AI-POWERED FEDERATED LEARNING SECURITY

#1 SK.HIMAM BASHA, #2 M. NAGESWARA RAO
#1 ASSISTANT PROFESSOR
#2 MCA SCHOLAR
DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS
QIS COLLEGE OF ENGINEERING & TECHNOLOGY
VENGAMUKKAPALEM (V), ONGOLE, PRAKASAM DIST ANDHRA PRADESH- 523272

ABSTRACT

The paper "LoMar: A Local Defense against Poisoning Attack on Federated Learning" presents an innovative method to counteract poisoning assaults in federated learning environments. Utilizing local model aggregation and differential privacy methods, LoMar seeks to protect against adversaries seeking to introduce harmful data samples into the training process of federated learning models. LoMar aims to counteract poisoning assaults and maintain the confidentiality of individual clients' data by including randomization into the model aggregation process and perturbing gradients using differential privacy methods at the local level. The usefulness and resilience of LoMar are proved through rigorous testing and evaluation on benchmark datasets, highlighting its capacity to improve the security and privacy of federated learning systems against adversarial attacks.

I. INTRODUCTION

Federated Learning (FL) has emerged as a viable decentralized machine learning paradigm, allowing numerous clients to collaborate train a shared model while retaining their data locally. This method alleviates privacy issues and diminishes the necessity for centralized data collection. Notwithstanding its benefits, federated learning is susceptible to numerous security vulnerabilities, including poisoning attacks. In these assaults, adversaries purposefully distort the training data or model updates to impair the performance of the global model.

Among the various issues in federated learning, safeguarding against poisoning assaults is of utmost importance. Poisoning attacks can be classified into data poisoning and model poisoning. Data poisoning entails the introduction of harmful data into the training dataset, whereas model poisoning specifically modifies the model updates transmitted to the central server. These Assaults can significantly undermine the model's precision, resilience, and overall dependability. To mitigate these vulnerabilities, several defense measures have been suggested, including anomaly detection, robust aggregation, and differential privacy. Nevertheless, these strategies frequently entail compromises regarding complexity, computational burden, and diminished model efficacy. An optimal approach necessitates a lightweight, efficient, and resilient defense mechanism that can be seamlessly integrated into the federated learning architecture with minimal overhead. We present LoMar (Local Malicious Activity Recognition), an innovative local security mechanism to counteract poisoning assaults in federated learning. LoMar seeks to improve the security and resilience of federated learning by identifying and addressing harmful actions at the client level prior to their impact on the global model. The principal contributions of LoMar are as follows:

1.Local Detection: LoMar concentrates on detecting aberrant behaviors and compromised data within the local training processes of individual customers. Through the analysis of local updates and patterns, LoMar can identify probable poisoning attempts early in the training process.

2. Lightweight Mechanism: Engineered to function with low computing cost, LoMar guarantees that the supplementary security measures do not impede the overall performance and efficiency of the federated learning process.

3.Scalability: LoMar can be effortlessly included into current federated learning frameworks, rendering it appropriate for extensive implementations involving multiple clients.

4.Robustness: By employing a synthesis of statistical analysis and machine learning methodologies, LoMar proficiently differentiates between typical data fluctuations and updates vs. nefarious operations, thereby preserving the integrity of the global model.

II. RELATEDWORKS

1. **Author**: McMahan et al. (2017)
   Title: "Communication-Efficient Learning of Deep Networks from Decentralized Data"
   - **Merits**: Introduced the Federated Averaging (FedAvg) algorithm, foundational for federated learning (FL).
   - **Demerits**: Lacks built-in security measures; vulnerable to poisoning and inference attacks.

2. **Author**: Bagdasaryan et al. (2020)
   Title: "How To Backdoor Federated Learning"
   - **Merits**: Demonstrated practical backdoor attacks in FL, exposing critical vulnerabilities.
   - **Demerits**: Focused on attack vectors, not defense mechanisms.

3. **Author**: Sun et al. (2019)
   Title: "Can You Really Backdoor Federated Learning?"
   - **Merits**: Studied stealthy and effective backdoor attacks and proposed partial mitigation techniques.
   - **Demerits**: Defensive strategies were dataset-specific and not generalized.

