Research Article

Corpus-Based Japanese Reading Teaching Database Cloud Service Model

Xiaoying Liu

College of International Education, Shanghai Jian Qiao University, Shanghai 201306, China

Correspondence should be addressed to Xiaoying Liu; 11034@gench.edu.cn

Received 11 January 2022; Revised 26 January 2022; Accepted 1 February 2022; Published 7 March 2022

Academic Editor: Naeem Jan

Copyright © 2022 Xiaoying Liu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the widespread application of mobile Internet technology, big data, and artificial intelligence, the demand for computing resources in the whole society is growing rapidly. Database cloud service is an online managed high-availability database service built on a cloud computing platform. It can be used as a basic service of the cloud computing platform and has the characteristics of high service availability and high data reliability. The purpose of this paper is to design a corpus-based Japanese reading teaching database cloud service system and integrate a machine learning-based Structure Query Language (SQL) injection detection function for the system. The proposed work is as follows: (1) Design the basic framework of database cloud service. The module of cloud database service is introduced, and the realization logic of main system functions is analyzed. (2) A SQL injection classifier based on feature engineering and machine learning is designed for database cloud services, which can determine whether the input SQL command is an injection statement. The classifier is deployed between the database instance and the front-end server to integrate the SQL injection detection function into the cloud database instance. Then use the support vector machine algorithm to perform simulation experiments on the classifier model to compare the classification performance of the support vector machine algorithm when using different kernel functions. It is found that when the classification decision threshold is 0.5, the classifier using linear kernel function or radial basis kernel function has better classification performance for SQL sentences.

1. Introduction

Cloud service providers are progressively investing in research and development as cloud computing technology continues to advance. Database (DB) is a fundamental component of many applications and thus plays a prominent role in the application system [1]. In recent studies, the well-known information technology (IT) consulting firm Gartner predicted that cloud services will dominate the database business in the future. By 2022, 75 percent of databases will be installed or transferred to the cloud platform, with just 5 percent evaluated for local deployment [2, 3]. This report shows that the research on database cloud services has broad prospects, and the database is the foundation of many applications, which is of great significance to the research on database cloud services. Database cloud service is the trend of future database technology development. After studying a large number of Japanese reading teaching database

materials, this paper proposes corpus-based research on the cloud service model of Japanese reading teaching database. Increasing the database's storage capacity while removing the need to repeatedly configure humans, hardware, and software is made possible by deploying the database in a virtual computing environment [4, 5]. To some degree, traditional databases suit the demands of user data management. However, as a result of its intrinsic flaws and the fast growth of information technology, particularly in the context of the administration and use of vast data on cloud computing platforms, its downsides are becoming more apparent. Database cloud services development has progressively become the development path of a new generation of databases. It tackles the issues of high hardware cost, poor scalability, high management difficulties, sluggish business response, and low resource usage as compared to conventional databases [6, 7]. In the database cloud service application, users can use database resources on demand and

do not need to know the implementation details of the database. The user likes using a database on a single server. The impact of database cloud services has greatly changed the data management mode of enterprise managers. Enterprise users may swiftly construct numerous database-related apps on the web for small- and medium-sized businesses to accomplish full enterprise administration [8]. Data management resources for big businesses may be reduced by using a database cloud service. Data management has seen a shift from on-premises data centers to cloud-based database services [9]. The wide range of possibilities for database cloud services has spurred innovation and progress in the sector. Database as a service (DBaaS) has become more popular due to the rise of cloud service models. Due to the fact that users are freed from having to worry about database setup and administration, conventional IT departments and business departments may operate at higher levels of productivity [10]. In Section 2, the review of literature is given which highlights the related works in the field. Section 3 completes the design of the database cloud service system based on analysis of system requirements. It introduces the implementation and design of all system's submodules and comprehensively analyzes the system functions' implementation process. In Section 4, the performance test of the DB cloud service system is designed and implemented in the previous article and the support vector machine (SVM) two-category test in SQL injection detection. Finally, Section 5 concludes the study.

