ISSN: 2182-2069 / E-ISSN: 2182-2077

Strengthening Cloud Security: AI-Integrated Bug Identification for Financial Applications

Bhagyashri Mankar[1*], Dr. Shoaib Mohammed[2], Rahul Mishra[3], Rakhi Chakraborty[4], Solomon Jebaraj[5], and Dr. Ritesh Rastogi[6]

[1*]Assistant Professor, Department of Information Technology, Yeshwantrao Chavan College of Engineering, Nagpur, India. bhagyashrimankar55@gmail.com, https://orcid.org/0000-0002-2328-9279

[2]Assistant Professor, Department of ISME, ATLAS SkillTech University, Mumbai, Maharashtra, India. shoaib.mohammed@atlasuniversity.edu.in, https://orcid.org/0009-0003-7819-1484

[3]Centre of Research Impact and Outcome, Chitkara University, Rajpura, Punjab, India. rahul.mishra.orp@chitkara.edu.in, https://orcid.org/0000-0001-7042-3058

[4]Assistant Professor, Department of Computer Science & IT, ARKA JAIN University, Jamshedpur, Jharkhand, India. rakhi.c@arkajainuniversity.ac.in, https://orcid.org/0000-0003-4012-3497

[5]Assistant Professor, Department of Computer Science and Information Technology, Jain (Deemed to be University), Bangalore, Karnataka, India. solomon.j@jainuniversity.ac.in, https://orcid.org/0000-0002-3385-207X

[6]Professor, Department of Computer Science, Noida Institute of Engineering & Technology, Greater Noida, Uttar Pradesh, India. ritesh.rastogi@niet.co.in, https://orcid.org/0000-0003-0427-5290

Received: January 20, 2025; Revised: March 01, 2025; Accepted: April 15, 2025; Published: May 30, 2025

Abstract

The process of bug identification is complicated and time-consuming. To effectively reconstruct, localize and solve issues need to select the appropriate tools, techniques and methods. To solve this issue, we propose the cloud security in the financial application by developing an innovative bug identifying method with the use of artificial intelligence (AI) called Shuffled Frog Leaping-driven Dynamic One-class Support Vector Machine (SFL-DOSVM). We gather a dataset and pre-process the obtained raw data by utilizing the text pre-processing techniques that eliminates the redundant data and improves the data quality. Term Frequency-Inverse Document Frequency (TF-IDF) method is employed to extract the significant features. Accuracy, recall, MCC, f1-measure and precision are the metrics used in the bug identification approaches. Performance evaluation carried out by the comparison of proposed and traditional techniques such as Long Short-Term Memory (LSTM), K-Nearest Neighbors (KNN) and Support Vector Machine (SVM). The efficient outcomes are demonstrated by our proposed method than other existing techniques.

Journal of Internet Services and Information Security (JISIS), volume: 15, number: 2 (May), pp. 136-145.
DOI: *10.58346/JISIS.2025.I2.010*

*Corresponding author: Assistant Professor, Department of Information Technology, Yeshwantrao Chavan College of Engineering, Nagpur, India.

Keywords: Financial Application, Artificial Intelligence (AI), Cloud Security, Bug Identification, Shuffled Frog Leaping-driven Dynamic One-class Support Vector Machine (SFL-DOSVM).

