WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

E- Commerce Application Service onto the Public Cloud Server.

Nimra Fatima[1],Tahera Abid[2]

[2 –] Assistant Professor & Co-ordinator of IT Department,[B.Tech,M.Tech)]

[1 –] Department of IT, Nawab Shah Alam Khan College of Engineering & Technology, Affiliated to Osmania University, Hyderabad, India.

A B S T R A C T :

The rising popularity of online shopping has led to a steady stream of new product evaluations. Consumers benefit from these evaluations as they make purchasing decisions. Many research projects rank products using these reviews, however, most of these methodologies have ignored negative polarity while evaluating products for client needs. The main contribution of this research is the inclusion of negative polarity in the analysis of product rankings alongside positive polarity. To account for reviews that contain many sentiments and different elements, the suggested method first breaks them down into sentences. This process aids in determining the polarity of products at the phrase level by extracting elements from product evaluations. The next step is to link the polarity to the review's sentence-level features. Products are prioritized following user needs by assigning relative importance to each of the polarities. The Amazon review dataset has been used in the experimental assessments so that the efficacy of the suggested approach can be estimated. Experimental evaluation of PRUS utilizes rank score (RS) and Reviews. Results indicate that PRUS gives independence to the user to select recommended list based on specific features with respect to positive or negative aspects of the products.

Keywords: E-Commerce, Cloud Computing, Public Cloud Server, Scalability, Infrastructure as a Service, Cloud Security.

Introduction:

Due to the growth in popularity of online shopping, a large number of opinions or reviews are being generated on a regular basis. These reviews are not only providing helpful insights for the companies which are planning to launch a new product but can also provide significant information for the user who intends to buy a product. People often. consider buying a product with different specifications as per their needs and desire. Several similar products are offered by different brands and companies which makes it difficult for users to find the best products according to their specifications. Customer reviews can be very useful for decision-making authorities at all tiers that might range from a decision regarding buying a specific product to stock price prediction in stock exchange markets. However, it is hard to extract relevant information from the huge number of reviews. This is because these reviews may contain different opinions and sentiments regarding different features of the same product. There exist several studies which have focused on summarizing user reviews to make it easy to have an idea of the user opinions. These methods include the extractive approach which selects the most important sentence from the input document, the abstractive method generates the summary, hybrid method which combines both the extractive and abstractive methods. However, most of these works do not consider extracting user opinions concerning their sentiments. Customer reviews consist of their opinions about a specific product which can represent varying sentiments about different features of that product. For example, consider a review ‘‘Phone has a great battery life and higher solution camera but the screen resolution is a little low, though it was acceptable’’. This review has overall positive sentiment but when you go deeper it is found that the review describes the screen resolution in a negative sense. The sentiment of a review plays an important role in decision making; however, aspect-level sentiment is more important than the overall sentiment of the review. For instance, the above review expresses user opinion about the camera, battery, and screen resolution but gives negative feedback for the screen resolution feature of the phone.

Nomenclature

PRUS	Product Recommender Using System sentiments.
NLP	Natural Learning Processing is a field that focuses on enabling computers to understand, interpret and generate human Language.
RS (Rank Score)	Used for evaluating products based on review polarity.

UI (User Interface)	A user interacts with a device or software application.
JSP (Java Server Pages)	Used for creating dynamic web pages.
IDE	A software application that provides a comprehensive set of tools .
TF-IDF	Term Frequency-Inverse Document Frequency that is used in semantic analysis
Accuracy	How often the system correctly identifies whether an app is safe or harmful.
Precision	Out of all apps marked as harmful, how many where actually harmful.
Recall	Out of all harmful apps, how many the system was able to identify.
Doc2Vec	Document to vector model used for text similarity.
UML	Unified Modelling Language includes Use case, Class, Activity diagrams etc., .
GRU (Grated Recurrent Unit)	Used in deep learning models.

1.1. Table: Resource Utilization Under Load Testing

The table below provides a concise summary and comparative analysis of various models and methods related to sentiment-based recommendation systems and e-commerce cloud deployment. The performance of the proposed system is evaluated alongside other prominent studies using key metrics such as Test case, Concurrent users, CPU Utilizations, Paging rate, Disk Traffic and feature methodologies. This comparative analysis demonstrates the effectiveness of the proposed model in handling user reviews, providing personalized recommendations, and leveraging cloud infrastructure for enhanced scalability.

.Table 1: Performance Comparison

Test Case	Concurrent Users	CPU Utilization(%)	Paging Rate	Disk Traffic
A	1000	33.8%	10.054%	2.9%
B	1200	38.1%	15.713%	0.875%

- At 1000 users, the system remains stable, with moderate CPU and disk usage.
- At 1200 users, CPU usage increases significantly while disk traffic decreases, likely due to more optimized resource handling in the clustered environment.
- Paging rate and minimum CPU use drop at higher load, pointing to better memory management and resource optimization.