2. Related Work

Corpus construction is a basic project to provide various knowledge resources for all aspects of language information processing [11]. The establishment of a high-quality, deep-processed corpus is a basic condition for the continuous development of natural language processing technology. However, corpus construction is a huge project. It is necessary to establish and maintain large-scale language data resources, matching unified and standardized management measures and high-end management techniques [12]. Because our country's Japanese reading teaching database is not ideal, it is critical to build a corpus-based Japanese reading teaching database. According to the needs of enterprises and individuals, database cloud services are divided into the following three categories: The first category is Microsoft's relational database cloud service SQL Azure, which can be used to store business data, consumer data, or system data [13, 14]. SQL Azure is the first cloud service model that fully supports relational databases. The bottom layer uses the latest stable version of SQL server database as the engine [15]. SQL Azure has the challenge of how to balance the potential goal of optimizing resource allocation efficiency with a positive user experience and understand when cloud database customers use their database instances and when they are idle, so as to develop idle profiles for customer databases; literature [16] realized the optimal allocation of database resources by predicting the usage of Azure SQL user database resources across data centers. The second category is "SimpleDB" products. Amazon's

"SimpleDB" is a web service that can provide core database functions, such as searching and querying structured data quickly and in real time [17]. "SimpleDB" is considered to be the most valuable data storage solution for building new web applications, because it usually performs many daily tasks related to database management and scalability and eliminates the burden of developers considering data modeling, maintaining indexes and performance adjustments or data operations, and it is also an ideal solution for existing applications [18]. The third category is Oracle's "ExaData" database machine, whose core part is composed of a database server and a storage server and directly provides users with a database machine [19]. Oracle's "ExaData" comes with its own CPU and memory, which realizes the transfer of query processing to the storage system and greatly reduces the amount of data sent to the server. But in terms of technology, because "ExaData" is not a true share-nothing MPP architecture, it does not support unlimited expansion of system resources. When a large number of parallel queries and data loading processes are processed, it must wait for other processing to release resources, resulting in a decrease in parallel processing performance [20]. "ExaData" series versions can provide large enterprises with a complete set of safe and reliable cloud database services through continuous optimization and upgrading of software and hardware configuration and storage services [21]. Domestic research on database cloud services by Alibaba cloud, Tencent cloud, and Baidu cloud has also achieved rapid development and significant results, especially the Alibaba cloud database which is the most prominent. Relational cloud database products and nonrelational cloud database products have been released one after another.

The database cloud service abstracts the underlying relational database into a series of service nodes and gives customized SQL for database access. It is mentioned in [22] that using database management systems (DBMSs) to schedule database cloud resources cannot effectively improve resource utilization and performance. In order to solve the problem of effective allocation of virtual services to physical resources and all component configurations, a resource allocation device is designed in the literature. Multitenant uses multitenant services through application interfaces, and each database service calculates and analyzes cost, workload, and related resource models through resource allocators to improve the performance of cloud database services.

Although the database cloud service technology is developing rapidly, at present, most of the database cloud products launched on the market are based on commercial products in the public cloud environment and the database cloud service system in the private cloud environment for internal use by enterprises or institutions. There are few solutions. This paper studies the current status of database cloud services at home and abroad and related technologies, proposes a solution to build a database cloud service system under a private cloud platform, and studies a corpus-based Japanese reading teaching database cloud service model to meet the needs of related enterprises or institutions for private cloud databases.

3. Method

On the basis of system requirements' analysis, this section completes the design of the database cloud service system. First it introduces the design and implementation of each submodule of the system and then analyzes the implementation process of system functions in detail. The basic functions of this system are realized by three main modules, namely, the back-end management module, the task management module, and the instance agent module. Among them, the instance proxy module is the most complex module in the system.

3.1. Back-End Console Module. The main function of the back-end console module is to process requests received by the database cloud service system from users or other modules within the system and to process business logic. Table 1 defines 14 user interfaces, which accept user requests and complete system functions.

