iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** X **Month of publication:** October 2022

DOI: https://doi.org/10.22214/ijraset.2022.47120

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue X Oct 2022- Available at www.ijraset.com

A Machine Learning Model to Protect Privacy Using Federal Learning with Homomorphy Encryption

Rejuwan Shamim[1], Md. Arshad[2], Dr. Vinay Pandey[3]
[1, 2]*Student of CSE, Data Science, Maharishi University of Information Technology, Noida, India*
[3]*Assistant Professor, dept. of CSE Maharishi University of Information Technology, Noida, India*

Abstract: ***Machine learning technologies have a marvelous success in emancipating real-world Artificial Intelligence applications. But still, vast numbers of sensitive data are produced every second time in different forms. This data can be in the form of health records, shopping records, internet searching records, mobile and laptop activities, and so on. This data can be used to train our Machine learning /Deep learning models to make Artificial intelligence-based technologies better than their previous generation. However, in today's world, one of the significant challenges that need to be a concern in machine learning is regarding data breaches while training the model. Since federated learning trains machine learning algorithms in various devices or servers without sharing sample data. This paper discusses the framework of federated learning and homomorphic encryption and how both frameworks work together so that the outcoming data will be more precious and accurate without bothering data breaches. Later, we focus on its futuristic applications in various fields to improve technology.***
Keywords: Privacy-preserving; Machine learning; Artificial intelligence; framework; deep learning; sample data; data breaches; Ciphertext.

I. INTRODUCTION

In the last few decades, an enormous amount of private and sensitive data has been produced worldwide. The developer is using this data to train their machine learning model to make their Artificial intelligence-based application more innovative and intelligent. So, it is not wrong to say that one of the significant benefits of this enormous data is the deputed computation of machine learning statistical models.

These trained machine learning models and solutions are widely used in fields like Artificial Intelligence voice recognition, the health sector, fully automatic self-driving car, online search engine, face recognition, and many more.

But the challenge is the model's traditional training method. All types of user data get combined in a central location for machine learning training that may violate privacy policy laws of certain countries and even there is a higher chance to get data breaches.

Federated learning is one of the best solutions to these challenges.

Federated learning trains machine learning algorithm models on multiple devices without exchanging data samples that don't violate privacy policies. Google, one of the big tech companies, is using this technology to improve the Google keyboard on Android. But the major challenge in federated learning arises that needs to be concerned about the malicious attack. Studies have shown that situations can occur where one of the federations can maliciously attack the other, injecting hidden loopholes into the central global model and causing problems. Also, another issue is data heterogeneity. Finally, we combine data from different devices or servers to create a better model. However, this can limit the generalization of the model training from some sources and reduce the accuracy of new models.

However, homomorphic encryption is a type of encryption that allows the client to perform computations on the encrypted data without exposing the input data and internal state. Homomorphic encryption includes different encryption schemes that perform other classes of analysis on encrypted data.

Following this encryption, clients can train machine learning models without exposing data. The best thing about this is that no one encrypts the data you send. Since there are no other third-party users, the possibility of a data breach is almost negligible. Input privacy is a crucial problem with great success in machine learning, and isomorphic encryption is a solution to this problem. Researchers are working to find ways to make it accessible and convenient for everyone.

Thinking about all the challenges of federated learning, we decided to combine it with homomorphic encryption, which will help fix the privacy problems of federated learning and make it more secure to use.

1) *Privacy-Preserving:* Insecurity systems generally privacy-preserving means hiding users' input data from other users.
2) *Machine Learning:* This is part of the artificial intelligence that helps apps make more accurate predictions.
3) *Artificial Intelligence:* It is intelligence that develops machines to do work more significantly and accurately.
4) *Framework:* It is a supportive structure of something that will build.
5) *Deep Learning:* Deep learning is a subset of machine learning that teaches the machine about human nature.
6) Sample Data: It is part or subset of the primary data.
7) Data Breaches: Data breaches are incidents where the data of the users get stolen by hackers.
8) *Ciphertext:* A ciphertext is the result of encryption performed by an encryption algorithm over plaintext so that it cannot be read by anyone or a computer until there is a suitable cipher to decrypt it.