1 Introduction

Enhancing bug identification, resolving efficiency and accuracy is the driving force of artificial intelligence (AI) learning-based categorization of bugs in cloud-based computing applications. System administrators' and developers' applications can be decreased by employing AI methods for bug classification (Tabassum et al., 2023). Increased cyber-attacks on cloud infrastructure caused significant data breaches that have cost billions of dollars and revealed millions of secret documents (Torkura et al., 2021). Fintech solutions can utilize the advantages offered by the cloud computing domain. Large data processing capacities are provided by cloud systems and improved security and availability (Stojanović & Božić, 2022). Software developers manage concerns using bug-tracking systems. Thousands of reports of bugs are generated every week by massive systems including Linux, Chromium and the cloud due to the increased frequency of software system changes (Wu et al., 2021). Software has become an essential component of every human's daily existence for its ease of use and rapidity. Several industries utilize embedded software, including manufacturing, commerce and transportation, health and finance (Hai et al., 2022). The use of intermediates like banks or accountants is eliminated by blockchain-based smart contracts (SCs) that manage complicated financial transactions (Sharma et al., 2025). These represent an important development in the field of blockchain that can change the method financial transactions operate (Krichen, 2023). An era of virtual software and data management, dynamic resource deployment and easy exporting of information technology (IT) services was introduced by cloud computing (Sharma & Pandey, 2024). It has greatly improved a business's operational efficiency, cost-effectiveness and energy efficiency (Shreyas, 2023). By utilizing technological advancements, technology for finance or fintech has altered traditional financial circumstances (Huy, 2018). The finance industry's growing dependence on cloud computing has created innovative possibilities for enhancing security protocols (Gajmal & Udayakumar, 2022). By using the AI-based SFL-DOSVM method for bug investigation to protect cloud infrastructure against possible vulnerabilities and attacks, the proposed method can enhance the security of financial applications (Pragadeswaran et al., 2024).

The following section provides an organization of the paper: A literature review is presented in part 2. In part 3, the proposed SFL-DOSVM method is demonstrated. Part 4 determines the performance evaluation and discussion. Conclusions are provided in part 5.

2 Literature Review

Aoudni et al., (2022) introduced Hidden Markov Model Transductive Deep Learning (HMM_TDL), the deep-learning (DL) framework for identifying and preventing cloud platform threats to manage issues associated with zero-day attacks. To prevent attacks on cloud systems, the estimated HMM_TDL data was modified with the estimated security level. Vinoth et al., (2022) employed an examination of research to examine and evaluate the most significant risks for network security and data security on cloud-based platforms. Alwaheidi & Islam, (2022) explored the security concerns around various cloud-based applications in e-commerce and finance. For cloud-based systems, the article's primary contribution was the unique data-driven threat analysis (d-TM) technique. As an outcome, four of the total seven threats that d-TM detected were deemed critical based on the assets that were identified. Targeted data in operation and transit more precisely, the risks focused on business and management data widely. Chen et al., (2023) provided Tyr, the free application that detected Consensus Failure Bugs

(CFBs) in blockchain-based systems that exhibited a high frequency of abnormal diverging consensus behaviours (Wu & Margarita, 2024). Tyr found 20 significant vulnerabilities that were previously undetected and the appropriate maintainers have fixed each one. Kunduru, (2023) considered cloud computing as flexible and inexpensive and has grown into one of the most significant developments in technology. The integration of the cloud into an established business framework offered serious safety hazards and significant implications (Madani & Taha, 2024). Bhargava et al., (2022) employed the recommended technique to replicate the Distributed Denial of Service (DDoS) threat while calibrating and designing a DDoS sensing mechanism. They had selected other machine learning methods related to created results for the objective of establishing the methodology with evaluation. Zhang, (2023) suggested a dynamic and safe solution for storing financial information for businesses based on cloud platforms. The financial database's data was used for data mining and classification based on the absolute correlation degree computation that can be estimated utilizing the grey correlation evaluation technique. The used technique can encrypt all information with an average reliability of the storage outcomes based on experimental findings. Awadallah et al., (2021) promoted the integrity of data with all homomorphic encryption algorithms. The research suggested a blockchain-based cloud computing approach. They determined that Ethereum exceeded Bitcoin as far as online performance and reduced customer financial expenditures.

3 Materials and Methods

The following section employs AI-integrated techniques for identifying bugs in financial applications by employing an innovative SFL-DOSVM technique. The block diagram illustrates the flow of the proposed method that is indicated by Figure 1.

