International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 1 (2026)

ijerst.editor@gmail.com

editor@ijerst.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 22, No. 1, 2026

Peer Reviewed, Referred & Indexed Journal

Research paper

A Secure Key-Aggregate Keyword Retrieval Scheme Over Encrypted Data in Cloud Computing

Mr. N. Kiran Kumar, M.Tech-Assistant Professor, Department of MCA, Bapatla Engineering College, Bapatla, Andhra Pradesh
Ms. JINKA SWATHI, Reg No: Y25MC23028, Ms. KAVUTARAPU VASUMATHI, Reg No: Y25MC23034,
Ms. GAJULA NAGALAKSHMI, Reg No: Y25MC23020, Ms. SYED KULSUMBE, Reg No: Y25MC23083, Department of MCA,
Bapatla Engineering College, Bapatla, Andhra Pradesh.

***Abstract*—Key-aggregate retrieval enables a cloud server, on behalf of delegated users, to perform keyword searches over data encrypted under multiple public keys. While existing key-aggregate searchable encryption schemes provide efficient data sharing, many of them remain vulnerable to keyword guessing attacks, leading to potential leakage of keyword ciphertexts and search trapdoors. Such vulnerabilities significantly undermine data privacy in cloud-based applications, particularly in emerging environments such as the Internet of Things (IoT).**

In this paper, we propose a secure and efficient key-aggregate keyword retrieval scheme for encrypted data in cloud computing. The proposed scheme allows a data owner to selectively share encrypted documents with multiple users using a single compact aggregate key, while enabling authorized users to delegate keyword search capabilities to the cloud server without revealing sensitive keyword information. To address privacy concerns, the scheme is designed to prevent keyword guessing attacks and ensure confidentiality of both keyword ciphertexts and trapdoors.

We formally define the security model of the proposed scheme and prove its security under the indistinguishable selective-file chosen keyword attack (IND-SF-CKA) model. Performance analysis and experimental simulations demonstrate that the proposed approach achieves strong security guarantees with low computation and communication overhead, making it practical for cloud-assisted IoT and large-scale encrypted data sharing applications.

Keywords: 1. Cloud Computing Security, 2. Key-Aggregate Encryption, 3. Searchable Encryption, 4. Keyword Retrieval, 5. Data Privacy, 6.Internet of Things (IoT)

I. INTRODUCTION

Cloud computing is really good for storing and sharing lots of data. It lets people use computers and store data over the internet from anywhere at any time. When people and companies store their data on servers they do not have to spend as much money on storing data on their own computers. They also do not have to do much work to keep their computers running smoothly. Cloud computing is very useful for things, like the Internet of Things and the Internet of Vehicles, where lots of devices that are not very powerful are making amounts of data all the time. Cloud computing is helpful because these devices do not have space to store all the Cloud computing data they are making.

Cloud computing has some things about it but it also has some big problems when it comes to keeping our personal information private and secure. This is a reason why a lot of

people are not using cloud computing as much as they could be. When we use cloud computing with internet connected devices and vehicles the people who own the data do not have control over what happens to their data.

We have to trust that the cloud servers are doing the thing but we also think that they might try to figure out secret information from the data they have or from the questions people ask. To make it safer people usually scramble their data before they send it to the cloud. Cloud computing is still a concern for people because of these security issues, with cloud computing.

Encryption is really good at keeping data secret. It makes it hard to get to that data when you need it. The old ways of doing things, where you have to download and decrypt all the data at once just do not work well for cloud storage systems especially when you are talking about Internet of Things devices that do not have a lot of power to compute things store data or communicate with other devices. So being able to search for things in encrypted data in a way that's efficient has become very important for cloud-based applications that need to be secure. Encryption and searching for data in the cloud is a challenge because encryption makes it hard to search for things. Efficient search over encrypted data is necessary, for cloud-based applications that use encryption to protect data.