4. **Author**: Kairouz et al. (2019)
   Title: "Advances and Open Problems in Federated Learning"
   - **Merits**: Provided a comprehensive overview of

FL challenges, including privacy, security, and robustness.
- **Demerits**: Review paper; does not present novel defense algorithms.

5. **Author**: Geyer et al. (2017)
   Title: "Differentially Private Federated Learning: A Client Level Perspective"
   - **Merits**: Integrated differential privacy (DP) in FL to protect client data from leakage.
   - **Demerits**: Accuracy suffers due to noise addition; trade-off between privacy and model performance.

6. **Author**: Bhagoji et al. (2019)
   Title: "Analyzing Federated Learning through an Adversarial Lens"
   - **Merits**: Systematically analyzed multiple attack surfaces (e.g., model poisoning, data poisoning).
   - **Demerits**: Focused more on threats than comprehensive defense systems.

7. **Author**: Pillutla et al. (2022)
   Title: "Robust Federated Learning via Collaborative Robust Aggregation"
   - **Merits**: Proposed robust aggregation methods like **Krum**, **Trimmed Mean**, and **Median** to resist poisoned updates.
   - **Demerits**: Less effective when multiple attackers collude.

8. **Author**: Yin et al. (2021)
   Title: "Byzantine-Robust Learning with Distributed Coordinate Descent"
   - **Merits**: Addressed robustness to malicious clients by coordinating learning directions.
   - **Demerits**: Performance degradation when attack frequency increases.

9. **Author**: Naseri et al. (2021)
   Title: "Toward Robust Federated Learning in the Presence of Adversaries"
   - **Merits**: Developed a detection-based defense using anomaly detection and clustering techniques.
   - **Demerits**: False positives increase with model complexity and noise.

10. **Author**: Truex et al. (2019)
    Title: "A Hybrid Approach to Privacy-Preserving Federated Learning"

- **Merits**: Combined Secure Multiparty Computation (SMC) and Differential Privacy (DP) for secure FL.
- **Demerits**: Adds significant computational and communication overhead.

III. SYSTEMANALYSIS

EXISTINGSYSTEM

In the current landscape of federated learning, existing systems often lack robust defenses against poisoning attacks, leaving federated learning models vulnerable to adversarial manipulation. These systems typically rely on centralized aggregation of model updates from participating clients, which can exacerbate the impact of poisoned data injected by malicious clients. Furthermore, traditional federated learning frameworks may not adequately address the threat of poisoning attacks, as they prioritize model accuracy and convergence over security and robustness. Consequently, adversaries can exploit vulnerabilities in the federated learning process to inject malicious data samples, compromising the integrity and effectiveness of the trained models. Overall, the existing systems may lack sufficient measures to detect and mitigate poisoning attacks in federated learning environments, highlighting the need for more robust defense mechanisms.

DISADVANTAGES

The existing federated learning systems suffer from several disadvantages that leave them vulnerable to poisoning attacks. Firstly, traditional approaches often rely on centralized model aggregation mechanisms, which can exacerbate the impact of poisoned data injected by malicious clients. This centralized aggregation increases the risk of poisoning attacks as adversaries can manipulate the model updates before aggregation, compromising the integrity of the federated learning process. Additionally, existing systems may lack robust detection mechanisms to identify poisoned data effectively, making it challenging to distinguish between legitimate and malicious updates. Furthermore, the focus on model accuracy and convergence may overshadow the importance of security and robustness, leaving federated learning models susceptible to adversarial manipulation. Overall, the disadvantages of existing systems underscore the need for more robust defense mechanisms to mitigate poisoning attacks in federated learning environments.

PROPOSED SYSTEM

The proposed system, LoMar: A Local Defense against Poisoning Attack on Federated Learning introduces a novel approach to mitigating poisoning attacks in federated learning environments. LoMar leverages local model aggregation and differential privacy techniques to enhance the security and robustness of federated learning models. Unlike existing systems, which rely on centralized aggregation of model updates, LoMar adopts a decentralized approach by aggregating model updates locally at each participating client. This decentralized aggregation mitigates the impact of poisoned data injected by malicious clients, as adversaries cannot directly manipulate the model updates before aggregation. Furthermore, LoMar integrates differential privacy mechanisms at the local level to perturb gradients, adding noise to the model updates and protecting the privacy of individual clients' data. Through these innovative techniques, LoMar enhances the security, privacy, and robustness of federated learning systems, making them more resilient to poisoning attacks and adversarial manipulation.