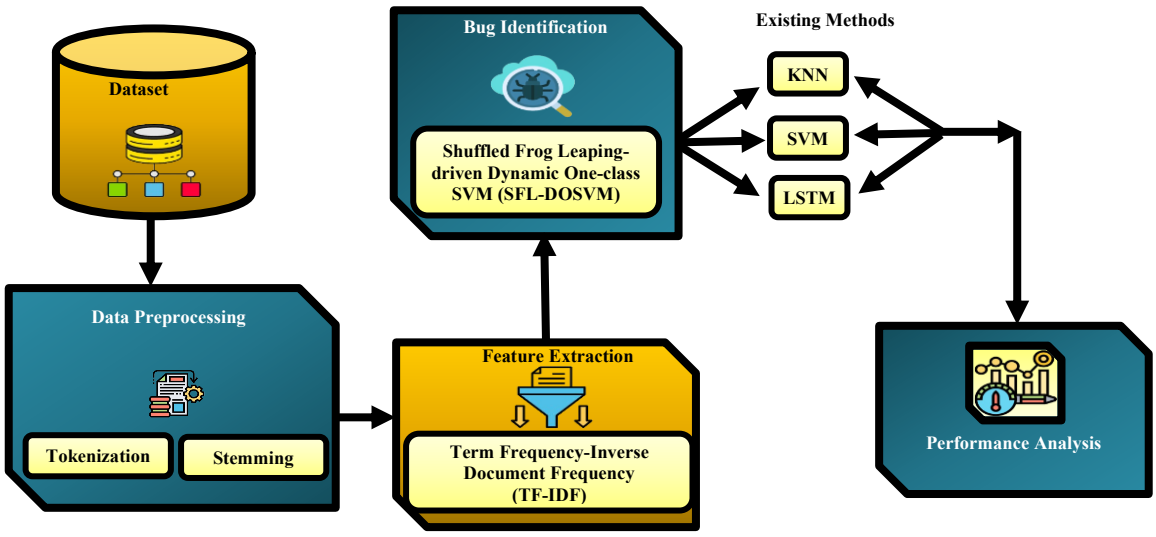

Figure 1: Flow of Proposed Methodology

3.1 Dataset

Data were gathered 54,755 issue tickets in total, all of that were closed tickets associated to processes that included any combination of the following words: financial error, defect in calculations, system breakdown or regression in functionality to generate an appropriate dataset for financial applications (Hirsch & Hofer, 2022). The final dataset showed a massive amount of noise caused by various financial approaches, individually categorized issues by finance developers and abnormalities in data retrieval

procedure from financial platforms. Certain issues were denied by considerations including repetition, resistance to settlement or being labeled as "not financial issue".

3.2 Data Preprocessing Using Text Pre-processing Techniques

Data preprocessing refers to the process of finding and fixing incorrect, inadequate, or unnecessary components of the data before changing, replacing or deleting irrelevant or unclean data from a dataset.

Tokenization: Tokenization is the process of breaking sentences into distinct phrases, words, and punctuation, which can be referred as tokens. The primary location to utilize the splitting criterion occurs when a punctuation or space occurs. In the following processing phases, this process contributes to filtering the unwanted words.

Stemming: An aggressive word-shortening method called stemming requires reducing a word to its fundamental format. The suffixes are removed to reduce it to the basic root and the semantic meaning of each variant remains the same. This rarely produces positive outcomes because the word becomes less meaningful.

3.3 Applying Term Frequency-Inverse Document Frequency (TF-IDF) for Feature Extraction

Following the pre-processing phase, each feature dimension is examined to extract significant phrases from the feature vector collection. The method is performed by employing the feature extraction approach referred as Term Frequency–Inverse Document Frequency (TF-IDF). The information retrieval method and numeric statistical measure known as TF-IDF are utilized to identify words in a corpus that are important and relevant to a certain document. The computation of these words is as follows.

$$TF\_IDF = TF * IDF \tag{1}$$

$$TF = \frac{Number\ of\ times\ the\ words\ occur\ in\ the\ text}{Total\ amount\ of\ words\ in\ text} \tag{2}$$

$$IDF = \frac{Total\ amount\ of\ documents}{amount\ of\ documents\ with\ the\ word\ (t)\ in\ it} \tag{3}$$