The idea of Searchable Encryption was brought up to deal with this problem. It lets cloud servers look for words in

encrypted data without showing what the encrypted data really says or what words are being searched for. Searchable Encryption has changed over time. Now there are a few different types, like symmetric Searchable Encryption, public- key Searchable Encryption and identity-based Searchable Encryption. Each type of Searchable Encryption has its good and bad points when it comes to how well it works how safe it is and what it can do. While these methods do make it possible to search for keywords in a way most of them are really meant for use by one person at a time. Searchable Encryption is still a tool and people are working to make it better.

To help people share data, with each special encryption schemes were created. These schemes let many people search for and get the data they need in a way. On people did more research and added more features to these schemes.

They added things like access control, which means only certain people can see the data and fuzzy keyword search which helps people find the data they need even if they do not know the words. They also added queries, which let people ask for very specific data.. Most of the current solutions have a big problem. They encrypt all the shared data with one public key. This makes it very hard to use these solutions in life when many people are sharing data. The data sharing becomes very inflexible. It is hard to make it work for a large number of people. Multi-user searchable encryption schemes are still not very flexible and scalable. When people need to share information, with users who have different keys the people who own the information usually have to encrypt the same information many times. This means that the information owners have to do a lot of work and it takes up a lot of space. The information owners have to deal with data. They have to encrypt the data multiple times, which is a big problem because it uses a lot of computer power and storage space to encrypt the data.

To get around these problems people came up with aggregate encryption. This is a way to combine lots of decryption keys into one key that works for certain encrypted messages. When you use aggregate encryption with searchable encryption it is a good way to search for keywords in encrypted data stored in the cloud. You can do this quickly and safely.

However the methods we have now for aggregate searchable encryption are not very good, at keeping keywords safe. If someone tries to guess the keywords they might be able to figure out what the encrypted keywords say and even get the search information. This means that the privacy of the search is not protected. Aggregate encryption is still a good idea but we need to make it better so that key-aggregate encryption can really keep our data safe.

Motivated by these challenges, this paper focuses on designing a secure and efficient key-aggregate keyword retrieval scheme suitable for cloud-assisted IoT applications. The proposed approach enables fine-grained data sharing across multiple users encrypted under different public keys, while allowing authorized keyword search delegation to the cloud without leaking sensitive keyword information. By addressing both security and efficiency concerns, the proposed scheme aims to provide a practical solution for privacy-preserving data retrieval in modern cloud computing environments.

II. LITERATURE SURVEY

LITERATURE SURVEY Searchable encryption has been extensively studied as a solution for enabling keyword search over encrypted data while preserving data confidentiality. Existing research spans multiple models, including symmetric searchable encryption, public-key

searchable encryption, and identity-based searchable encryption. Early schemes were primarily designed for single-user environments, where only one authorized receiver could generate search queries. However, such models are insufficient for practical cloud storage systems, where data owners often need to share encrypted documents with multiple users.

To address this limitation, multi-user searchable encryption schemes were proposed, allowing multiple authorized users to perform keyword searches over shared encrypted data. Initial approaches assumed that all documents were encrypted under a single public key, with search privileges regulated either by a trusted user manager or through pre-established coordination with the cloud server. Although functional, these schemes suffer from poor scalability, as ciphertext size or management overhead often increases with the number of users.

Subsequent research extended multi-user searchable encryption to support more expressive query capabilities, including boolean queries, wildcard search, range queries, and substring matching. Attribute-based encryption with keyword search further enabled fine-grained access control by associating access policies with ciphertexts. However, these solutions incur high computational and storage overhead, as the sizes of secret keys and trapdoors grow with the number of attributes, limiting their efficiency in large-scale systems.

Another research direction focuses on multi-key searchable encryption, which allows different documents to be encrypted under different keys while still supporting keyword search across all ciphertexts. Early frameworks demonstrated the feasibility

of searching over multi-key encrypted data, but the size of secret keys and trapdoors increased linearly with the number of searchable files. Although later schemes improved security definitions and functionality, key-size inefficiency remained a significant challenge.

