects) is like “building blocks” “, And because the dynamic module technology uses a micro-kernel mechanism, it can also ensure the stability and efficiency of the system. The HERCULES framework can allow designers to complete the system design work more standardized, and it is easier to divide the module design business architecture first and then conduct a detailed module design. All designers are based on a design specification to avoid the blooming of flowers. Each designer has its own set of design methods, and the disadvantages of design are universal.

(2) Intelligent and automated deposit and loan sub-module. Due to the strict modularity of the system modules developed based on HERCULES, the most cutting-edge neural network technology is added. Compared with the traditional development method, the organization, reuse and expansion of system modules become easier, and each system can look at one Business models, smart deposits, white-collar loans, and other related businesses can be modeled based on customer data, and automatically manage customer deposits based on machine learning algorithms.

B. SHORTAGE WITH FUTURE WORK

Insufficient: HERCULES architecture intelligent online banking system requires users and business analysts to conduct demand analysis based on the experience of the financial industry and relevant business knowledge of online banking, so the experience of business personnel and analysts, financial expertise, basic calculation knowledge, risk awareness, etc There are very high requirements. In terms of open source and MVC control modes, the system still has many deficiencies. The development of intelligent online banking systems based on the HERCULES architecture requires developers to learn the HERCULES architecture first. The plug-in process should be familiar, and also requires developers to have the following foundation, object-oriented design, and programming; Java language foundation; SOA service architecture [29], [30]; OSGI foundation [31]; database application; J2EE foundation Servlet [32], JSP, familiar with the business knowledge of a J2EE application server and online banking. In the implementation of smart deposit codes, developers need to understand machine learning algorithms, and the requirements for developers are relatively high.

Outlook: This paper proposes that the fusion of smart deposits, white-collar loan business models and machine learning belongs to the application of machine learning in the field of financial portfolios. The applications of machine learning in the financial field such as algorithmic

transactions, fraud recognition, and customer sentiment analysis are all worth discussing In the future, in-depth research will be done in online banking adaptive systems, customer sentiment analysis, and financial time series analysis.

Summary of This Section: The design and implementation of the online banking system based on the HERCULES architecture guarantee system maintainability, scalability, reliability, etc., which greatly reduces the operation and implementation costs of the bank’s online banking system, and at the same time make the online banking system begin to change to intelligent.