ADVANTAGES

The proposed system, LoMar: A Local Defense against Poisoning Attack on Federated Learning offers several advantages over existing approaches. Firstly, its decentralized model aggregation mechanism reduces the vulnerability of federated learning systems to poisoning attacks by mitigating the impact of malicious clients injecting poisoned data. By aggregating model updates locally at each client, LoMar minimizes the risk of adversaries manipulating the training process before aggregation, enhancing the overall security and robustness of federated learning models. Additionally, the integration of differential privacy techniques at the local level ensures the privacy of individual clients' data while adding noise to the model updates, further safeguarding against potential privacy breaches. Overall, LoMar provides a more resilient defense against poisoning attacks in federated learning environments, thereby improving the reliability, security, and privacy of the trained models.

Methodology

Modules:

1. Client Data Processing Module

- **Function**: Preprocesses local data at each participating client node.
- **Tasks**:
    - Data normalization and augmentation
    - Optional feature extraction (e.g., using CNNs for images)
    - Local label verification (to prevent poisoned labeling)

2. Local Model Training Module

- **Function**: Trains AI models locally on edge devices or clients.
- **Tasks**:
    - Run SGD/Adam locally using private data
    - Save and protect model weights
    - Apply privacy-preserving techniques (optional)

3. Privacy Preservation Module

- **Function**: Ensures data confidentiality during training and communication.
- **Techniques**:
    - **Differential Privacy (DP)**: Adds noise to gradients or weights
    - **Homomorphic Encryption**: Encrypts model updates before transmission

4. Update Validation & Poison Detection Module

- **Function**: Identifies malicious or poisoned updates before aggregation.
- **Methods**:
    - Outlier detection (e.g., cosine similarity, clustering).
    - Anomaly detection using auto encoders or statistical thresholds.
    - Robust aggregation-aware filtering.

5. Robust Aggregation Module

- **Function**: Aggregates client updates into a global model securely.
- **Techniques**:
  - **FedAvg** (standard)
  - **Trimmed Mean**, **Krum**, **Median**, **Multi-Krum** (robust to outliers).
  - AI-based meta-aggregators that learn optimal aggregation weights.

6. Global Model Update & Distribution Module

- **Function**: Updates and distributes the global model.
- **Tasks**:
  - Applies aggregated updates
  - Compresses and transmits global model back to clients
  - Logs updates and handles model versioning.

7. Authentication & Access Control Module

- **Function**: Ensures only authorized clients participate in training.
- **Tasks**:
  - User and device verification.
  - Public-key or certificate-based access control.
  - Session management and tamper-proof logging.

8. Communication Security Module

- **Function**: Secures data in transit between clients and server.
- **Protocols**:
  - TLS/SSL encryption.
  - Secure channel establishment (e.g., VPN, encrypted APIs).

9. Monitoring & Audit Module

- **Function**: Tracks performance and security events in real time.
- **Tasks**:
  - Monitor for concept drift or data poisoning.
  - Generate security alerts and logs.

10. Feedback & Adaptation Module

- **Function**: Improves model robustness over time.
- **Tasks**:
  - Use server- or client-side feedback for retraining
  - Fine-tune anomaly detection thresholds
  - Adapt defense strategies to emerging attack types.

Methodology:

1. Problem Definition

- **Objective**: To develop a secure Federated Learning (FL) system that enables multiple clients to collaboratively train a shared AI model without sharing raw data, while protecting against privacy leaks and malicious attacks.
- **Challenges**: Model poisoning, data leakage, adversarial updates, and communication threats.

2. System Initialization

- Define the FL architecture (centralized or decentralized).
- Register clients with authenticated identities.
- Distribute an initial global model to all clients.

3. Local Training at Client Devices

- Each client uses its **private local dataset** to train a copy of the global model.
- **Model Architecture**: CNNs for image tasks, LSTM for sequence tasks, etc.
- **Training Process**:
  - Perform several epochs of training.
  - Record gradients or weight updates.
  - Apply **differential privacy (DP)** or **local encryption** if enabled.