To improve scalability, key-aggregate searchable encryption (KASE) was introduced, enabling a data user to obtain a single compact aggregate key that supports constant-size trapdoors for searching across multiple authorized documents. While KASE significantly reduces key and trapdoor overhead, subsequent studies revealed critical security weaknesses. In particular, aggregate keys were shown to potentially leak information about accessible documents, and several schemes were found vulnerable to keyword-guessing attacks, leading to the complete compromise of keyword and trapdoor privacy.

Summary: Existing searchable encryption schemes provide important functionality for secure data retrieval in cloud environments, but they face persistent challenges related to scalability, efficiency, and security. In particular, vulnerabilities to keyword-guessing attacks and inefficient key management motivate the need for more secure and practical key-aggregate keyword retrieval schemes suitable for multi-user cloud computing scenarios.

III. METHODOLOGY

This part is about how the proposed key-aggregate keyword retrieval scheme for encrypted data in cloud computing works. The main goal of the design is to make sure that people can search for keywords, in encrypted data in an efficient way. This is important when many users share the data. The secure aggregate keyword retrieval scheme is designed to prevent people from guessing keywords and to reduce the amount of work needed to manage keys. The secure aggregate keyword retrieval scheme should be flexible and secure so that users can trust it to keep their data safe.

A. System Model:

The system they are talking about has four parts:

* The system
* The users of the system
* The people who manage the system
* The system itself is also made up of things that help it work properly which is the system.

So the system is made up of these four things that make the system work. The system is what they are trying to explain.

Data Owner (DO):

The person who owns the data does a things. They encrypt the documents so they are safe. Then they make a list of what's in the documents so people can search for things. After that they put the encrypted data on the internet in a place called the cloud.

The data owner also decides who can look at the documents and what they can do with them. They make keys, for the people who are allowed to see the documents and these keys help the authorized users see the information they need.

Data User (DU):

People who are allowed to use the system and have a secret key can make a search trapdoor to find documents that have certain keywords in them. These authorized users can use the search trapdoor to get the documents they need. The documents they find will have the keywords that they are looking for.

Cloud Server (CS):

A semi-trusted entity is like a storage place that keeps documents safe by encrypting them. This storage place can look for words in these documents without actually seeing what is inside the documents or knowing what the person is searching for. The entity uses something called trapdoors to do the searching. The semi-trusted entity does all this without getting to see the real information in the documents or knowing what the search is, about.

Key Authority (optional):

Assists in system initialization and parameter generation, if required.

The cloud server is like a person who does what they are supposed to do. They might also try to figure out some secret things about the cloud servers users. The cloud server follows the rules. It is curious, about sensitive information that it is not supposed to know.

B. Threat Model and Security Goals

The plan they have, in mind is supposed to protect against: When someone is trying to figure out the keywords they do

something called a keyword guessing attack. This is where the bad person tries to guess the keywords from the code or from a trapdoor. The bad person is trying to infer the keywords from the messages. Keyword guessing attacks are a problem because the bad person is trying to get the keywords, from the ciphertexts or trapdoors.

Trapdoor privacy leakage, preventing the cloud server from learning query contents.

We need to make sure that only the right people can get to information. This means we have to stop data access. We have to ensure that authorized users can retrieve the documents they are supposed to see which are the matching documents.

The main thing we want to do for security is to make sure our system is safe from kinds of attacks. These attacks are called file chosen keyword attacks. We want to make it so that these attacks cannot tell our system apart, from any system. This is what we call indistinguishability under file chosen keyword attacks.

C. Key-Aggregate Searchable Encryption Framework

The way we do this is, by combining key-aggregate encryption with encryption. This combination helps us to make it easy for many users to search for keywords at the time. We use key-aggregate encryption and searchable encryption together to make this work.

System Setup:

The system starts by setting up some parameters and a master secret key. The system then shares these parameters with every single participant so they all have the same information to work with. The master secret key is also part of the system. The public parameters are shared with all participants.

Key Generation:

The person who owns the data makes a pair of keys one that everyone can see and one that is secret. They use these keys to lock up the documents. When someone is allowed to see these documents the data owner gives them a key that lets them unlock just the documents they are supposed to see. This special key is, like a master key that works for a bunch of documents at the time and it is made just for that person. The data owner makes sure that each person only gets to see the documents they are supposed to see by using these keys.

