International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Privacy-Preserving AI Model for IoT Networks Utilizing Differential Privacy and Secure Aggregation in Decentralized Learning Environments

G.Dhanalakshmi[1*], A.Balamanikandan[2], S.G.Rahul [3], T.Vithyaa[4], A.N. Arularasan[5], A. Ishwariya[6]

Submitted: 19/01/2024 **Revised**: 28/02/2024 **Accepted**: 05/03/2024

Abstract: The level of privacy protection needed for IoT depends on various factors and considerations. Factors such as the sensitivity of the data at stake, compliance with stringent data protection regulations, alignment with user privacy expectations, and the potential for misuse underscore the need for a nuanced approach. Additionally, the security integrity of the entire IoT ecosystem, encompassing device security, network fortification, and data storage, significantly influences the imperative for robust privacy safeguards. The work introduces a pioneering decentralized Privacy-Preserving AI system tailored for the Internet of Things (IoT), emphasizing user privacy while harnessing the collaborative power of decentralized learning. The system integrates two key algorithms, K-Means clustering and Random Forest (RF), each fortified with robust privacy-preserving mechanisms. The K-Means algorithm strategically applies Differential Privacy, introducing controlled noise during clustering to protect sensitive individual data points. Concurrently, the Random Forest algorithm employs ensemble learning, allowing each IoT device to contribute to local decision tree creation with the addition of Laplace noise for safeguarding sensitive data. The decentralized model ensures secure, peer-to-peer collaboration during updates, facilitating the creation of a Global AI Model that embodies collective knowledge. The workflow involves local data processing, privacy-preserving algorithms, secure collaborative updates, and model distribution. The proposed approach strikes a delicate balance between utility and data protection, offering a powerful and privacy-respecting solution for AI models in decentralized IoT environments. The system's architecture, algorithms, and workflow, emphasizing its significance in ensuring robust privacy while advancing AI capabilities in IoT applications.

Keywords: *Internet of things, K-Means clustering, Random Forest, decentralized Model, Privacy Preserving AI*

1. Introduction

The essential need for privacy-preserving AI within the realm of the Internet of Things (IoT) arises from the pervasive presence of interconnected smart devices and sensors in this modern world [1]. These devices accumulate vast amounts of sensitive data, ranging from personal health metrics to daily habits. The critical need to safeguard this information arises not only from the potential for unauthorized access but also due to the security vulnerabilities inherent in IoT ecosystems [2]. By integrating privacy-preserving AI techniques, organizations can mitigate these risks, ensuring that sensitive data remains secure even in the face of evolving threats. Moreover, user trust is paramount for the widespread acceptance of IoT technologies, and the implementation of robust privacy measures serves to reassure individuals that their data is handled responsibly. Additionally, as IoT devices operate across diverse domains, including healthcare and smart homes, privacy-preserving AI facilitates seamless integration while respecting the unique privacy requirements of each sector [3]. Ultimately, these measures empower users with greater control over their data, fostering a sense of ownership and control in an increasingly interconnected and data-driven world.

Differential Privacy and Secure Aggregation play pivotal roles in addressing privacy concerns in decentralized learning environments, particularly in the context of the Internet of Things (IoT) [4]. Differential Privacy focuses on safeguarding individual data points within a dataset, ensuring that the inclusion or exclusion of any single data entry does not significantly impact the overall outcome of a machine learning model. This is crucial in decentralized settings where data from multiple sources, such as IoT devices, is utilized for collaborative learning without compromising the privacy of individual contributors. Secure Aggregation, on the other hand, involves the secure combination of model updates from multiple decentralized devices without exposing the individual contributions [5]. It prevents malicious actors from extracting sensitive information during the aggregation process. Both of these techniques are instrumental in maintaining data privacy and security in decentralized learning

[1]*Department of Electronics and Communication Engineering, AVN Institute of Engineering and Technology, Hyderabad, Telangana-501510, India. Email: drdhanabhavesh07@gmail.com**

[2]*Department of Electronics and communication engineering Mohan Babu University (Erstwhile Sree Vidyanikethan Engineering College), Tirupati, India balamanieee83@gmail.com*