4. Privacy-Preserving Techniques

- **Differential Privacy (DP)**: Add calibrated noise to model updates before sharing.
- **Homomorphic Encryption**: Encrypt model parameters to ensure computation on encrypted data.
- **Secure Multi-Party Computation (SMC)**: Enables secure aggregation without exposing individual updates.

5. Poisoning & Attack Detection

- Detect and mitigate malicious updates using AI-driven anomaly detection:
  - **Statistical Filtering**: Detect outlier updates (e.g., cosine similarity).
  - **Autoencoders or Isolation Forests**: Spot suspicious parameter deviations.

6. Robust Aggregation

- Aggregate filtered updates from trusted clients using secure protocols:
  - **FedAvg** (standard averaging)
  - **Krum**, **Trimmed Mean**, or **Median**: Handle adversarial or noisy updates
  - **AI-Driven Aggregators**: Dynamically learn weights for reliable updates

7. Global Model Update & Redistribution

- Server updates the global model using the secure aggregation results.
- Distributes the updated model back to participating clients.
- Verifies model integrity and versioning before redistribution.

8. Communication Security

- Encrypt all communication channels using TLS/SSL**.**
- Optionally use blockchain or digital signatures to log and verify update transactions.

9. Continuous Monitoring & Adaptation

- Monitor metrics such as:
  - Model accuracy, convergence rate
  - Client participation logs
  - Anomaly scores of client updates
- Retrain or adjust thresholds and defense mechanisms over time.

10. Evaluation & Validation

- **Security Evaluation**:

  - Test against known attacks (e.g., model poisoning, backdoors).
  - Analyze robustness under various adversarial settings.
- **Performance Metrics**:
  - **Accuracy**, **Loss**, **F1-score** of the global model
  - Communication Overhead

IV.RESULTS AND DISCUSSION

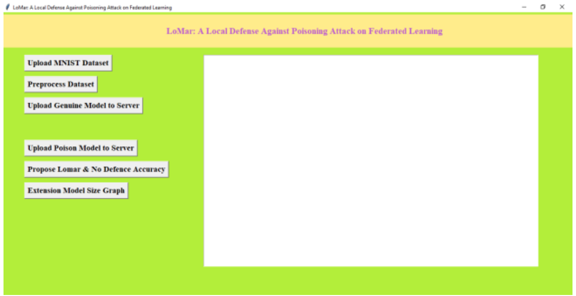
LoMar: A Local Defense Against Poisoning Attack on Federated Learning
LoMar: A Local Defense Against Poisoning Attack on Federated Learning
Upload MNIST Dataset
Preprocess Dataset
Upload Genuine Model to Server
Upload Poison Model to Server
Propose Lomar & No Defence Accuracy
Extension Model Size Graph

Fig 1

The graphical user interface (GUI) shown in the image is part of a system designed to demonstrate and evaluate LaMar**,** a local defense mechanism against poisoning attacks in federated learning**.** The application allows users to interact with key components of a federated learning workflow using the MNIST dataset. The interface features clearly labeled buttons on a bright green background, each representing a step in the pipeline. These include options to upload the MNIST dataset**,** preprocess the data**, and** upload both genuine and poisoned models to the server. Additional functionality includes a button to **evaluate model accuracy** under both LaMardefense and no defense, as well as a feature to **visualize the model size through graphs**. A large white panel on the right is likely reserved for visual output such as accuracy plots, training logs, or model comparisons. At the top, a title in orange and purple text reinforces the system's focus: "LaMar: A Local DefenseAgainst Poisoning Attack on Federated Learning".

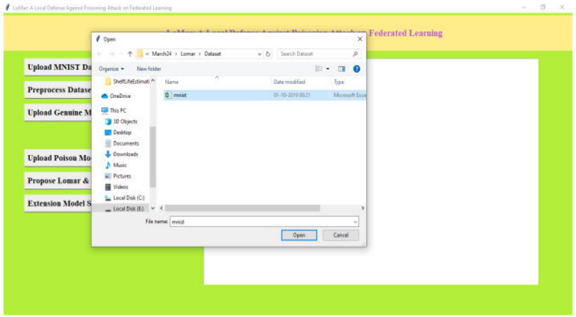

Fig 2

The image displays a snapshot of a user interacting with the LaMar GUIa tool developed for defending against poisoning attacks in federated learning environments. The interface is active, and a file upload dialog box is currently open, suggesting the user is in the process of uploading a dataset**.**The directory shown in the above Figure is "C:\Users\...\Lamar\Dataset", and the file selected is named "mnist"**,** likely it representing the popular MNIST dataset of handwritten digits used in machine learning and computer vision tasks.