Document Encryption and Index Construction:

Each document is locked with encryption. A special list called a keyword index is made to keep things safe. This list connects encrypted keywords to the names of the documents. It does this without giving away what the keywords are. The keyword index is really good, at keeping the keywords secret. The documents and the keyword index work together to keep everything secure.

Aggregate Key Distribution:

The aggregate key is given to people who are allowed to have it. It is done in a very safe way. The aggregate key is always the size it does not change even if you are sharing a lot of documents, with the aggregate key.

D. Trapdoor Generation and Keyword Search

People who are allowed to use the system make a search tool for a specific word they are looking for using a secret code that they all share. This search tool is made so that it keeps the search private. It is hard for someone to guess the word they are looking for. When the search tool is sent to the computer that stores all the information it looks through all the lists and only sends back the secret files that match the word they are looking for without knowing what the word is or what is in the files. The search tool for the queried keyword is really important for the whole process to work. The queried keyword is what the whole search is, about.

E. Keyword-Guessing Attack Resistance

To stop people from guessing keywords the system uses parts to encrypt keywords and make trapdoors. This means that even if two keywords are the same they will look totally different when they are encrypted and turned into trapdoors. This happens every time someone searches for something and every time a different person uses the system. The reason for this is to keep keywords safe, from others. The keyword privacy is protected because the system makes sure that the same keywords do not produce the encrypted keywords and trapdoors. This is what the system does to protect keyword privacy.

F. Performance Evaluation Strategy

We need to see how the proposed scheme works. This is done by looking at the following things, about the proposed scheme:

1. Aggregate key size,
2. Trapdoor generation cost,
3. Search complexity at the cloud server, and
4. Communication overhead.

The proposed scheme is really good. It has constant size aggregate keys and trapdoors. This means it does not need a lot of power to work. Theoretical. Experimental evaluation of the proposed scheme demonstrate that it is good. Because of this the proposed scheme is suitable for environments like cloud and Internet of Things. The proposed scheme achieves low computation overhead which's great, for large scale cloud and Internet of Things environments.

G. Methodology Summary

The proposed methodology combines key-aggregate encryption and searchable encryption to enable secure, scalable, and privacy-preserving keyword retrieval over encrypted data in cloud computing. By addressing key management efficiency and keyword-guessing vulnerabilities, the scheme provides a practical solution for secure data sharing in multi-user cloud environments.

IV. **ALGORITHM**

Secure Key-Aggregate Keyword Retrieval Scheme Input: Security parameter λ
Document set D = D1, D2, ...Dn Keyword set W
Authorised document index set S ¡ 1, 2, ..., n Output:
Encrypted documents
Secure keyword search results

Step 1: System Setup Generate public system parameters PP and master secret key MSK using security parameter λ Publish PP; keep MSK secret with the data owner.

Step 2: Key Generation Data owner generates a pub- lic–private key pair (p k, s k) .
For each document D i, associate a document identifier i.

Step 3: Document Encryption and Index Construction For each document D i $\in D$
Encrypt document content using P K to obtain ciphertext C
i

Extract keywords w $\in$ *WfromDi*
Generate encrypted keyword indexes for each W
Upload C i and encrypted indexes to the cloud server.

Step 4: Aggregate Key Generation For an authorized user with access set S:
Compute a single aggregate secret key K S = AggregateKey(MSK, S)
Securely distribute KS to the authorized user.

Step 5: Trapdoor Generation For a query keyword w q, the user generates a search trapdoor:
Twq = TrapdoorGen(KS, wq)
Send Twq to the cloud server.

Step 6: Keyword Search Upon receiving Twq, the cloud server:

Matches Twq against encrypted keyword indexes. Identifies ciphertexts Ci such that i $\in$ *SandDicontainsWq* Return matching encrypted documents to the user.

Step 7: Document Decryption The authorized user decrypts returned ciphertexts using the appropriate secret key material. Obtain plaintext documents containing the queried keyword.

Step 8: Security Enforcement Ensure trapdoors and key- word ciphertexts are randomized to prevent keyword-guessing attacks. Cloud server learns neither plaintext data nor keyword information.