[3] *Department of Electronics and Communication Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai 600062, Tamil Nadu, India, Email: rahulgopi1993@yahoo.com*

[4] *Department of Computer Science and Business System, PSNA College of Engineering and Technology,Dindigul-624622,Tamil Nadu, India Email: vithu2511@gmail.com*

[5] *Department of Artificial Intelligence and Data Science, Panimalar Engineering College, Chennai - 600123, Tamil Nadu, India. Email: arularasan@live.com*

[6] *Department of Computer Science and Engineering, P.S.R Engineering College, Sivakasi, Tamil Nadu 626140, Email: ishwariya8888@gmail. com*

environments [6]. In the realm of IoT, where numerous devices generate and process data locally, implementing these privacy-preserving techniques becomes paramount. By applying Differential Privacy, IoT devices can contribute to collaborative learning models without revealing sensitive information about their users [7]. Secure Aggregation ensures that the collaborative model is built without exposing the specifics of each device's data. For instance, in a scenario where multiple IoT devices are involved in health monitoring, these privacy-preserving techniques would allow for the collaborative training of a predictive model without compromising the privacy of individual health records [8]. This facilitates the development of more robust and accurate models while upholding the principles of privacy and security. As IoT continues to expand, the integration of such privacy-preserving mechanisms becomes essential to encourage data sharing and collaborative learning without sacrificing individual privacy. The objectives of the work are:

- Develop a secure K-Means algorithm with Differential Privacy for grouping IoT data[17], ensuring pattern recognition without compromising individual data privacy.
- Integrate Random Forest, using Laplace noise for secure decision tree creation on each IoT device, enhancing data protection during collective predictions.
- Enable confidential, peer-to-peer collaboration among IoT devices during model updates, ensuring aggregated information sharing without revealing individual details.
- Establish a workflow for creating a Global AI Model, encompassing local data processing, privacy-preserving algorithms, secure updates, and model distribution, balancing utility and data protection in IoT applications.

2. Literature Review

Data encryption is a standard practice for securing data during transmission and storage by converting it into a coded format that can only be deciphered with the appropriate encryption key [9]. While encryption is a fundamental component of information security, its application in privacy-preserving techniques for IoT has limitations. In IoT scenarios, the distribution and management of encryption keys become critical factors, and if these keys are compromised, the encrypted data can be vulnerable. Moreover, encrypted data may still be susceptible to attacks on the encrypted form, such as differential analysis or side-channel attacks. Thus, while encryption remains a crucial aspect of data security, it is most effective when complemented by additional privacy-preserving mechanisms, especially in the complex and dynamic ecosystems characteristic of the Internet of Things [10].

Data masking, or obfuscation, is a privacy-preserving technique that involves altering sensitive information within datasets by substituting or scrambling actual values with fake or randomized ones. This method is employed to protect sensitive information from unauthorized access during data handling or transmission [11]. While data masking is a relatively straightforward approach, it has limitations. Simple masking may not be sufficient to thwart determined attackers or sophisticated algorithms designed to reverse-engineer the masked data. In scenarios where the underlying structure of the data is discernible, attackers may employ statistical methods to uncover patterns and reveal the original information. Therefore, while data masking provides a basic level of protection, it may not be suitable for securing data against more advanced privacy threats in dynamic and interconnected environments, such as those found in the Internet of Things [12][19].

Data perturbation is a privacy-preserving technique that involves introducing random noise or small alterations. This method is often used to anonymize or obscure specific details in sensitive data [13]. The idea is to add enough noise to make it challenging for attackers to discern individual information while ensuring that the perturbed data remains useful for analytical purposes [14]. However, achieving the right balance is a delicate task, as excessive perturbation can compromise the quality and accuracy of the data, rendering it less valuable for meaningful analysis. Additionally, while data perturbation can provide a level of privacy protection, it may not be resilient against advanced attacks, such as those leveraging machine learning algorithms to infer patterns in the perturbed data [15]. Therefore, while data perturbation offers a basic level of privacy enhancement, it may not be the most robust solution for preserving privacy, especially in the face of evolving and sophisticated privacy threats in IoT environments [16][[18].