II. FRAMEWORK OF HOMOMORPHIC ENCRYPTION

Some of the world's largest companies are pushing for homogenous encryption to increase the security of user data. Homomorphic encryption is a type of data encryption. Craig Gentry created it in 2009, and in 2016 IBM released the first version of the HELib C++ isomorphic cryptographic library.

The main agenda of homomorphic encryption is to do computation on encrypted data. In starting, the client while transferring its data. The client needs to generate a pair of keys: a private key and a public key. The private key can also be called a hidden key, as it is the only key required to decode the encrypted data. This key is not being shared with anyone; only the client knows about it, and because of this private key, the process becomes secure. On the other hand, the public key is a type of asymmetric encryption that can be available for anyone to use. Currently, the recommended length of the key is about 2048 bits or 3072 bits.

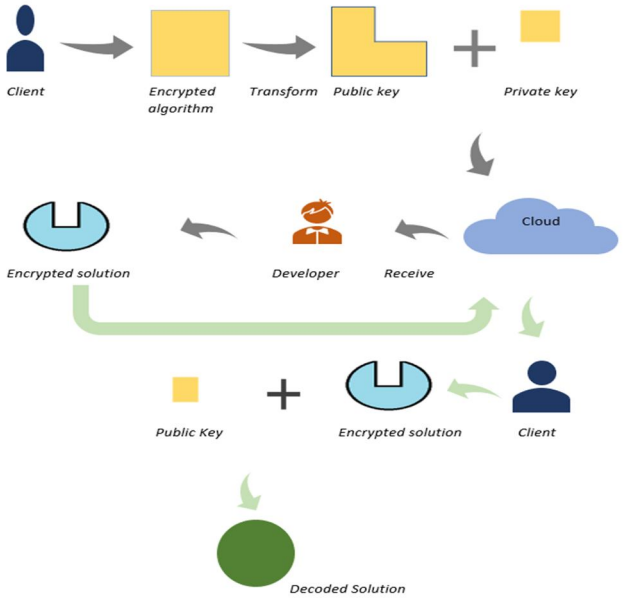


Figure 1. The framework of homomorphic encryption.

The encrypted data is being shared on the cloud. After this encryption, data becomes ciphertext, which is impossible for hackers or other users to decrypt. The developer receives the encrypted data, and then the developer starts to solve it. Although he is solving the information, he is entirely unaware of what he is precisely solving.

Further, when he gets the encrypted solution for the client, he sends it back to the cloud, and from there, the client receives the required resolution. Later, the client uses the private key generated at the beginning of the process to decode the encrypted solution. This way, the client does his work without knowing others about his data. The process is End-to-End encryption and makes our input data more protective.

III. FRAMEWORK OF FEDERATED LEARNING

Federated training allows businesses to train data across multiple devices or servers without data sharing. Federated learning can be defined as a distributed form of machine learning. This method uses user data to train a model on the user's device. Federated learning reduces power consumption, making the model more thoughtful and intelligent. This approach immediately benefits the model.

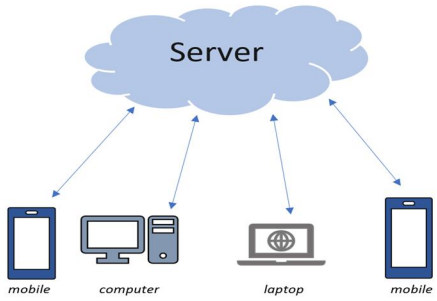


Figure 2. The framework of federated learning.

Today, people are surrounded by the latest technology and cutting-edge gadgets. They use those devices in their daily work. It seems that devices are becoming a part of human life. This is where federated learning uses the device. These devices already have artificial applications that users use according to their needs. The model gradually learned and started learning each time the user used the device. When a user enters data into a device to shop online, order food, or entertainment, the model starts learning and training accordingly, sometimes making the model brighter. After introducing the model, the AI application sends the model's results back to the central server. However, we do not send any user data to a central server when submitting results. This process happens simultaneously on millions of devices. All results are combined on a central server. The machine learning model on the primary server is then updated according to the aggregated results of the model, and the newly updated model is more innovative than the previous version. Now the modeler has sent an AI app update to the user's device, and after updating that AI app, the app is more innovative and user-friendly.