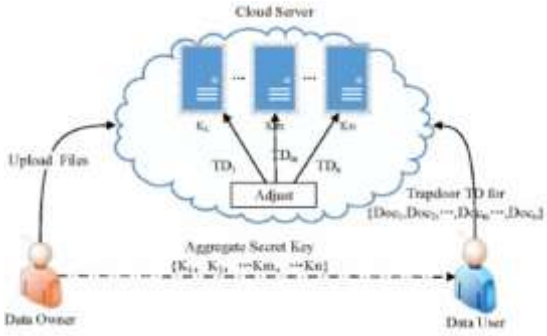

Fig. 1. Architecture diagram

V. RESULT ANALYSIS

This part looks at how the secure key-aggregate keyword retrieval scheme works for encrypted data in cloud computing. It checks if the scheme is really secure and effective. The evaluation is about making sure the secure aggregate keyword retrieval scheme works as it should and that it is safe, from attacks. It also checks how efficient and scalable the secure aggregate keyword retrieval scheme is when many users are using it in cloud computing.

A. Functional Correctness:

The new plan was checked to make sure it works correctly when people search for keywords in documents that are shared with many users. When users who are allowed to see these documents use a secret key they can look for the keywords they want and get all the secret documents that have those keywords. People who are not supposed to see these documents and the computer server that stores them cannot get to the text or figure out what the keywords are which shows that the plan is good, at controlling who can access the documents and search for things.

B. Security Analysis Results:

The system is really good at stopping people from guessing keywords, which's a big problem with other systems that let you search for encrypted keys. Because the system makes keyword codes and search codes in a way the same keyword will look totally different when you search for it at different times. This means that even if someone, like the company that stores your data tries to guess keywords when no one is looking they will not be able to figure out what the keywords are. The keyword guessing attacks are not possible because the system makes it so that the same keyword looks different every time you search for it.

The scheme is really good because it meets the security standards for keeping things secret. This means that even if someone looks at the encrypted indexes or trapdoors they will not be able to figure out what the keywords are or what is in the documents. The scheme is safe from something called a file chosen keyword attack, which is a way that someone might try to get secret information. This is important because it means that the scheme is secure and people can trust it to keep their information private. The scheme satisfies something called the indistinguishability under file chosen keyword attack security model, which is a big phrase that just means it is really good, at keeping things secret.

C. Efficiency Evaluation:

The performance analysis shows that the proposed scheme is really good at keeping the keys and trapdoors a constant size. This means it does not matter how many documents are shared. The proposed scheme always keeps the size for these keys and trapdoors. This is a deal because it reduces the work needed to manage all these keys. Traditional multi-key searchable encryption schemes are not as good, at this. For these schemes the size of the keys and trapdoors gets bigger as more files are added.

The Trapdoor generation and the keyword search operations do not use a lot of computer power. This makes the Trapdoor generation and the keyword search operations a good choice, for systems that use the cloud to help them work, like Internet of Things systems that do not have a lot of power to start with. The cloud server can do the search operations quickly. It does this

without looking at the actual data, which means the Trapdoor generation and the keyword search operations can handle a lot of data.

D. Scalability and Practicality:

The system works well when you have a lot of users and documents.

Data owners only have to share one key for each user so it does not take up much space or time to communicate even

when a lot of people are using it.

When we tried it out we saw that it takes a little time to search when you have more documents stored but this is okay for real world cloud storage systems like the cloud storage system.

The search time for the cloud storage system increases in a line, with the number of documents stored in the cloud storage system.

E. Comparative Discussion:

The new way of doing things is better than the key- aggregate searchable encryption schemes. It is more secure because it gets rid of the problems that happen when people try to guess keywords. This new way is just as fast or even faster than the way.

The old multi-user searchable encryption methods are not as good as this scheme. This is because the new scheme does not need to encrypt data over again. It also does not need to give out a lot of keys. This makes it more useful for world cloud computing situations. The proposed approach is an improvement, over existing aggregate searchable encryption schemes.