2.1 Proposed Work

The planned Privacy-Preserving AI system for IoT aims to give top priority to user privacy. It achieves this by using decentralized learning, which means the learning process happens on individual devices, emphasizing a more distributed and private approach. At its core, the system integrates two pivotal algorithms—K-Means clustering and Random Forest (RF)—both fortified with robust privacy-preserving mechanisms. The K-Means clustering algorithm plays a central role in grouping IoT device data into distinct clusters, facilitating pattern recognition and relationship understanding without compromising individual data points. Differential Privacy is strategically applied to this algorithm, introducing controlled noise during clustering to ensure the confidentiality of sensitive information. Complementing this, the Random Forest algorithm, a powerful ensemble learning approach, utilizes decision trees for collective predictions.

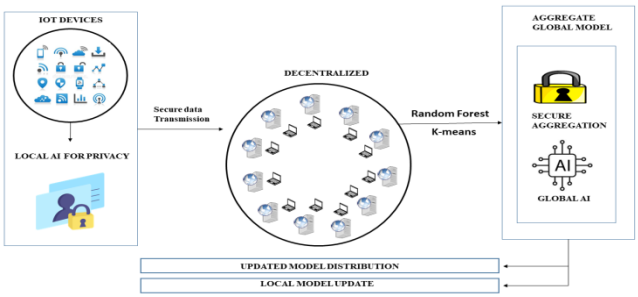

Fig.1 Decentralized Learning for IoT Privacy-Preserving AI

In this decentralized system, each IoT device contributes to local decision tree creation, safeguarding sensitive data through the addition of Laplace noise to leaf labels and split thresholds. This decentralized approach enables secure, peer-to-peer collaboration among IoT devices during model updates, ensuring that aggregated information is shared directly without revealing individual details. The collaborative learning process results in the creation of a Global AI Model that embodies the collective knowledge of all decentralized devices. The overall workflow involves local data processing, privacy-preserving algorithms, secure collaborative updates, and model distribution. This pioneering approach ensures that AI models in IoT applications

are not only powerful but also respectful of user privacy, striking a delicate balance between utility and data protection in a decentralized learning environment.

Fig.1 illustrates the flow of data and operations within the decentralized Privacy-Preserving AI system for IoT. It starts with data generation by IoT devices, followed by local processing with K-Means clustering applying Differential Privacy. The decentralized model then undergoes Secure Aggregation and Model Distribution, resulting in the creation of a Global AI Model. Random Forest, with Laplace Noise, is applied locally, and secure collaborative model updates contribute to an updated model for individual devices. The decentralized architecture ensures collective knowledge without compromising privacy.

2.2 Algorithmic Framework

The algorithm, named Privacy-Preserving K-Means with Secure Aggregation, takes an encrypted dataset as input and conducts K-means clustering while ensuring privacy through differential privacy with Laplace noise addition. The parameters include the number of clusters 'K', the privacy parameter '$\boldsymbol{\varepsilon}$', the maximum number of iterations 'max_iterations', and 'sigma' controlling the smoothness of weights in local centroid updates. The algorithm outputs the final cluster centers ('centroids') and the assignment of each data point to a cluster ('assignments'). It employs secure aggregation to protect sensitive information during the aggregation process, making it suitable for decentralized learning environments with privacy concerns.