On the left side of the GUI, we still see the series of buttons representing key steps in the federated learning pipeline:

- Uploading the MNIST dataset,
- Preprocessing it,
- Uploading both genuine and poisoned models,
- Testing accuracy with and without defense,
- And visualizing model performance via graphs.

This action suggests the initial phase of the workflow, where the user begins by loading the necessary dataset. The large white display panel on the right is still visible, reserved for outputs such as graphs or logs once further steps are executed. Overall, this screenshot captures the **data-loading phase**, essential for beginning experiments with the LaMardefense framework.

V. FUTURE SCOPE AND CONCLUSION

LoMar offers a comprehensive answer to the significant issue of poisoning assaults in federated learning. LoMar implements a client-level detection method that proficiently identifies and mitigates harmful behaviors prior to their impact on the global model. This solution mitigates a critical vulnerability in federated learning systems, as conventional techniques frequently neglect the possibility of localized attacks that can spread and diminish overall model efficacy. LoMar's novel design improves the security and dependability of federated learning, guaranteeing that collaborative training procedures are trustworthy and successful.

The lightweight and efficient characteristics of LoMar guarantee that its implementation does not place excessive computing demands on individual clients. This attribute is vital for the effective implementation of LoMar in actual federated learning settings, where sustaining system performance and scalability is crucial. LoMar utilizes sophisticated statistical analysis and machine learning methodologies to offer an extensive and refined detection capability that differentiates between normal changes and harmful activities. This accuracy in detection aids in preserving the integrity of the global model without compromising performance. Furthermore, the scalability of LoMar's design renders it appropriate for various federated learning applications, ranging from small-scale implementations to large networks with numerous clients. Its modular architecture enables effortless incorporation into current federated learning systems, promoting its implementation across several sectors. As federated learning gains prominence, the necessity for effective security mechanisms such as LoMar becomes increasingly imperative. LoMar effectively meets this requirement by offering a scalable and efficient solution, hence assuring the safe and widespread implementation of federated learning.

In conclusion, LoMar signifies a substantial progression in the domain of federated learning security. The local defense mechanism offers essential protection against poisoning attacks, hence augmenting the system's robustness and reliability. By emphasizing lightweight, efficient, and scalable solutions, LoMar guarantees that federated learning maintains its advantages of privacy and decentralization without sacrificing security. As federated learning advances, LoMar emerges as an essential instrument for preserving the integrity of collaborative machine learning initiatives, enhancing confidence and reliability in this novel method of data-driven insights.

REFERENCES

1. Abadi, M., Chu, A., Goodfellow, I., et al. (2016). Deep Learning with Differential Privacy. Proceedings of the 2016 ACM SIGSAC Conference

on Computer and Communications Security, 308-318.

2. Aono, Y., Hayashi, T., Wang, L., &Moriai, S. (2017). Privacy-preserving deep learning via additively homomorphic encryption. IEEE Transactions on Information Forensics and Security, 13(5), 1333-1345.

3. Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., &Shmatikov, V. (2020). How to backdoor federated learning. Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics, 2938-2948.

4. Bhagoji, A. N., Chakraborty, S., Mittal, P., &Calo, S. (2019). Analyzing federated learning through an adversarial lens. International Conference on Machine Learning, 634-643.

5. Bonawitz, K., Ivanov, V., Kreuter, B., et al. (2017). Practical secure aggregation for federated learning on user-held data. Advances in Neural Information Processing Systems, 30, 61-71.

6. Cappello, F., Snir, M., Hoefler, T., Gropp, W., & Beckman, P. (2020). Toward exascale resilience: 2014 update. Supercomputing Frontiers and Innovations, 1(1), 51-64.