F. Summary of Results:

The results, from the experiments and the analysis show that the proposed scheme is an idea. The proposed scheme really. This is what the experiments and analysis found out about the proposed scheme.

Supports secure and correct keyword retrieval over en- crypted data, Effectively resists keyword-guessing and trap- door leakage attacks. The system gets size aggregate keys and trapdoors it really gets these constant-size aggregate keys and trapdoors.

Reduces key management and communication overhead, and Is scalable and practical for multi-user cloud and IoT applications.

Fig. 2. Home Screen

Fig .3 .Data Owner Login Page

Fig . 4 Upload Report

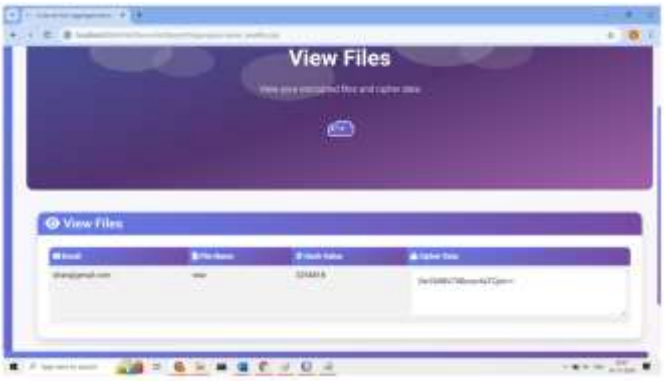

Fig . 5 Encrypted Data

This paper is a way to keep data safe in cloud computing. We came up with a plan that lets the person who owns the data share lots of encrypted files with people they trust. They can do this using one key that has all the important information in it. This way the cloud server can help find words in the files without actually seeing what is in them.

We made this plan by combining a few ideas. We used something called key-aggregate encryption and something called encryption. We also added some elements to the way we scramble the data. This helps stop people from guessing what the keywords are. It also keeps the keywords. The way we search for them private. The data owner and the people they share with can be sure that cloud computing is secure, with this plan.

The security analysis shows that the proposed scheme is really good at keeping things secret. It can protect against

people trying to figure out the keywords, which's a big deal. This means that the scheme is very good at keeping information from getting out.

When we look at how the scheme works we see that it does a great job of keeping the keys and trapdoors small. This makes it much better than solutions that are out there because it saves a lot of space and time. The proposed scheme is great for world situations like when we use cloud storage or have a lot of devices connected to the internet like in Internet of Things systems. The scheme is practical. It can handle a lot of things at the same time, which is very important, for these kinds of environments.

Overall, the proposed key-aggregate keyword retrieval scheme provides a secure, efficient, and flexible solution for privacy-preserving data sharing and retrieval in cloud computing. Future work may focus on extending the scheme to support more expressive search queries, dynamic user revocation, and deployment in real-world large-scale cloud systems.

VI. CONCLUSION

Cloud computing is really popular for storing and sharing amounts of data especially when lots of people are using it like with Internet of Things applications.. It is still a big problem to keep data private while also being able to search for specific information in encrypted data. The main issue is that current methods for searching encrypted data, like encryption and key-aggregate searchable encryption have some major flaws. Cloud computing has issues, with being able to handle a number of users managing all the keys is a hassle and it is not that hard for someone to guess the keywords. Cloud computing and data storage are affected by these problems.

REFERENCES

[1] M. Armbrust, A. Fox, R. Griffith, et al., "A view of cloud computing, "Communications of the ACM, vol. 53, no. 4, pp. 50-58, 2010.

[2] L. Ale, N. Zhang, H. Wu, et. al. "Online Proactive Caching in Mobile Edge Computing Using Bidirectional Deep Recurrent Neural Network, Reinterned of Things Journal, vol. 6, no. 3, pp. 5520-5530, 2019.

[3] J. Li, Q. Yu, Y. Zhang, et al., "Key-policy attribute-based encryption againstcontinual auxiliary input leakage", Information Sciences, vol. 470, pp. 175-188, 2019.