3.4 Bug Identification by Employing the Shuffled Frog Leaping-driven Dynamic One-class SVM (SFL-DOSVM)

3.4.1 Dynamic One-class Support Vector Machine (DOSVM)

An unsupervised variant of SVM called Dynamic One-Class SVM (DOSVM) trains the technique employing one class label rather than two. To divide points of information into two groups in the linear framework of this technique, the hyperplane is used. Several hyper planes can be required to correctly divide the data in many instances. DOSVM appears for the hyperplane with the largest margin to maximize the difference between the two groups. The shortest path between the closest data point for each category and the dividing hyperplane is added together to generate the margin. Equation (4) can be used to express it. DOSVM connects the input space to a feature space when the data cannot be separated linearly. An inside product is not utilized in this instance, whereas the kernel function is employed. Thus it makes it possible for a nonlinear decision boundary to develop.

$$E(x) = \beta_0 + \sum_{j=1}^{N} \alpha_j \langle y, y_j \rangle \tag{4}$$

Where y represents the new observations, y_j is an observation from the training data and β_0 is a bias factor. The parameters are represented by $\alpha_j, j = 1, \ldots, N$ (one for each training observation). It is

designed to indicate the instances that have the same group, similar to its two-class equivalent vector machine. It establishes boundaries for instances that are associated with the group because this technique only trains from one class. As an anomaly or outlier, each occurrence that cannot be represented inside the established border is categorized as belonging to the other group.

3.4.2 Shuffled Frog Leaping (SFL) Algorithm

The Shuffled Frog Leaping-driven (SFL) is the type of meta-heuristic optimization approach that mimics the memetic evolution of many frogs attempting to identify the location with the greatest food. It is predicated on the spread of memes through social interaction and global information sharing among participants. In the SFL, a group of homogeneous frogs is divided into groups known as memeplexes, each of which is allowed to develop independently so that they can explore the area distinctly. Each memeplex is thought of as a distinct culture of frogs that conducts a local search. Every frog is a viable solution to an optimization issue when the frogs serve as hosts or carriers of memes that are units of cultural development.

Consider that E randomly produced frogs Y_J comprise the starting population in the SFL. After that, use a known method to assess the level of $fit(j)$ of each frog. Following that, arrange the frogs according to their fitness values in decreasing order to split them into n memeplexes $X_1, X_2, \ldots, X_n$. A memeplex consists of m memeplexes, where m frogs are assumed. That $E = n \times m$ is apparent. One method to build the memeplex X^L is,

$$X^L = \left\{ Y_j^l | Y_j^l = Y_{l+n(j-1)}, j = 1,2, \ldots, m \right\} \qquad l = 1,2, \ldots., m \tag{5}$$

The m^{th} frog goes to the n^{th} memeplex, one frog for the first, another frog for the second and $(n + 1)^{th}$ frogs returned to the first memeplex in the way. The next stage is to continuously update the placements of the worst frog by running concurrent local evaluation algorithms in each sub-memeplex. The following rule updates the present location of the memeplex's worst frog:

$$E = rand \cdot (Y_{tbest} - Y_{worst})$$
$$Y_{worst}^{l+1} = Y_{worst}^{l} + C, \; C_{min} \le C \le C_{max} \tag{6}$$

Where the values of Y_{tbest} and Xwor st denote the submemeplex's best and worst frog positions. C is the frog's step size fluctuating; The lowest and maximum permitted changes to a frog's location are denoted by C_{min} and C_{max}. In the sub-memeplex, l is the number of iterations and $rand$ represents a random number between 0 and 1. Replace Y_{worst}^{l} with Y_{worst}^{l+1} if it performs better than the initial lowest performance Y_{worst}^{l}. If not, the above updating technique uses the global best frog, Y_{tbest} in place of Y_{tbest}. The following changes can be made to the operating process:

$$E = rand \cdot (Y_{tbest} - Y_{worst})$$
$$Y_{worst}^{l+1} = Y_{worst}^{l} + C, \; C_{min} \le C \le C_{max} \tag{7}$$

A random workable solution is created to take the place of the Y_{worst} if there is still no improvement in this situation. For a certain number of memetic development stages inside each memeplex, the process is repeated. For bug identification, all frogs are then shuffled. Local exploring and the shuffle operations take turns until the specified convergence threshold is reached.