Table 2 defines the 6 most important internal interfaces. Internal interfaces are interfaces that are open to other modules of the system or other systems within the cloud computing platform, and they are used to perform logical interactions between internal modules. The separate design of user interface and internal interface is to prevent illegal external access and reasonably divide the categories of access authority management. When the user interface is accessed, the cloud authentication system verifies the user's identity and key, while the internal interface verifies the identity and key of the internal module that initiated the request. The separate management of the two different identities can clarify the logic of the authentication module of the cloud computing platform, which is convenient for development and maintenance.

As shown in Figure 1, it is the information flow diagram of all virtual machine resources used by the back-end console to manage the system. System tasks can be defined as two types of tasks: synchronous tasks and asynchronous tasks. When the system creates, deletes, upgrades, or automatically repairs an instance, it executes asynchronous tasks.

3.2. Task Management Module. The task management module is an important part of the database cloud service system, and its role is to execute the asynchronous tasks of the system. It receives the information of the task initiated by the back-end console from the message queue and generates the configuration file of the task information for use by the corresponding task execution script, then calls the task script to execute the task, and manages the execution status of the task; finally, it backfills the relevant information of the task to the system database. The task management module realizes the unified management, concurrent scheduling, and status monitoring of the task execution program, so that the task execution program can be separated from the front-end, and the development efficiency is improved. Multiple task management modules can be deployed at the same time to monitor and manage tasks. After the task management

module is started, a task monitoring subprocess is generated and the monitoring port is bound to perform task monitoring. As soon as a task message has been received and processed, a file relating to that task is formed and the task execution subprocess is initiated, which waits for the task to complete. Data information during task execution will generate a data file, and log information will generate a log file; if the task is successfully executed, the data will be backfilled and the log information will be sent to the system database, the task status will be changed to successful execution, and the configuration, data, and log information will be deleted. When the task is executed abnormally, backfill the data, log information to the system database, and change the task status to execution failed.

3.3. Instance Proxy Module. The role of the instance proxy module is to receive requests from other modules, such as the back-end console and task management module, and operate on the instance's active and standby virtual machines and the database engine in the instance. The operations of the user and other modules of the system on the instance all must be implemented through the instance proxy module. The function submodules of the example proxy module are divided into the following. The functions of the instance proxy module can be divided into two categories, one is the function of interacting with the external module of the instance, and the other is the management function for the internal of the instance.

- (1) The functions that interact with the external modules of the instance can be divided into remote request processing functions and monitoring functions. Remote request processing function: In order to solve the problem that other modules in the database cloud service system are inconvenient for instance operation due to separate deployment and to facilitate the unified management of database cloud service instances, it is necessary to implement the Hypertext Transfer Protocol (HTTP) server function for the instance proxy module to provide Representational State Transfer (RESTful) compliance style Application Programming Interface (API) for other modules to call and execute. The remote requests sent by other modules mainly include SQL commands and call commands for custom task scripts. Monitoring function consists of two functions: monitoring data collection and monitoring data push. Monitoring data collection refers to the collection of related operating status information of the virtual machines, engines, and components of the instance main library at regular intervals, and the collected monitoring data is written to a file. Monitoring data push refers to pushing the latest monitoring data to the cloud monitoring system at regular intervals. Users can view the monitoring information of the database cloud service instance on the front-end interface, grasp the running status of the instance, and set alarm policies.

TABLE 1: User interface definition table.

Number	Interface name	Category
1	Instance creation	Instance management
2	Instance list query	Instance management
3	Instance information query	Instance management
4	Instance restart	Instance management
5	Instance delete	Instance management
6	Instance mating	Instance management
7	Management account creation	Account management
8	Manage account list query	Account management
9	Manage account deletion	Account management
10	Backup creation	Data management
11	Backup file list query	Data management
12	Backup and restore	Data management
13	Backup strategy modification	Data management
14	Whitelist modification	Data management

TABLE 2: Internal interface definition table.