[4] N Zhang, P Yang, J Ren, et.al., "Synergy of big data and 5g wireless networks: opportunities, approaches, and challenges", IEEE Wireless Com-medications vol. 25, no.1, pp.12-18, 2018.

[5] A. Botta, W. De Donato, V. Persico, et al., "Integration of cloud computing and internet of things: a survey", Future Generation Computer Systems, vol.56, pp. 684-700, 2016.

[6] Y. Miao, J. Ma, X. Liu, et al., "Lightweight fine-grained search over en-crypted data in fog computing", IEEE Transactions on Services Computing, Vol. 12, no. 5, pp. 772-985, 2019.

[7] D. Chen, N. Zhang, et al., "An LDPC code based physical layer message authentication scheme with prefect security", IEEE Journal on Selected Areas in Communications, vol. 36, no. 4, pp. 748-761, 2018.

[8] Rongali, L. P. (2025). Utilizing AI-Driven DevOps for Predictive Maintenance and Anomaly Detection in Smart Grids. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.5229587

[9] Nandigama, N. C. (2019). Energy-Efficient Virtual Machine Placement in Cloud Data Centers Using Reinforcement Learning-Enhanced Adaptive Greedy Dingo Optimization Algorithm. Research Journal of Nanoscience and Engineering, 3(1), 25–32. https://doi.org/10.22259/2637-5591.0301006

[10] Jonnalagadda, A. K., Natarajan, G. N., Veerapaneni, S. M., & Vikram, S. (2025). Edge-Aware Federated AI: Scalable LLM Integration for Privacy-Preserving Big Data Networks. 2025 5th International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), 1–7. https://doi.org/10.1109/iceccme64568.2025.11277672

[11] Y. Chen, N. Zhang, Y. Zhang, X. Chen, W. Wu, X. Shen. Energy Ef-client Dynamic Offloading in Mobile Edge Computing for Internet of Things. IEEE Transactions on Cloud Computing, accepted, to appear, DOI10.1109/TCC.2019.2898657.

[12] Y. Chen, N. Zhang, Y. Zhang, X. Chen, W. Wu, X. Shen. TOFFEE: TaskOffloading and Frequency Scaling for Energy Efficiency of Mobile Devices in Mobile Edge Computing. IEEE Transactions on Cloud Computing, accepted, to appear, DOI 10.1109/TCC.2019.2923692.

[13] Todupunuri, A. (2025). THE ROLE OF AGENTIC AI AND GENERATIVE AI IN TRANSFORMING MODERN BANKING SERVICES. American Journal of AI Cyber Computing Management, 5(3), 85–93. https://doi.org/10.64751/ajaccm.2025.v5.n3.pp85-93.

[14] M. Jia, D. Li, Z. Yin, et al. "High Spectral Efficiency Secure Communications With Nonorthogonal Physical and Multiple Access Layers", Reinterned of Things Journal, vol. 6, no. 4, pp. 5954-5961, 2018.

[15] Nandigama, N. C. (2018). Deep Vision Networks for Multimodal Biometric Authentication: A Hybrid Feature-Level Fusion Approach with Machine Learning Optimization. Research Journal of Nanoscience and Engineering, 2(4), 22–29. https://doi.org/10.22259/2637-5591.0204005

[16] N. Zhang, N. Cheng, A. Gamage, K. Zhang, J.W. Mark, and X. Shen, "Cloud Assisted HetNets Toward 5G Wireless Networks," IEEE Communization Magazine, vol. 53, no. 6, pp. 59 - 65, 2015.

[17] L. Hou, S. Zhao, X. Xiong, K. Zheng, P. Chaoticities, MS Hossain, and W.Xiang, "Internet of things cloud: architecture and implementation," Telecommunication's Magazine, vol.54, no. 12, pp.32-39, 2016.

[18] D. Chen, N. Zhang, Z. Qin, X. Mao, Z. Qin, X. Shen, and X. Y. Li, "S2M:A Lightweight Acoustic Fingerprints-Based Wireless Device Authentication Protocol," IEEE Internet of Things Journal, vol. 4, no. 1, pp. 468-47788-100, 2017..