REFERENCES

- [1] C.-S. Lin, S.-H. Chiu, and T.-Y. Lin, “Empirical mode decomposition-based least squares support vector regression for foreign exchange rate forecasting,” *Econ. Model.*, vol. 29, no. 6, pp. 2583–2590, Nov. 2012.
- [2] P. D. Yoo, M. H. Kim, and T. Jan, “Machine learning techniques and use of event information for stock market prediction: A survey and evaluation,” *IEEE Comput. Soc.*, to be published.
- [3] F. Wang, K. Dong, and X. Deng, “Algorithmic trading system: Design and applications,” *Frontiers Comput. Sci. China*, vol. 3, no. 2, pp. 235–246, Jun. 2009.
- [4] P. E. Tsinaslanidis and D. Kugiumtzis, “A prediction scheme using perceptually important points and dynamic time warping,” *Expert Syst. Appl.*, vol. 41, no. 15, pp. 6848–6860, Nov. 2014.
- [5] B. Vanstone and G. Finnie, “An empirical methodology for developing stockmarket trading systems using artificial neural networks,” *Expert Syst. Appl.*, vol. 36, no. 3, pp. 6668–6680, Apr. 2009.
- [6] L. Guo, “Regulating investment robo-advisors in China: Problems and prospects,” *Eur. Bus. Org. Law Rev.*, vol. 21, no. 1, pp. 69–99, Mar. 2020.
- [7] W. Zheng, “Application and key technology of artificial intelligence in customer service system,” *Telecommun. Sci.*, vol. 34, no. 12, pp. 84–91, 2018.
- [8] G. Saon, G. Kurata, T. Sercu, K. Audhkhasi, S. Thomas, D. Dimitriadis, X. Cui, B. Ramabhadran, M. Picheny, L.-L. Lim, B. Roomi, and P. Hall, “English conversational telephone speech recognition by humans and machines,” 2017, *arXiv:1703.02136*. [Online]. Available: <http://arxiv.org/abs/1703.02136>
- [9] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.
- [10] W. Kun, “Development and supervision of banking industry in the era of FinTech,” *China Markets, Tech. Rep.*, Apr. 2020.
- [11] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.
- [12] L. Tianxin, Y. Xiaoping, and W. Liang, “Research and development of financial transaction system based on reinforcement learning,” (in Chinese), *J. Softw.*, vol. 30, no. 3, pp. 355–374, 2019.
- [13] J. Moody and M. Saffell, “Learning to trade via direct reinforcement,” *IEEE Trans. Neural Netw.*, vol. 12, no. 4, pp. 875–889, Jul. 2001.
- [14] V. Klima, “Side channel attacks on CBC encrypted messages in the PKCS7,” *Cryptol. Eprint Arch., Tech. Rep.* 2003, 2003.
- [15] G. Singh and S. Supriya, “A study of encryption algorithms (RSA, DES, 3DES and AES) for information security,” *Int. J. Comput. Appl.*, vol. 67, no. 19, pp. 33–38, Apr. 2013.
- [16] K. Kohara, T. Ishikawa, Y. Fukuhara, and Y. Nakamura, “Stock price prediction using prior knowledge and neural networks,” *Int. J. Intell. Syst. Accounting, Finance Manage.*, vol. 6, no. 1, pp. 11–22, Mar. 1997.
- [17] H. Tang, “Peer-to-peer lenders versus banks: Substitutes or complements?” *Rev. Financial Stud.*, vol. 32, no. 5, pp. 1900–1938, May 2019.
- [18] P. Bharadwaj, W. Jack, and T. Suri, “FinTech and household resilience to shocks: Evidence from digital loans in Kenya,” *Nat. Bur. Econ. Res., Tech. Rep.*, 2019.
- [19] S. Xuan, G. Liu, Z. Li, L. Zheng, S. Wang, and C. Jiang, “Random forest for credit card fraud detection,” in *Proc. IEEE 15th Int. Conf. Netw., Sens. Control (ICNSC)*, Zhuhai, China, Mar. 2018, pp. 1–6.
- [20] L. Zheng, G. Liu, C. Yan, and C. Jiang, “Transaction fraud detection based on total order relation and behavior diversity,” *IEEE Trans. Comput. Social Syst.*, vol. 5, no. 3, pp. 796–806, Sep. 2018.