Number	Interface name	Function
1	Disk usage push	Push disk usage when the main library accesses the port
2	Virtual machine creation	Apply for virtual machine call when creating an interface
3	Virtual machine deletion	Called when the instance is deleted
4	Backup precheck	Check if backup is allowed
5	Backup information backfill	Push the information to the system database after the backup is completed
6	Cloud host system callback	Push the information to the console after assigning the virtual machine

FIGURE 1: Unified management of virtual machines by the back-end console.

- (2) The functions for instance-oriented internal management can be divided into timed task scheduling functions and log management functions. Timed task scheduling: Realize the function of regularly executing custom tasks, such as regularly executing custom task script update tasks and automatic database backup tasks. This function repeatedly checks the configuration file of the scheduled task and executes the corresponding custom task script regularly according to the scheduling strategy in the configuration. Log management: It includes log

output, log segmentation, and log file cleaning functions. The log output function includes defining the log category, formatting the output content, and defining the path of the output log file. If the log information is written to the same log file, as the instance runs, a single log file will become larger and larger, the I/O operation performance of the log file will be severely reduced, and it will also cause inconvenience to the log transfer. Therefore, the log file needs to be divided regularly, and a single log file is divided into multiple log files named by time. In

order to reduce unnecessary consumption of instance disk space, log files should be uploaded to the cloud storage platform regularly and the old logs on the instance disk should be cleaned up. In summary, the instance agent module is divided into four submodules, which are the remote request processing submodule, the timed task scheduling submodule, the log management submodule, and the monitoring submodule.

3.4. SQL Injection Detection of Database Cloud Service Instance. Data security is one of the important concerns for evaluating the pros and cons of a database service, and it is a factor that must be considered when designing a database cloud service. Database security issues include the control of database access permissions, the security audit of database operations, the fine-grained permission control of database objects, and the detection and filtering of SQL injection attacks. The web application attack statistics report released by the network security organization OWASP in 2017 shows that injection attacks rank first, and SQL injection is the most common type of injection attacks. Therefore, it is of great practical significance to design SQL injection defense functions for database instances. This paper designs a SQL injection detection system integrated into the cloud database instance. At first, it introduces the design and deployment of the detection system, uses the support vector machine algorithm to train the SQL classifier and evaluate the performance of the classifier, and finally optimizes the classification threshold of the classifier.

3.4.1. Introduction to SQL Injection. SQL is the abbreviation of structured query language, which is a language used to manipulate databases. SQL injection refers to the use of HTTP requests to make the web server execute malicious SQL commands. The specific method is to insert a specific SQL command into the HTTP request and then send the HTTP request to the web server to deceive the server and execute malicious SQL commands, thereby stealing, tampering, or destroying data. Web applications can usually be abstracted into three layers, including presentation layer, logic layer, and storage layer. The presentation layer is the front-end client, including browsers on PCs and mobile devices and apps with online service functions on the mobile terminal. The client is used for interaction between web applications and users. The client sends an HTTP request to the web server and receives the HTTP response from the web server, thereby realizing front-end and back-end interaction. The logic layer is located between the storage layer and the presentation layer. It is the execution part of the business logic of the web application, which executes the business and returns the result to the client. The storage layer is dedicated to providing data storage services for web services. For data security, the logic layer and storage layer of modern web services are usually deployed on different servers. The web server uses SQL commands to read and write data in the database to complete its business. If there are security holes in the logic of the web server, after receiving some malicious

HTTP requests from the client, it may generate unexpected SQL commands and send them to the back-end database. If the database cannot determine the security of the SQL command, it may execute the malicious SQL and cause data loss.

3.4.2. The Defense Method of SQL Injection. Traditionally, protection against SQL injection has been carried out in two phases. The first stage is to protect during the writing of the web service source code. Developers follow the empirical guidelines for preventing SQL injection when writing the source code and use methods such as “check parameter format,” “filter special characters,” and “bind parameters” to make SQL injection fails. However, the effectiveness of these methods depends on the comprehensiveness of the security knowledge of the developers, which often leave undetectable loopholes due to human oversight. The second stage is to protect when the web service is running, that is, to deploy an intrusion detection system. There are two traditional SQL injection detection systems: SQL injection detection systems based on program analysis and SQL injection detection systems based on feature matching.