System Analysis and Design

2.1 Existing System:

Many research projects rank products using these reviews, however, most of these methodologies have ignored negative polarity while evaluating products for client needs Due to the growth in popularity of online shopping, a large number of opinions or reviews are being generated on a regular basis.

These methods have several **problems:**

Data Breaches: Risk of sensitive customer data being compromised due to inadequate security measures.

- Unauthorized Access: Potential for unauthorized access to cloud resources and data.
- Limited Control: Limited control over infrastructure and resources, which can impact performance and scalability.
- Downtime: Risk of downtime due to cloud provider outages or maintenance.
- Cost Overruns: Potential for unexpected cost overruns due to resource usage and scaling.
- Scalability Limitations: Limitations in scaling resources to meet sudden spikes in demand.

- Data Sovereignty: Concerns about data storage and processing locations, which can impact compliance and governance.

2.2 Proposed System:

We propose This process aids in determining the polarity of products at the phrase level by extracting elements from product evaluations. The next step is to link the polarity to the review's sentence-level features. Products are prioritized following user needs by assigning relative importance to each of the polarities. The Amazon review dataset has been used in the experimental assessments so that the efficacy of the suggested approach can be estimated. Experimental evaluation of PRUS utilizes rank score (RS) and Reviews. Results indicate that PRUS gives independence to the user to select recommended list based on specific features with respect to positive or negative aspects of the products.

The proposed system aims to develop and deploy a scalable and intelligent E-Commerce Application Service using Public Cloud Infrastructure. It leverages cloud computing resources to enhance performance, reliability, and cost-effectiveness for hosting and managing an online retail platform.

The system includes a suite of services that enable online businesess to operate and scale efficiently.

- Deploys the e-commerce application on public cloud servers (e.g., AWS, Azure, Google Cloud).
- Utilizes Infrastructure-as-a-Service (IaaS) for dynamic scaling and high availability.
- Integrates sentiment analysis using NLP on customer reviews.
- Ranks and recommends products based on positive/negative feedback and aspect-level opinions.
- Divides the application into modules like product catalog, user management, order processing, and payments.
- Enables independent development, deployment, and scaling of each service.
- Uses encryption and cloud-native security features to ensure secure communication and payment processing.
- Implements role-based access control (RBAC) and secure API endpoints.
- Provides a responsive and intuitive user interface (UI) for customers and admins.
- Real-time search, filtering, and personalization.