Bug identification utilizing the Shuffled Frog Leaping-driven Dynamic One-class SVM (SFL-DOSVM) utilizes the benefits of the combined knowledge of dynamic one-class SVM (DOSVM) and shuffled frog leaping-driven (SFL) algorithm. By constantly altering data separation, this technique identified the anomaly or software faults in financial application. Repetitive process of the SFL-DOSVM

framework preserves financial estimation by increasing bug identification efficiency in the assessments. Effective performance in bug identification process is depicted by proposed method that constantly develops the design. By employing the repetitive method, it can obtain the optimal financial application and software usage to increase the accuracy of bug identification. In financial application the SFL-DOSVM model is an efficient technique for identifying and solving the issues of bugs.

4 Performance Evaluation and Discussion

This section provides the performance evaluation and discussion of proposed SFL-DOSVM for the identification of bugs. Research used python language, Windows 10, 4GB RAM and an Intel core i5-4210U CPU on the computer that estimated for this study. The comparison made between the SFL-DOSVM and various traditional techniques like LSTM (Bani-Salameh et al., 2021), KNN (Bani-Salameh et al., 2021) and SVM (Bani-Salameh et al., 2021) by using different metrics such as recall, f1-measure, accuracy, MCC and precision. Table 1 depicts the comparison of traditional and proposed methods.

Table 1: Comparison of Existing and Proposed Techniques

Methods	Parameters				
	Accuracy	Precision	Recall	F-Measure	MCC
LSTM Bani-Salameh et al., (2021)	0.898	0.876	0.897	0.892	0.796
SVM Bani-Salameh et al., (2021)	0.865	0.866	0.865	0.865	0.732
KNN Bani-Salameh et al., (2021)	0.741	0.743	0.741	0.742	0.485
SFL-DOSVM (Proposed)	0.981	0.965	0.972	0.954	0.946

Recall also known as sensitivity, the percentage of positive instances in the data collection is employed to evaluate the capacity of the model for identifying each true instance. The total is evaluated by dividing the total amount of true positives by the total amount of positive instances. An average precision and recall by determining inadequate outcomes and inaccurate negatives called f-measure. It is more effective if the outcomes of inaccurate results and false negatives are the same.

The f1- measure and recall comparison outcomes are displayed in Figure 2. Recall (0.972) and f1-measure (0.954) indicated by the SFL-DOSVM approach and f1- measure with 0.742 and recall with 0.741 performance are shown by KNN, LSTM (f1- measure 0.892 and recall of 0.897) and SVM demonstrates recall (0.865), f1- measure (0.865). The evaluation of the SFL-DOSVM method improved significantly in the f1-measure and recall when compared to existing methods.

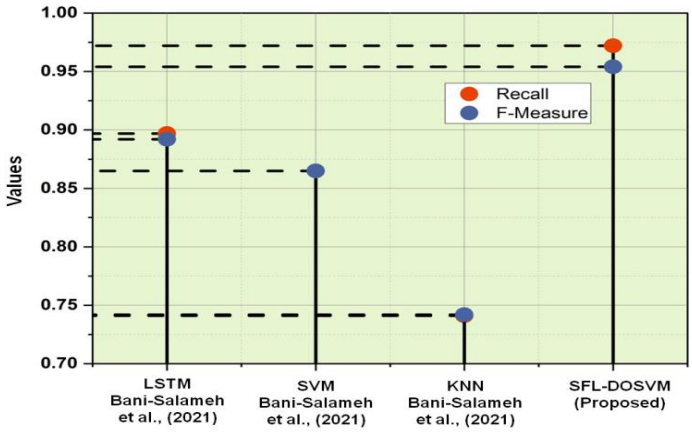

Figure 2: Comparison Outcomes of f1-measure and Recall

Research computed Mathews Coefficient Correlation (MCC) to determine the classifier's quality. Figure 3 presents the comparison between existing and proposed approaches. 0.796 Of MCC showed by LSTM, 0.732 MCC from SVM and KNN offers 0.485. The SFL-DOSVM technique provides 0.946 of MCC and it shows more efficiency than traditional techniques.