7. Chen, Y., Sun, X., & Jin, Y. (2018). Communication-efficient federated deep learning with layer wise asynchronous model update and temporally weighted aggregation. IEEE Transactions on Neural Networks and Learning Systems, 31(10), 4229-4238.

8. Fang, M., Cao, X., Jia, J., & Gong, N. (2020). Local model poisoning attacks to Byzantine-robust federated learning. Proceedings of the 29th USENIX Security Symposium, 1605-1622.

9. Fung, C., Yoon, C. J. M., &Beschastnikh, I. (2018). Mitigating Sybils in federated learning poisoning. Proceedings of the 22nd International Symposium on Research in Attacks, Intrusions, and Defenses, 297-313.

10. Geyer, R. C., Klein, T., & Nabi, M. (2017). Differentially private federated learning: A client-level perspective. NIPS Workshop on Privacy in Machine Learning and Artificial Intelligence.

11. Hard, A., Rao, K., Mathews, R., et al. (2018). Federated learning for mobile keyboard prediction. arXiv preprint arXiv:1811.03604.

12. Hitaj, B., Ateniese, G., & Perez-Cruz, F. (2017). Deep models under the GAN: Information leakage from collaborative deep learning. Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 603-618.

13. Kairouz, P., McMahan, H. B., et al. (2019). Advances and open problems

in federated learning. arXiv preprint arXiv:1912.04977.

14. Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S., & Suresh, A. T. (2020). SCAFFOLD: Stochastic controlled averaging for federated learning. Proceedings of the 37th International Conference on Machine Learning, 5132-5143.

15. Li, T., Sahu, A. K., Talwalkar, A., & Smith, V. (2020). Federated learning: Challenges, methods, and future directions. IEEE Signal Processing Magazine, 37(3), 50-60.

16. Lyu, L., Yu, H., & Yang, Q. (2020). Threats to federated learning: A survey.arXiv,preprintar,Xiv:2003.02133.

17. McMahan, H. B., Moore, E., Ramage, D., Hampson, S., &Arcas, B. A. Y. (2017). Communication-efficient learning of deep networks from decentralized data. Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, 1273-1282.

18. Melis, L., Song, C., De Cristofaro, E., &Shmatikov, V. (2019). Exploiting unintended feature leakage in collaborative learning. 2019 IEEE Symposium on Security and Privacy (SP), 691-706.

19. Mo, Y., Ding, S., Ma, T., & Liu, Y. (2020). Detecting adversarial attacks in federated learning through dynamic model behavior analysis. Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, 256-272.

20. Sato, H., Takeuchi, I., et al. (2019). Bagging by design: Hands-off aggregation and robust training for federated learning. arXiv preprint arXiv:1904.04196.

21. Sharma, R., Gupta, S., &Goyal, S. (2020). Secure and efficient federated learning using homomorphic encryption. International Journal of Computer Applications, 175(19), 1-7.

22. Shejwalkar, V., Houmansadr, A. (2021). Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. Proceedings of the Network and Distributed System Security Symposium (NDSS).

23. Sun, X., Wang, Y., & Liu, Z. (2021). Federated learning with adversaries: Taxonomy, threats, and defenses. IEEE Internet of Things Journal, 8(6), 4375-4392.

24. Wang, S., Tuor, T., Salonidis, T., et al. (2019). Adaptive federated learning in resource-constrained edge computing systems. IEEE Journal on Selected Areas in Communications, 37(6), 1205-1221.

AUTHORS PROFILE

Mr. SK. HIMAM BASHA is an Assistant Professor in the Department of Master of Computer Applications at QIS College of Engineering&Technology, Ongole,Andhra

Pradesh. He earned his Master of Computer Applications MCA from Anna University, Chennai. With a strong research background, He has authored and co-authored research papers published in reputed peer-reviewed journals. His research interests include ML, AI,CloudComputing,andProgrammingLanguages. He is committed to advancing research and fostering innovation while mentoring students to excel in both academic and professional pursuits.

Mr.M. NAGESWARA RAO is a postgraduate student pursuing a MCA in the Department of Computer Applications at QIS College of Engineering & Technology, Ongole an Autonomous college in Prakasam dist. He completed his undergraduate degree in B.Com(Computers)from(Acharya Nagarjuna University).

His academic interests include Cloud Computing, Artificial Intelligence, Cyber security and Data structures.