- [21] Z. Li, G. Liu, and C. Jiang, "Deep representation learning with full center loss for credit card fraud detection," *IEEE Trans. Comput. Social Syst.*, vol. 7, no. 2, pp. 569–579, Apr. 2020.
- [22] H. Wu and G. Liu, "A hybrid model on learning cross features for transaction fraud detection," in *Proc. 19th Ind. Conf. Data Mining (ICDM)*, New York, NY, USA, Jul. 2019, pp. 1–5.
- [23] E. S. Votaw and J. J. Siekman, "Intelligent ATM check image deposit engine," U.S. Patent 8 510 221, Aug. 13, 2013.
- [24] F. Sönmez and S. Bübül, "An intelligent software model design for estimating deposit banks profitability with soft computing techniques," *Neural Netw. World*, vol. 25, no. 3, p. 319, 2015.
- [25] W. U. Ani, D. O. Ugwunta, I. J. Ezeudu, and G. O. Ugwuanyi, "An empirical assessment of the determinants of bank profitability in Nigeria: Bank characteristics panel evidence," *J. Accounting Taxation*, vol. 4, no. 3, pp. 38–43, 2012, doi: [10.5897/JAT11.034](https://doi.org/10.5897/JAT11.034).
- [26] J. Jagtiani and C. Lemieux, "Do FinTech lenders penetrate areas that are underserved by traditional banks?" *J. Econ. Bus.*, vol. 100, pp. 43–54, Nov. 2018.
- [27] C. Lagarde, "Central banking and FinTech: A brave new world," *Innov. Technol., Governance, Globalization*, vol. 12, nos. 1–2, pp. 4–8, Jul. 2018.
- [28] J. Jagtiani and C. Lemieux, "The roles of alternative data and machine learning in FinTech lending: Evidence from the LendingClub consumer platform," *Financial Manage.*, vol. 48, no. 4, pp. 1009–1029, Dec. 2019.
- [29] C. Lin, J. Yao, H. Zhang, D. Ji, and G. Ji, "Research on shipping E-commerce platform based on J2EE and SOA development technique," in *Proc. Int. Conf. Service Syst. Service Manage. (ISSSM)*, 2010, pp. 10–11.
- [30] T. Rademakers, *Jos Dirksen Open Source ESB in Action*. Shelter Island, NY, USA: Manning Publications, 2009, pp. 23–29.
- [31] The OSGi Alliance, *Osgi Service Platform, Release 3*. IOS Press, 2013, pp. 400–405.
- [32] J. Hunt and C. Loftus, *Guide to J2ee: Enterprise Java*. Springer, 2012, pp. 300–303.



GUANGSHENG LUO (Graduate Student Member, IEEE) was born in Huangshi, Hubei, China, in 1982. He received the bachelor's degree in computer science and technology from the Huazhong University of Science and Technology, in 2005, and the master's degree in software engineering from Fudan University, in 2015, where he is currently pursuing the Ph.D. degree. From 2005 to 2013, he was with Shanghai Huawei Technologies Company Ltd., China Telecom Shanghai

Ideal Technology Company Ltd., the Neusoft Group, and other well-known enterprises as a Senior Project Manager, a Senior System Architect, and other positions. From 2013 to 2016, he was with Beijing Kelan Software Systems Company Ltd., as the Project Director, the Technical Director, a Large Data Engineer, and other positions. He is currently the Senior Project Director, a Senior System Architect, the CTO, and a Big Data Engineer. National key special development team members, Key Laboratory of Intelligent Information PhD in Shanghai, banking and other financial core system development experts, Senior Architect of medical systems. He is engaged in the development of more than ten years, is rich in large-scale project development and management, financial, medical, and government industry experience. He is also a proficient in a variety of development languages and large-scale system architecture design.



WENWEI LI (Graduate Student Member, IEEE) was born in Xiangfan, Hubei, China, in 1985. He is currently pursuing the Ph.D. degree. His research interests include artificial intelligence, cloud computing, and distributed systems. He has published an article in the workshop of IJCAI 2019. His balance theory is a unique insight into AI.



YUZHONG PENG (Student Member, IEEE) was born in Nanning, China, in 1980. He is currently pursuing the Ph.D. degree with Fudan University. He is also a Professor and the master's tutor with Fudan University. He was selected as a Young Scholar of the Ministry of Education in China as a Visiting Scholar Program and visited the Laboratory of Prof. Chen Songcan, Nanjing University of Aeronautics and Astronautics, from 2015 to 2016.

He also visited the School and participated in Beijing Chaotou Company Ltd.'s research and development of the large-scale software system MeteogIS of the National Meteorological Administration, from 2010 to 2011. In recent years, he has presided over one national project, five provincial research projects, one prefectural-level research project, participated in four national projects, and three provincial research projects. He has published 27 articles, including the first authors have published 15 articles (one SCI journal article, two EI journal articles, three EI conference papers, one ISTP index, six core journals, and one scientific and technological core journal), and five authors (including Chinese core three journals), two invention patents, and nine software copyrights. His main research interests include intelligent computing, data modeling and mining, and medical bioinformatics. He is a member of the Chinese Computer Society and the American Computer Association (ACM).

• • •