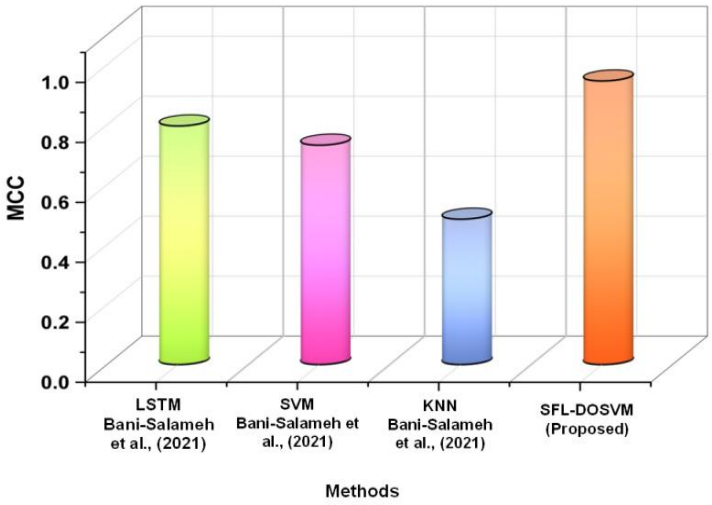

Figure 3: Comparison Results of MCC

The amount of truly predicted instances in every case is known as accuracy and it serves as significant performance data when utilizing unbalanced datasets, indicating occurrences when false negative and false positive values are equivalent. The ratio of accurately predicted positive forecasts to all positive forecasts is known as precision. In predicting the percentage of bug identification, precision was considered.

The comparison of the accuracy and precision with the conventional and proposed methods is depicted in Figure 4. While SVM (precision (0.866), accuracy (0.865)), KNN (accuracy (0.741), precision (0.743)) and LSTM deliver 0.898 of accuracy as well as 0.876 of precision, SFL-DOSVM displays the accuracy (0.981) and precision (0.965). While compared to existing techniques, our SFL-DOSVM approach has greatly improved.

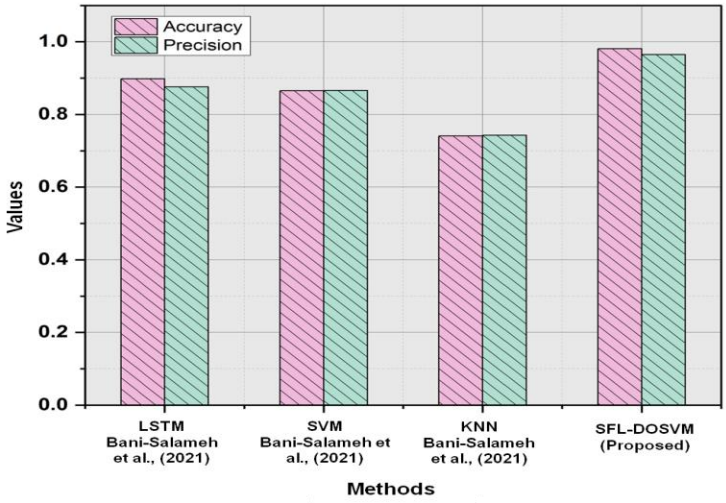

Figure 4: Outcomes of Accuracy and Precision Comparison

5 Conclusion

Using AI-integrated methods to classify bugs in financial systems is a significant effort that involves an integration of domain knowledge and technological capacity. The study's assessment shows that various kinds of bugs for financial applications can be appropriately identified and categorized using AI-integrated methods. An innovative SFL-DOSVM method for finding bugs in financial applications to improve security in the cloud is provided. By using different parameters, a comparison is provided between the SFL-DOSVM proposed approach and conventional methods. Our SFL-DOSVM technique performs more effectively by the offered parameters such as f1-measure (0.954), precision (0.965), recall (0.972), accuracy (0.981) and MCC (0.946). It has the potential to alter conclusions in the identification of bugs to resolve issues in financial applications. The implementation of an AI-integrated proposed approach to bug identification in financial applications is an appealing field for further study and development.