Algorithm 1: Privacy-Preserving K-Means with Secure Aggregation

```
def privacy_preserving_k_means(D, K,ε, max_iterations, σ):
# Initialization
    centroids = np.random.choice(D, K, replace=False)
    for t in range(max_iterations):
# Local Data Processing
    perturbed_distances = local_data_processing(D, centroids, ε)
# Local Centroid Update
    local_centroids = local_centroid_update(D, perturbed_distances, σ)
# Secure Aggregation
    aggregated_centroids = secure_aggregation(local_centroids)
# Global Centroid Update
    centroids = global_centroid_update(aggregated_centroids)
# Final Clustering Result
    assignments = assign_data_to_clusters(D, centroids)
    return centroids, assignments
def initialize_centroids(D, K):
# Randomly initialize centroids
indices = np.random.choice(len(D), K, replace=False)
return D[indices]
def local_data_processing(D, centroids,ε):
# Perturb distances for differential privacy
perturbed_distances = [cdist(d_i, centroids) + (scale = 2 × len(d_i)/ε) for d_i in D]
return np.array(perturbed_distances)
def local_centroid_update(D, perturbed_distances, σ):
# Update local centroids based on perturbed distances
local_centroids = [np.sum(softmax(−pd[:,k],σ)[:,np.newaxis]*d_i,axis =0) / i·np·sum(softmax(−pd[:,k],σ))for k, pd_i in enumerate(perturbed_distances) for d_i in D]
return np.array(local_centroids)
def secure_aggregation(local_centroids):
# Securely aggregate local centroids
aggregated_centroids = np.sum(local_centroids, axis=0)
return aggregated_centroids
def global_centroid_update(aggregated_centroids):
# Update global centroids based on aggregated values
return aggregated centroids / len(aggregated centroids)
def assign_data_to_clusters(D, centroids):
# Assign each data point to the nearest global centroid
distances = cdist(D.reshape(-1, D.shape[-1]), centroids)
assignments = np.argmin(distances, axis=1)
return assignments
def softmax(x,σ):
# Softmax function
exp_x = np.exp(x/σ)
return exp_x / np.sum(exp axis=0))
# Example usage
D = np.random.rand(100, 2)  # Replace with your dataset
K = 3
ε= 1.0
max_iterations = 10
σ= 1.0
# Run the algorithm
centroids, assignments = privacy_preserving_k_means(D, K, ε, max_iterations,σ)
# Display final results
print("Final Centroids:", centroids)
print("Final Assignments:", assignments)
```

This algorithm takes an encrypted dataset and corresponding binary labels as input in a decentralized learning environment. It leverages a privacy-preserving Random Forest methodology, training individual decision trees with differential privacy through the addition of Laplace noise to leaf labels and split thresholds. The algorithm outputs a privacy-preserving Random Forest ('forest') composed of multiple differentially private decision trees. For predictions on new data, the 'privacy_preserving_ random_forest_predict' function is employed, which aggregates predictions from each tree using the median to ensure privacy. The resulting 'predictions' represent the model's predictions for the provided data. This algorithm is designed to address privacy concerns in decentralized learning scenarios while maintaining the utility of Random Forest models.

Algorithm 2: Privacy-Preserving RF in Decentralized Environments

```
def privacy_preserving_random_forest(D, labels, n_trees, ε, max_depth):
    forest = []
    for _ in range(n_trees):
# Sample with replacement for each tree
    D_bootstrap, labels_bootstrap = resample(D, labels, replace=True)
# Train a differentially private decision tree on the bootstrapped sample
    tree = privacy_preserving_decision_tree(D_bootstrap, labels_bootstrap, ε, max_depth)
# Add the tree to the forest
    forest.append(tree)
```

```
7.  return forest
8.  def privacy_preserving_decision_tree(D, labels, ε,
max_depth):
9.  tree = DecisionTreeClassifier(max_depth=max_depth)
10. for node in tree.tree_:
# Each node in the decision tree
11. if tree.tree_.children_left[node] ==
tree.tree_.children_right[node]:  # Leaf node
# Noisy label for the leaf
12. laplace_noise = laplace(scale=1/ε)
13. leaf_label = np.random.choice(np.unique(labels), 1)[0]
14. tree.tree_.value[node] = [[leaf_label + laplace_noise]]
else:  # Non-leaf node
# Noisy threshold for the split
15. laplace_noise = laplace(scale=1/ε)
16. split_threshold = tree.tree_.threshold[node] + laplace_noise
17. tree.tree_.threshold[node] = split_threshold
18. return tree
19. def privacy_preserving_random_forest_predict(forest, X):
# Make predictions for each tree and aggregate results
20. predictions = [tree.predict(X) for tree in forest]
21. aggregated_predictions = np.median(np.array(predictions),
axis=0)
22. return aggregated_predictions.astype(int)
# Example usage
23. D = np.random.rand(100, 10)  # Replace with your dataset
24. labels = np.random.randint(0, 2, 100)  # Replace with your
labels (binary classification)
25. n_trees = 5
26. ε= 1.0
27. max_depth = 3
# Train a privacy-preserving random forest
28. forest = privacy_preserving_random_forest(D, labels,
n_trees, ε, max_depth)
# Make predictions on new data
29. new_data = np.random.rand(10, 10)  # Replace with your
new data
30. predictions =
privacy_preserving_random_forest_predict(forest, new_data)
31. print("Predictions:", predictions)
```