- (1) The SQL injection detection system based on program analysis analyzes the web service program to find vulnerabilities and then by patching vulnerabilities it defends against SQL injection. Program analysis includes static analysis and dynamic analysis. Static analysis is to analyze the source code to find potential security vulnerabilities. Dynamic analysis is done while the program is running, and the program is tested by writing specific test cases to find out security vulnerabilities. The disadvantage of the method based on program analysis is that the source code of the web application must be known, and the analysis coverage cannot be guaranteed, and it is easy to miss the test.
- (2) The SQL injection method based on feature matching includes two methods. The first method is to establish the characteristic syntax tree of the current database service, extract the legal SQL command syntax tree from it, and then perform pattern matching. If the detected SQL command does not conform to the legal pattern, a warning and blocking will be generated. But this method requires the source code to be known in advance. The second method is to analyze a large number of SQL injection instances, generate a matching model of illegal SQL commands, and then perform pattern matching on the actual running SQL statement. If the detected SQL statement conforms to the illegal model, it is judged as SQL injection. This method does not require knowledge of the source code of the application. However, these two matching methods require a lot of manpower and time for analysis and are prone to omissions in detection. The SQL injection detection system designed in this article is a defense

against SQL injection during the running phase of web services. Different from traditional string matching, this detection system extracts the lexical features of the original SQL statement and uses machine learning to train the classification model and then deploys the classification model to the cloud database instance to determine whether the original SQL received by the instance is SQL injection and make corresponding interception or release.

3.4.3. Introduction to Machine Learning Algorithms. There are many algorithms suitable for two classifiers in machine learning algorithms, such as the support vector machine (SVM) algorithm. The following describes the SVM algorithm. SVM uses the strategy of maximizing the interval to find the dividing hyperplane that divides the two classes of samples in the space. For linear inseparable problems, in order to realize the search of dividing the hyperplane, it is necessary to use the kernel function to map the samples into the high-dimensional space. Following kernel functions are commonly used:

$$C(x, x') = x^T x'. \quad (1)$$

$$C(x, x') = (x^T x')^b. \quad (2)$$

$$C(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right). \quad (3)$$

$$C(x, x') = \tanh(\lambda x^T x' + \mu). \quad (4)$$

Formula (1) gives the linear kernel function, formula (2) presents the polynomial kernel function, formula (3) expresses the radial basis kernel function, and formula (4) is the Sigmoid kernel function. Different kernel functions are suitable for different classification samples, and the classification performance of different kernel functions needs to be evaluated in the test.

Decision tree is a learning method for classification and fitting. This algorithm requires less data and is easy to understand and explain. Decision trees generally use information gain or Gini index as an index to measure the purity of a sample set. Assuming that the proportion of the i -th sample in the sample set S is w_i , ($i = 1, 2, \dots, n$), the information entropy is

$$I_{\text{ent}}(S) = -\sum_{i=1}^n w_i \log_2 w_i. \quad (5)$$

Use the feature f of the sample to divide the sample, and the information gain obtained by the division is defined as

$$I_{\text{gain}}(S, f) = I_{\text{ent}}(S) - \sum_{m=1}^M \frac{|S^m|}{|S|} I_{\text{ent}}(S^m). \quad (6)$$

The greater the information gain, the greater the increase in purity obtained by using the attribute a to divide the sample, and the better the division effect. Decision trees are prone to overfitting, which can be solved by training multiple decision trees in an integrated learning method.

3.4.4. Performance Evaluation Criteria of the Classifier. To judge the performance of the classifier, comparison methods and performance indicators quoted from medical tests are often used. The method is to calculate the confusion matrix based on the classification results and the real labels. The confusion matrix consists of four parts: True positive (TP), false negative (FN), false positive (FP), and true negative (TN). The number of true positive events (PE) is the sum of TP and FN; the number of true negative events (NE) is the sum of FP and TN; the number of positive observations (PO) is the sum of TP and FP, and the number of negative observations (NO) is the sum of FN and TN; the total number of events (TE) is the sum of TP, FN, FP, and TN. Performance indicators include true positive rate, false positive rate, accuracy, precision rate, recall rate, F1 score, false positive rate and missed positive rate, and the area under the ROC curve. The meaning of each performance index is introduced below.