References

[1] Alwaheidi, M. K., & Islam, S. (2022). *Data-driven threat analysis for ensuring security in cloud enabled systems. Sensors, 22*(15), 5726. https://doi.org/10.3390/s22155726

[2] Aoudni, Y., Donald, C., Farouk, A., Sahay, K. B., Babu, D. V., Tripathi, V., & Dhabliya, D. (2022). Cloud security based attack detection using transductive learning integrated with Hidden Markov Model. *Pattern recognition letters, 157,* 16-26. https://doi.org/10.1016/j.patrec.2022.02.012

[3] Awadallah, R., Samsudin, A., Teh, J. S., & Almazrooie, M. (2021). An integrated architecture for maintaining security in cloud computing based on blockchain. *IEEE Access, 9,* 69513-69526. https://doi.org/10.1109/ACCESS.2021.3077123

[4] Bani-Salameh, H., Sallam, M., & Al shboul, B. (2021). A deep-learning-based bug priority prediction using RNN-LSTM neural networks. *e-Informatica Software Engineering Journal, 15*(1), 29-45. https://doi.org/10.37190/e-Inf210102

[5] Bhargava, R., Singh, Y. P., & Narawade, N. S. (2022, April). Deep Learning Dependent DDOS Attack Sensing in Environment of Cloud Computing. *In 2022 IEEE 7th International conference for Convergence in Technology (I2CT) (pp. 1-5).* IEEE. https://doi.org/10.1109/I2CT54291.2022.9825199

[6] Chen, Y., Ma, F., Zhou, Y., Jiang, Y., Chen, T., & Sun, J. (2023, May). Tyr: Finding consensus failure bugs in blockchain system with behaviour divergent model. *In 2023 IEEE Symposium on Security and Privacy (SP) (pp. 2517-2532).* IEEE. https://doi.org/10.1109/SP46215.2023.10179386

[7] Gajmal, Y. M., & Udayakumar, R. (2022). Privacy and utility-assisted data protection strategy for secure data sharing and retrieval in cloud system. *Information Security Journal: A Global Perspective, 31*(4), 451-465.

[8] Hai, T., Zhou, J., Li, N., Jain, S. K., Agrawal, S., & Dhaou, I. B. (2022). Cloud-based bug tracking software defects analysis using deep learning. *Journal of Cloud Computing, 11*(1), 32. https://doi.org/10.1186/s13677-022-00311-8

[9] Hirsch, T., & Hofer, B. (2022). Using textual bug reports to predict the fault category of software bugs. *Array, 15,* 100189. https://doi.org/10.1016/j.array.2022.100189

[10] Huy, D. T. N. (2018). The Risk Level of Viet Nam Commercial Electric Industry under Financial Leverage during and after the Global Crisis 2009-2011. *International Academic Journal of Organizational Behavior and Human Resource Management, 5*(1), 109–121. https://doi.org/10.9756/IAJOBHRM/V5I1/1810008

[11] Krichen, M. (2023). Strengthening the security of smart contracts through the power of artificial intelligence. *Computers, 12*(5), 107. https://doi.org/10.3390/computers12050107

[12] Kunduru, A. R. (2023). Artificial intelligence advantages in cloud Fintech application security. *Central Asian journal of mathematical theory and computer sciences, 4*(8), 48-53.

[13] Kunduru, A. R. (2023). Security concerns and solutions for enterprise cloud computing applications. *Asian Journal of Research in Computer Science, 15*(4), 24-33. https://doi.org/10.9734/AJRCOS/2023/v15i4327

[14] Madani, S. E., & Taha, O. M. (2024). Assessing the Effectiveness of Cloud Computing Adoption in Small and Medium Enterprises. *International Academic Journal of Innovative Research, 11*(3), 34–39. https://doi.org/10.71086/IAJIR/V11I3/IAJIR1122

[15] Pragadeswaran, S., Subha, N., Varunika, S., Moulishwar, P., Sanjay, R., Karthikeyan, P., ... & Vaasavathathaii, E. (2024). Energy Efficient Routing Protocol for Security Analysis Scheme Using Homomorphic Encryption. *Archives for Technical Sciences, 31*(2), 148-158. https://doi.org/10.70102/afts.2024.1631.148

[16] Sharma, A., & Pandey, A. (2024). Design and Implementation of a Secure Cloud-Based Storage System. *International Academic Journal of Science and Engineering, 11*(4), 22–26. https://doi.org/10.71086/IAJSE/V11I4/IAJSE1166