2.3 Mathematical Indication

The Privacy-Preserving K-Means algorithm commences with the random selection of initial centroids.

$$centroids = np.random.choice(D, K, replace = False) \quad (1)$$

where K represents the number of clusters. The subsequent step perturbs distances for differential privacy during local data processing. Each data point's distance to the centroids is perturbed using Laplace noise, denoted by:

$$laplace(scale = \frac{2 \cdot len(d_i)}{\epsilon}) \quad (2)$$

Where ϵ governs the privacy level. Following the perturbation, local centroids are updated based on the perturbed distances. The update involves a weighted average of data points, employing the softmax function. This process is captured by the equation, with σ controlling the smoothness of the softmax. The subsequent step involves securely aggregating local centroids. The equation is:

$$aggregated\_centroids = np \cdot argmin(distances, axis = 1) \quad (3)$$

This equation captures the aggregation, consolidating information from all local computations in a secure manner. Global centroids are then updated by dividing the aggregated centroids by their total count, ensuring a global representation of the clusters. Finally, data points are assigned to the nearest global centroids based on Euclidean distances. The equation is:

$$assignments = np.argmin(distances, axis = 1) \quad (4)$$

Additionally, the algorithm employs a softmax function to normalize values, converting them into probabilities. The softmax equation is:

$$softmax(x, \sigma) = \frac{exp(x/\sigma)}{\sum exp(x/\sigma)} \quad (5)$$

This equation elucidates the normalization, with σ influencing the smoothness of the softmax. Overall, these mathematical formulations underpin the Privacy-Preserving K-Means algorithm, delineating the steps taken to achieve privacy-preserving clustering.

3. Result

The interplay between privacy and accuracy within a decentralized learning framework implementing Differential Privacy and Secure Aggregation for K-Means clustering. The $\boldsymbol{\varepsilon}$ parameter, denoting the privacy level, is systematically varied in a simulated environment to elucidate its impact on both model accuracy and privacy loss. While this script provides an initial perspective, a robust security analysis necessitates a holistic consideration of additional dimensions, including resilience against adversarial attacks, communication overhead, and real-world experimentation. The applicability and efficacy of Differential Privacy and Secure Aggregation must be evaluated in the specific context of the decentralized learning environment, factoring in distinctive goals, threat models, and application prerequisites. Only through this nuanced assessment can one discern the most suitable equilibrium between privacy preservation and model performance.

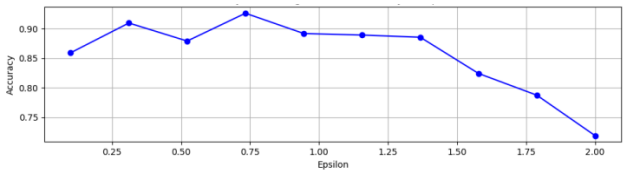

Fig.2 K-means Accuracy vs. ε

The Fig.2 displays how the accuracy of the K-Means clustering model changes with different values of $\boldsymbol{\varepsilon}$. The $\boldsymbol{\varepsilon}$ is a parameter controlling privacy in the model. Each point on the blue line represents a different setting of $\boldsymbol{\varepsilon}$, ranging from 0.1 to 2.0. The y-axis (vertical) shows the accuracy of the model, ranging from 0.7 to 0.95. Observing this graph helps us understand the relationship

between privacy settings (***ε***) and the accuracy of the K-Means clustering algorithm. A higher accuracy value indicates better performance, while changes in epsilon highlight the trade-off between privacy and model accuracy.