- (1) True positive rate (TPR): It means the ratio of the number of true positive samples to the number of true positive samples, reflecting the accuracy of the classifier's detection of positive samples. The formula is

$$\text{TPR} = \frac{\text{TP}}{\text{PE}} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \quad (7)$$

- (2) False positive rate (FPR): the meaning is the proportion of the number of false positive samples to the number of true negative samples. The higher the false positive rate, the higher the false positive rate of the classifier. The formula is

$$\text{FPR} = \frac{\text{FP}}{\text{NE}} = \frac{\text{FP}}{\text{FP} + \text{TN}}. \quad (8)$$

- (3) Accuracy (Acc): characterize the overall detection accuracy of positive and negative events. The formula is

$$\text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TE}} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}}. \quad (9)$$

- (4) Precision (Pre): It reflects the accuracy of detection of positive events, and its complement is the rate of misjudgment. The formula is

$$\text{Pre} = \frac{\text{TP}}{\text{PO}} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \quad (10)$$

- (5) Recall rate (Rec): The value is consistent with the TPR, which reflects the comprehensiveness of the detection of positive events, and its complement is the missed detection rate. The formula is

$$\text{Rec} = \frac{TP}{PE} = \frac{TP}{TP + FN} = \text{TPR}. \quad (11)$$

- (6) False Discovery Rate (FDR): The formula is the ratio of the number of times that the classifier judges normal SQL as SQL injection to the total number of SQL injection observations. The complement is the precision rate. The formula is

$$\text{FDR} = \frac{FP}{PO} = \frac{FP}{TP + FP} = 1 - \text{Pre}. \quad (12)$$

- (7) Missing Discovery Rate (MDR): It represents the ratio of the number of times that the classifier judges SQL injections as normal SQL to the total number of real SQL injections. The complement is the recall rate. The formula is

$$\text{MDR} = \frac{FN}{PE} = \frac{FN}{TP + FN} = 1 - \text{Rec}. \quad (13)$$

- (8) F measurement ($F1S$): the harmonic average of precision and recall, which can comprehensively evaluate precision and recall. The formula is

$$F1S = 2 \cdot \frac{\text{Rec} \cdot \text{Pre}}{\text{Rec} + \text{Pre}}. \quad (14)$$

4. Experiment and Analysis

The main content of this chapter is the performance test of the database cloud service system designed and implemented in the previous article and the SVM two-category test in SQL injection detection.

4.1. Performance Test. The performance of the database cloud service system and the performance of the mysql database instance are tested below. The performance of the cloud database service system is mainly tested in terms of response time and the number of concurrent users. The performance of the mysql database instance is mainly tested in terms of the transaction volume TPS processed by the instance per second, the query volume per second QPS, CPU computing performance, disk IO performance, etc. Siege is an open-source HTTP/FTP load test and benchmark stress test tool. It can simulate N users to visit a certain URL in a cycle of R times and return the test result in JSON format. The important parameter is elapsed time, which means the test time. In this paper, Siege is used to simulate 10,000 users to access the system simultaneously once in a loop to obtain a graph of system response time changes. The system performance test response time is shown in Figure 2. It can be seen that the system performance is better when the number of concurrent users of the system is within 5000.