2.3 Architecture:

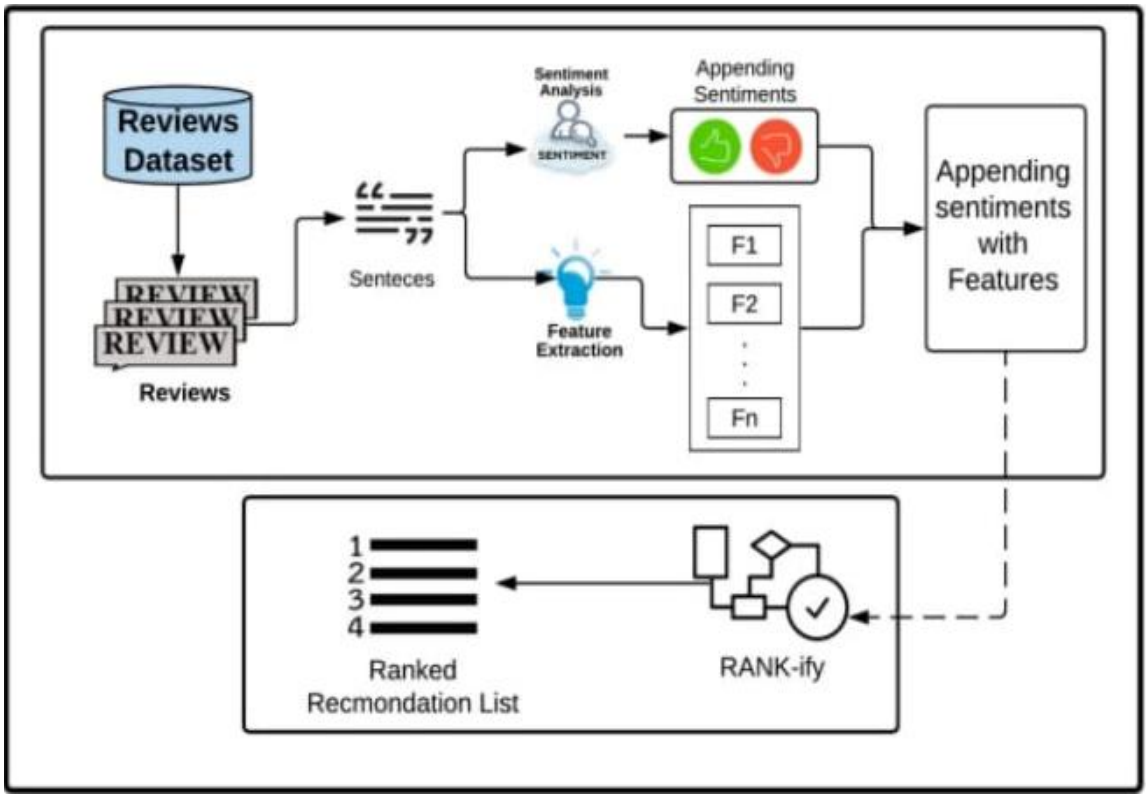

Fig. 1 – System Architecture

The system architecture for an e-commerce application service on a public cloud server is designed to ensure scalability, flexibility, and high availability. A microservices-based architecture is employed, where the application is broken down into smaller, independent services that communicate with each other through APIs. Each microservice is containerized using Docker and orchestrated using Kubernetes, allowing for efficient deployment, scaling, and management.

The application is hosted on a public cloud provider, such as AWS, Azure, or GCP, which provides a range of services to support the application. Serverless computing services, like AWS Lambda or Azure Functions, are used to optimize resource utilization and reduce costs. Managed database services, such as Amazon Aurora or Azure Cosmos DB, ensure high availability and scalability for data storage and retrieval.

To ensure security and compliance, robust security measures are implemented, including encryption, firewalls, and access controls. The application is designed to comply with relevant regulations, such as PCI-DSS, GDPR, and HIPAA.

Methodology:

A cloud-based e-commerce methodology encompasses several key aspects: establishing a cloud infrastructure, developing core e-commerce functionalities, integrating with cloud services, ensuring security and compliance, and managing data and operations
The proposed methodology for the E-Commerce on public cloud system includes the following steps:

- Agile Development: Use agile development methodologies to enable iterative and incremental development.
- Cloud-Native Design: Design the application with cloud-native principles, such as scalability, flexibility, and resilience.
- Microservices Architecture: Use a microservices architecture to enable independent development, deployment, and scaling of services.
- DevOps: Implement DevOps practices, such as continuous integration and continuous deployment (CI/CD), to automate testing and deployment
- Security and Compliance: Ensure security and compliance by implementing robust security measures and adhering to relevant regulations.
- Monitoring and Logging: Implement monitoring and logging mechanisms to track performance, detect issues, and debug problems.
- Testing and Quality Assurance: Perform thorough testing and quality assurance to ensure the application meets requirements and is free of defects.

Results:

CPU Utilization vs Concurrent Users :

- As concurrent users increased from 1000 to 1200:
- Max CPU usage rose from 33.8% to 38.1%.
- Average CPU usage significantly increased, showing heavier load.
- Minimum CPU usage dropped at 1200 users, showing potential load balancing variability.

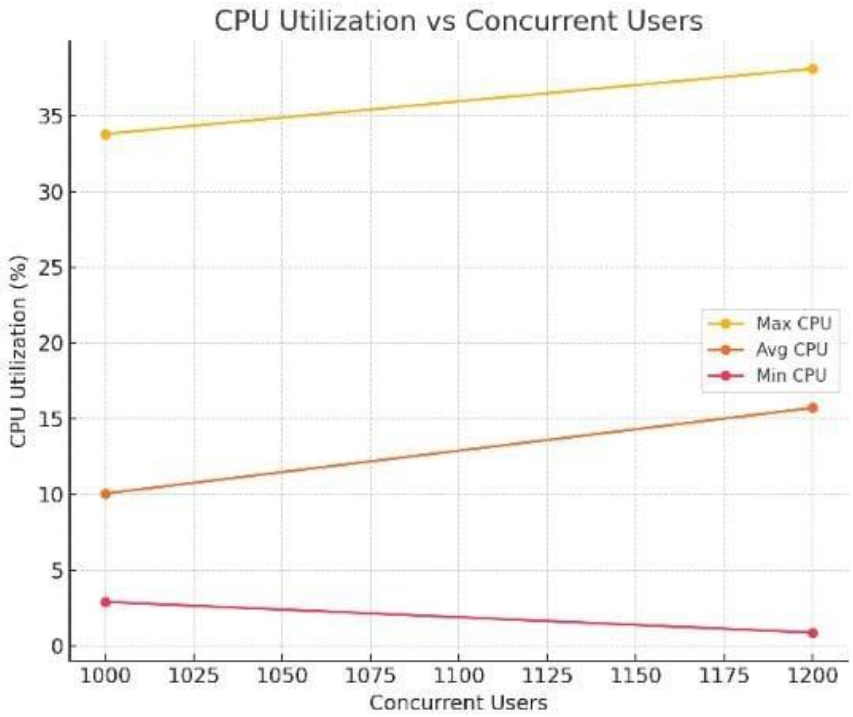

Fig. 2 - CPU Utilization vs Concurrent Users

Disk Traffic vs Concurrent Users:

- Interestingly, disk traffic decreased as users increased.
- Possibly due to efficient clustering or coaching mechanisms under heavier load.