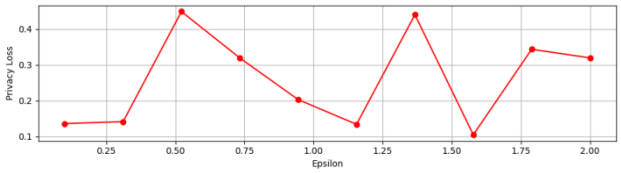

Fig.3 K-means Privacy Loss vs. ε

The Fig.3 illustrates the trade-off between privacy and ε. Here, the red line represents how much privacy is lost at different epsilon values. Privacy loss is a measure of how much individual data points may be revealed during the learning process. As ε increases, the privacy loss may decrease or increase, and this graph helps visualize the impact. Lower privacy loss values indicate better privacy preservation. Understanding this trade-off is crucial for finding an optimal balance between privacy and model utility in privacy-preserving K-Means clustering.

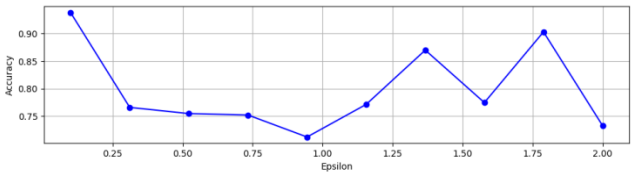

Fig. 4 Random Forest - Accuracy vs. ε

The Fig.4 illustrates the relationship between the privacy parameter (ε) and the accuracy of a Privacy-Preserving Random Forest model. Each point on the blue line represents a different value of ***ε***, ranging from 0.1 to 2.0. The y-axis shows the accuracy of the model, ranging from 0.7 to 0.95. The graph provides insights into how changes in privacy settings, represented by ε, impact the accuracy of the Random Forest. Higher accuracy values indicate better performance, while different ε values showcase the trade-off between privacy and model accuracy.

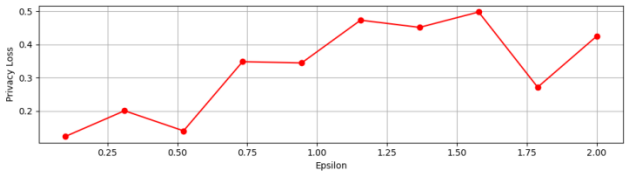

Fig.5 Random Forest - Privacy Loss vs. ε

The Fig.5 demonstrates the trade-off between privacy and ***ε*** in the Privacy-Preserving Random Forest. The red line represents the privacy loss, indicating how much privacy is compromised at various ε values. Privacy loss is a measure of potential information leakage during the learning process. Lower values on the y-axis signify better preservation of privacy. This graph offers a visual understanding of how adjusting the privacy parameter affects the trade-off between the model's privacy preservation and its utility in making accurate predictions.

4. Conclusion and Future Work

The decentralized Privacy-Preserving AI system for IoT seamlessly blends advanced algorithms with robust privacy mechanisms, ensuring a powerful and secure collaborative learning environment. By prioritizing user privacy through techniques like Differential Privacy and secure aggregation, the system strikes a delicate balance between utility and data protection. The creation of a Global AI Model reflects the success of the decentralized approach, embodying collective knowledge while preserving individual device privacy. Future endeavors will focus on refining privacy-preserving techniques, optimizing model distribution mechanisms, and exploring scalability for diverse IoT environments. Additionally, ongoing efforts will address emerging privacy challenges and further enhance the system's adaptability to evolving IoT landscapes.

References

[1] Zaman, U., Imran, Mehmood, F., Iqbal, N., Kim, J., & Ibrahim, M. (2022). Towards Secure and Intelligent Internet of Health Things: A Survey of Enabling Technologies and Applications. *Electronics*, *11*(12), 1893.

[2] Srivastava, A., Gupta, S., Quamara, M., Chaudhary, P., & Aski, V. J. (2020). Future IoT-enabled threats and vulnerabilities: State of the art, challenges, and future prospects. *International Journal of Communication Systems*, *33*(12), e4443.

[3] Deebak, B. D., Memon, F. H., Cheng, X., Dev, K., Hu, J., Khowaja, S. A., ... & Choi, K. H. (2022). Seamless privacy-preservation and authentication framework for IoT-enabled smart eHealth systems. *Sustainable Cities and Society*, *80*, 103661.

[4] Rivadeneira, J. E., Silva, J. S., Colomo-Palacios, R., Rodrigues, A., & Boavida, F. (2023). User-centric privacy preserving models for a new era of the Internet of Things. *Journal of Network and Computer Applications*, 103695.