Sysbench is an open-source multithreaded performance test tool that supports various tests. The main purpose is to perform OLTP benchmark tests on the mysql database to test the load situation of the database under different parameters. In addition, sysbench also supports CPU computing performance testing, disk IO read and write

FIGURE 2: The system response time changes with the amount of concurrency.

performance testing, etc. Next, it uses sysbench for stress test mysql. The sysbench stress test is divided into three phases, the data prepare phase, the run phase, and the cleanup phase. The data prepare phase refers to the creation of test data, the run phase is to execute test commands, and the cleanup phase refers to the removal of invalid data generated by the test. In this test, the execution pressure measurement script is "oltp_read_write.lua", which means that the "Lua" script for reading and writing is about 7 points for reading and 3 points for writing. 3 tables are created, each with 5,000,00 data items, and finally connected to the database. After the data preparation is completed, use the run command to perform stress testing. The number of threads created for the stress test is 64, the stress test is executed for 60 s, and the reporting interval is 6 s. This test selects transactions per second (TPS) as an indicator to measure the performance of the mysql cloud database instance. Comparing the performance of the database cloud service system instance TPS with that of the local database instance TPS, the comparative result presented in Figure 3 shows that the database cloud service system instance performs better.

Comparing the performance of the database cloud service system instance TPS with that of the local database instance TPS, the comparative result presented in Figure 3 shows that the database cloud service system instance performs better. The performance of the cloud database instance has reached a good level and can meet the performance requirements of users for cloud database instances.

4.2. Support Vector Machine Two-Classification Test. Use the training data set to train SVM two classifiers and input the test data set to the classifier for testing. For each input, the classifier outputs a probability within the range of $[0, 1]$. Select the default classification threshold (the default is 0.5); when the score is greater than the threshold, it will be judged as 1, which means negative (normal SQL); when the score is less than the threshold, it will be judged as 0, which means it

FIGURE 3: Comparison of cloud service database and local database instance TPS.

FIGURE 4: FPR comparison between linear kernel and polynomial kernel function.

FIGURE 5: FPR comparison between radial basis kernel and Sigmoid kernel function.

is positive (SQL injection). Finally, the classification results are compared with the real labels in the test training set, and the ROC curves of the SVM two classifiers using different kernel functions are drawn, and Figures 4 and 5 are obtained. In the figure, the x -axis direction is the FPR; the larger the value, the worse the classification performance; the y -axis direction is the TPR; the larger the value, the better the classification performance. From the ROC graph, it can be judged that the SVM two classifiers using the Sigmoid kernel function have a poor classification effect on SQL statements, and the other three kernel functions have a better classification effect.

5. Conclusion

The database industry will be dominated by cloud computing in the future, and more databases will be placed on or transferred to cloud platforms than there already are. Because of technical advancements and budgetary concerns, this is unavoidable. Because of the lower costs, small- and medium-sized businesses may choose public cloud databases. A private cloud database will be built for major organizations and institutions who are concerned about data protection. Cloud database service system solutions are few in the present private cloud environment, and the Japanese corpus-based teaching database for reading in Japan is much more difficult to locate. Based on the laboratory private cloud platform, this paper proposes the construction of database cloud services under the private cloud platform. It designed and implemented a database cloud service system. The proposed work is as follows: (1) Research the technical background of database cloud service, analyze the key characteristics of the database cloud service system, and clarify the requirements of system. (2) The design and implementation of the database cloud service system divided the components of the database cloud service system and designed the task execution mode. Tests show that the system can provide highly available database cloud services.

Data Availability

The datasets used during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest

The author declares that there are no conflicts of interest.