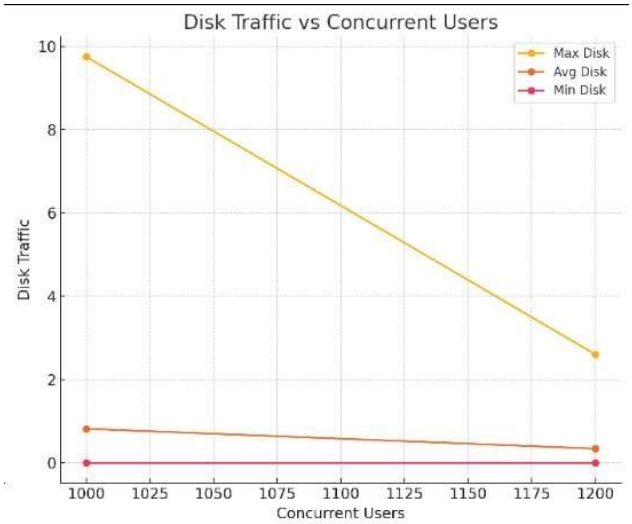

Fig.- 3 Disk Traffic vs Concurrent Users

Conclusion:

In this paper, This study proposed a framework, called PRUS, to generate recommended ranked list of products from customer reviews Existing literature in this domain has not considered negative aspects while recommending ranked products list. Moreover, existing approaches generally rank and display the products to all the users in a similar manner irrespective of the actual needs of the individual user. The proposed PRUS approach worked in three phases. In the first phase, features and sentiments from each sentence of a review have been extracted which are used to calculate the positive and negative sentiments of the products on the sentence level. In the second phase, the extracted features are appended with obtained sentiments, and in the last phase, features from the user query are extracted which are used to rank products considering the calculated Rank Score based on the positive and negative sentiments of their individual features. The positive and negative sentiments of each feature are attributed with a weighted parameter that can be used to control the impact of positive and negative sentiments on the recommended ranked list.

Future Enhancements:

In the future, The proposed PRUS approach is evaluated by conducting comprehensive experiments. This research opens several new opportunities in the product ranking domain. A future direction in this domain is to explore product ranking on multilingual product review datasets. Another interesting future direction is to provide more flexibility to the user to generate recommended ranked product lists based on negative and positive polarities of user-specified features.

REFERENCES

[1] X. Chen, F. Wang, Z. Huang, and Y.-C. Chen, ‘‘Smart grid aggregation billing scheme based on blockchain,’’ J. Netw. Intell., vol. 7, no. 4, pp. 878–893, Nov. 2022. 51628 VOLUME 11, 2023 T.-T. Tsai et al.: Leakage-Resilient AMR-CB-KE Scheme.

[2] Z. Wang, C. Zhang, and X. Mu, ‘‘Decentralized solution for cold chain logistics combining IoT and blockchain technology,’’ J. Netw. Intell., vol. 8, no. 1, pp. 47–61, Feb. 2023.

[3] J. Chen, H. Xiao, M. Hu, and C.-M. Chen, ‘‘A blockchain-based signature exchange protocol for metaverse,’’ Future Gener. Comput. Syst., vol. 142, pp. 237–247, May 2023.

[4] C.-M. Chen, S. Liu, X. Li, S. H. Islam, and A. K. Das, ‘‘A provablysecure authenticated key agreement protocol for remote patient monitoring IoMT,’’ J. Syst. Archit., vol. 136, Mar. 2023, Art. no. 102831.

[5] S. Khasim and S. S. Basha, ‘‘An improved fast and secure CAMEL based authenticated key in smart health care system,’’ Cloud Comput. Data Sci., vol. 3, no. 2, pp. 77–91, May 2022.

[6] T. Saito, K. Xagawa, and T. Yamakawa, ‘‘Tightly-secure key-encapsulation mechanism in the quantum random Oracle model,’’ in Advances in Cryptology—EUROCRYPT (Lecture Notes in Computer Science), vol. 10822. Berlin, Germany: Springer, 2018, pp. 520–551.