[5] Tsaloli, G., Liang, B., Brunetta, C., Banegas, G., & Mitrokotsa, A. (2021, November). DEVA: Decentralized, verifiable secure aggregation for privacy-preserving learning. In *International Conference on Information Security* (pp. 296-319). Cham: Springer International Publishing.

[6] Shafeeq, S., Alam, M., & Khan, A. (2019). Privacy aware decentralized access control system. *Future Generation Computer Systems*, *101*, 420-433.

[7] Jiang, B., Li, J., Yue, G., & Song, H. (2021). Differential privacy for industrial internet of things: Opportunities, applications, and challenges. *IEEE Internet of Things Journal*, *8*(13), 10430-10451.

[8] Kasyap, H., & Tripathy, S. (2021). Privacy-preserving decentralized learning framework for healthcare system. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, *17*(2s), 1-24.

[9] Abiodun, M. K., Awotunde, J. B., Ogundokun, R. O., Adeniyi, E. A., & Arowolo, M. O. (2021). Security and information assurance for IoT-based big data. In *Artificial Intelligence for Cyber Security: Methods, Issues and Possible Horizons or Opportunities* (pp. 189-211). Cham: Springer International Publishing.

[10] Satyanarayana, P., Diwakar, G., Subbayamma, B. V., Phani Sai Kumar, N. V., Arun, M., & Gopalakrishnan, S. (2023). Comparative analysis of new meta-heuristic-variants for privacy preservation in wireless mobile adhoc networks for IoT applications. Computer Communications, 198, 262–281. https://doi.org/10.1016/j.comcom. 2022.12.006.

[11] Yamac, M., Ahishali, M., Passalis, N., Raitoharju, J., Sankur, B., & Gabbouj, M. (2020). Multi-level reversible data anonymization via compressive sensing and data hiding. *IEEE Transactions on Information Forensics and Security*, *16*, 1014-1028.

[12] Grammatikis, P. I. R., Sarigiannidis, P. G., & Moscholios, I. D. (2019). Securing the Internet of Things: Challenges, threats and solutions. *Internet of Things*, *5*, 41-70.

[13] Rai, R. K. (2022). Data Mining Of Perturbation Data Approach Using Different Noise For Privacy Preserving Techniques. *Neuro Quantology*, *20*(9), 4039.

[14] Keshk, M., Moustafa, N., Sitnikova, E., Turnbull, B., & Vatsalan, D. (2020, December). Privacy-preserving techniques for protecting large-scale data of cyber-physical systems. In *2020 16th international conference on mobility, sensing and networking (MSN)* (pp. 711-717). IEEE.

[15] Arachchige, P. C. M. (2020). *Scalable data perturbation for privacy preserving large scale data analytics and machine learning* (Doctoral dissertation, RMIT University).

[16] Perumal, G., Subburayalu, G., Abbas, Q., Naqi, S. M., & Qureshi, I. (2023). VBQ-Net: A Novel Vectorization-Based Boost Quantized Network Model for Maximizing the Security Level of IoT System to Prevent Intrusions. Systems, 11(8), 436. MDPI AG.

[17] Rachapudi V., Venkata Suryanarayana S., Subha Mastan Rao T. "Auto-encoder based K-means clustering algorithm". International Journal of Innovative Technology and Exploring Engineering. 2019. 8(5), pp. 1223-1226

[18] Kallam, Suresh, A. Veerender, K. Shilpa, K. Ranjith Reddy, K. Reddy Madhavi, and Jonnadula Narasimharao. "The Adaptive Strategies Improving Design in Internet of Things." In *Proceedings of Third International Conference on Advances in Computer Engineering and Communication Systems: ICACECS 2022*, pp. 691-699. Singapore: Springer Nature Singapore, 2023.

[19] Kumar Apat, S., Mishra, J., Srujan Raju, K., Padhy, N. (2023). IoT-Assisted Crop Monitoring Using Machine Learning Algorithms for Smart Farming. In: Kumar, R., Pattnaik, P.K., R. S. Tavares, J.M. (eds) Next Generation of Internet of Things. Lecture Notes in Networks and Systems, vol 445. Springer, Singapore. https://doi.org/10.1007/978-981-19-1412-6_1