References

- [1] M. Seibold and A. Kemper, "Database as a service," *Datenbank-Spektrum*, vol. 30, no. 2, pp. 1216-1217, 2012.
- [2] N. Leavitt, "Is cloud computing really ready for prime time?" *Computer*, vol. 42, no. 1, pp. 15-20, 2009.
- [3] P. Xiong, Y. Chi, S. Zhu, H. J. Moon, C. Pu, and H. Hacgumus, "SmartSLA: cost-sensitive management of virtualized resources for CPU-bound database services," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 5, pp. 1441-1451, 2015.
- [4] K. Channabasavaiah, K. Holley, and E. Tuggle, "Migrating to a service-oriented architecture," *IBM Developer Works*, vol. 16, no. 5, pp. 1-21, 2003.
- [5] R. Alaa, "Design of SOA-based grid computing with enterprise service," *Bus. AISS: Advances in Information Sciences and Service Sciences*, vol. 2, no. 1, pp. 71-82, 2012.
- [6] J. L. Mar E Chaux, "Combining service-oriented architecture and event-driven architecture using an enterprise service bus," *IBM Developer Works*, vol. 13, no. 2, pp. 1269-1275, 2006.
- [7] J. Weis and J. Alves-Foss, "Securing database as a service: issues and compromises," *IEEE Security and Privacy Magazine*, vol. 9, no. 6, pp. 49-55, 2011.
- [8] A. Brown, S. Johnston, and K. Kelly, "Using service-oriented architecture and component-based development to build web service applications," *Rational Software Corporation*, vol. 10, no. 4, pp. 3-15, 2002.
- [9] A. Khan, A. Zugenmaier, D. Jurca, and W. Kellerer, "Network virtualization: a hypervisor for the Internet?" *IEEE Communications Magazine*, vol. 50, no. 1, pp. 136-143, 2012.
- [10] R. Jia, Y. Yang, J. Grundy, J. Keung, and Li Hao, "A systematic review of scheduling approaches on multi-tenancy cloud platforms[J]," *Information and Software Technology*, vol. 132, no. 1, Article ID 106478, 2020.
- [11] S. Sharoff, "Web corpus construction[J]," *Catalan Journal of Communication & Cultural Studies*, vol. 6, no. 1, pp. 19-33, 2015.
- [12] N. Tongtep and T. Theeramunkong, "Multi-stage automatic NE and PoS annotation using pattern-based and statistical-based techniques for Thai corpus construction," *IEICE - Transactions on Info and Systems*, vol. E96.D, no. 10, pp. 2245-2256, 2013.
- [13] K. Kline, "Azure SQL database edge[J]," *Database Trends and Applications*, vol. 33, no. 4, pp. 324-327, 2019.
- [14] S. Acheciński, "Obsługa danych temporalnych Na platformie MS SQL server azure SQL database support of temporal data on the MS SQL server and azure SQL database platform[J]," *Przegląd Elektrotechniczny*, vol. 12, no. 96, pp. 95-101, 2020.
- [15] L. J. Tu, "Six reasons to choose cloud database SQL azure[J]," *Network & Information*, vol. 26, no. 06, p. 37, 2012.
- [16] D. Dundjerski, S. Lazi, M. Tomaevi, and D. Bojić, "An extended evaluation of schema issues advisor in the azure SQL database[J]," *Telfor Journal*, vol. 10, no. 2, pp. 91-96, 2018.
- [17] M. S. Ramanathan, M. S. Goel, and M. S. Alagumalai, "Comparison of cloud database: amazon's SimpleDB and google's bigtable, recent trends[J]," *International Journal of Computer Science Issues*, vol. 8, no. 6, pp. 243-246, 2011.
- [18] D. Ash, C. Brown, and B. Kluger Bell, "Creating hybrid communities using inquiry as professional development for college science faculty [J]," *Journal of College Science Teaching*, vol. 38, no. 6, pp. 68-76, 2009.
- [19] M. Fotache, A. Tică, I. Hrubaru, and T. M. Spînu, "Big data proprietary platforms. The case of Oracle e," *Review of Economic and Business Studies*, vol. 11, no. 1, pp. 45-78, 2018.
- [20] T. Yasuike and T. Konishi, "New world of database appliance: the technical advantages of Oracle exadata V2[J]," *Ieice Technical Report Data Engineering*, vol. 109, pp. 19-24, 2009.
- [21] Q. Liu, "Oracle exadata: integration of IT hardware and software [J]," *Zhongguo Xinxihua*, vol. 5, p. 1, 2010.
- [22] T. Kiefer and W. Lehner, "Private table database virtualization for DBaaS," vol. 12, pp. 328-329, in *Proceedings of the The 2011 IEEE 4th International Conference on Utility and Cloud Computing (UCC 2011)*, vol. 12, IEEE, Los Alamitos, CA, USA, 5 December 2011.