[17] Sharma, S., Lakshmanan, L., Sahu, P. K., Agarwal, T., & Tejeshwari, M. R. (2025). Blockchain for Library Records Management: A Secure and Decentralized Approach. *Indian Journal of Information Sources and Services, 15*(2), 28–37. https://doi.org/10.51983/ijiss-2025.IJISS.15.2.05

[18] Shreyas, S. (2023). Security model for cloud computing: case report of organizational vulnerability. *Journal of Information Security, 14*(4), 250-263. https://doi.org/10.4236/jis.2023.144015

[19] Stojanović, B., & Božić, J. (2022). Robust financial fraud alerting system based in the cloud environment. *Sensors, 22*(23), 9461. https://doi.org/10.3390/s22239461

[20] Tabassum, N., Namoun, A., Alyas, T., Tufail, A., Taqi, M., & Kim, K. H. (2023). Classification of bugs in cloud computing applications using machine learning techniques. *Applied Sciences, 13*(5), 2880. https://doi.org/10.3390/app13052880

[21] Torkura, K. A., Sukmana, M. I., Cheng, F., & Meinel, C. (2021). Continuous auditing and threat detection in multi-cloud infrastructure. *Computers & Security, 102,* 102124. https://doi.org/10.1016/j.cose.2020.102124

[22] Vinoth, S., Vemula, H. L., Haralayya, B., Mamgain, P., Hasan, M. F., & Naved, M. (2022). Application of cloud computing in banking and e-commerce and related security threats. *Materials Today: Proceedings, 51,* 2172-2175. https://doi.org/10.1016/j.matpr.2021.11.121

[23] Wu, X., Zheng, W., Chen, X., Zhao, Y., Yu, T., & Mu, D. (2021). Improving high-impact bug report prediction with combination of interactive machine learning and active learning. *Information and Software Technology, 133,* 106530. https://doi.org/10.1016/j.infsof.2021.106530

[24] Wu, Z., & Margarita, S. (2024). Based on Blockchain and Artificial Intelligence Technology: Building Crater Identification from Planetary Imagery. *Natural and Engineering Sciences, 9*(2), 19-32. https://doi.org/10.28978/nesciences.1567736

[25] Zhang, L. (2023). The dynamic and secure storage of enterprise financial data based on cloud platform. *International Journal of Critical Infrastructures, 19*(5), 449-462. https://doi.org/10.1504/IJCIS.2023.133285

Authors Biography

Bhagyashri Mankar is an Assistant Professor in the Department of Information Technology at Yeshwantrao Chavan College of Engineering, Nagpur, India. Her academic interests include data structures, software development, and emerging technologies in IT. She is committed to research, teaching, and student development.

Dr. Shoaib Mohammed is an Assistant Professor in the Department of ISME at ATLAS SkillTech University, Mumbai, Maharashtra, India. His academic and research interests span business strategy, innovation management, and entrepreneurship. He actively contributes to academic development through research and teaching.

Rahul Mishra is affiliated with the Centre of Research Impact and Outcome at Chitkara University, Rajpura, Punjab, India. His work focuses on research evaluation, academic impact analysis, and interdisciplinary innovation. He contributes to institutional research growth through data-driven strategies.

Rakhi Chakraborty is an Assistant Professor in the Department of Computer Science & IT at ARKA JAIN University, Jamshedpur, Jharkhand, India. Her academic interests include computer programming, data analytics, and emerging trends in IT. She is dedicated to research, curriculum development, and student mentoring.

Solomon Jebaraj is an Assistant Professor in the Department of Computer Science and Information Technology at Jain (Deemed-to-be University), Bangalore, Karnataka, India. His areas of expertise include software engineering, data science, and artificial intelligence. He is actively involved in academic research and mentoring students in advanced computing technologies.

Dr. Ritesh Rastogi is a Professor in the Department of Computer Science at Noida Institute of Engineering & Technology, Greater Noida, Uttar Pradesh, India. He specializes in software engineering, data analytics, and emerging computing technologies. With extensive academic and research experience, he contributes to scholarly publications and student development.