IV. WORKFLOW AFTER THE COMBINATION OF BOTH TECHNIQUES

Federated learning has its advantages and disadvantages, which we have discussed above. To get rid of the challenges regarding privacy, we decided to combine the technique of federated learning and homomorphic encryption.

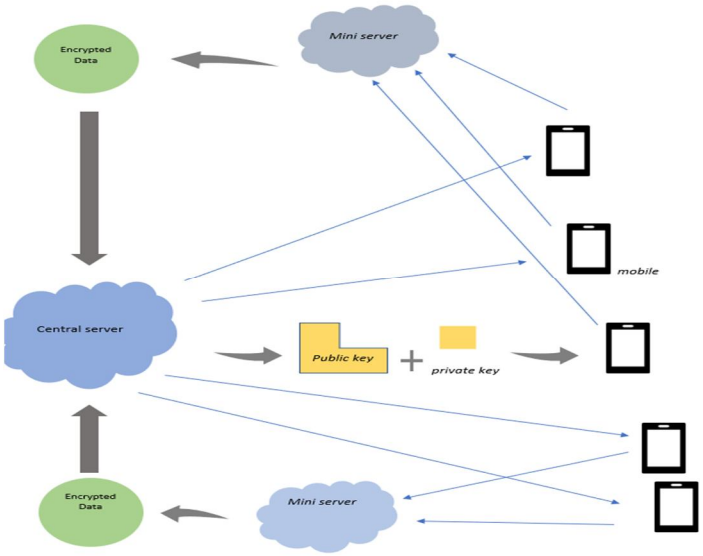


Figure 3. Combination framework of federated learning and homomorphic encryption.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue X Oct 2022- Available at www.ijraset.com

Like federated learning, the central server also sends an update to the machine learning model of Artificial intelligence applications or devices in an encrypted mode. In this technique, we use homomorphic encryption for privacy protection, so while sending updates to the apparatus, the server generates two types of keys: private-key encryption and public-key encryption. The private key encryption of data is not being shared. At the same time, the public key encryption is shared with the other devices or servers in the form of updates.

The artificial intelligence application is already installed on the user's device. After getting a notification for an update, the device user updates it. The application then got some new features, making our daily life easier and better.

For entertainment purposes or working purposes, we are almost dependent on technology. Using devices, we create massive data, and the model gets trained from this data. When the model gets prepared, the final encrypt results are sent to a mini cloud server. If any user thinks of creating backdoors into the central global model, he can't, as the update is encrypted. Simply put, the user can't find out what is
happening.

Many models are getting trained from many devices simultaneously from all over the world. Under a specific region, for all the devices or servers, there is a mini-server cloud that receives data from those devices. We can say that many devices send data to the mini-server cloud, and from many mini servers' data is transferred to the central centralized server.

The mini cloud receives the final results and aggregates the encrypted results. The final aggregate results are then sent to the central server. Finally, when the aggregation results from the major cloud, the central server decrypts it and updates its machine-learning model. This way, a model can be trained and advanced without fear of data breaches.

This framework helps us solve the federated learning's privacy and accuracy issues. For the specific region, there is a specific mini-server cloud and because of this, the connection between the devices and the cloud becomes speedier. As a result, the accuracy of getting the next model is high. Following the algorithm helps to demonstrate it.

```
import TensorFlow as tnf
import tensorflow_federated as tnff

#Data loaded.
source, _ = tnff.simulation.datasets.emnist.load_data()
def client_data(n):
  return source.create tnf_dataset_for client (source.client_ids[n]) .map(
    lambda e: (tnf.reshape(e['pixels'], [-1]), e['label'])
  ) .repeat (10) .batch (20)

#Receiving a subset from user.
data_trained = (client_data (n) for n in range(3)]

# Using of Keras model with TNFF.
def model_fn():
  model = tnf.keras.models.Sequential ([
      tnf.keras.layers.Dense(10,tnf.nn.softmax, input_shape=(784,),
        kernel_initializer='zeros')])
  return tnff.learning.from_keras model (model,
    input_spec=data_trained[0].element_spec,
    losses=tnf.keras. losses. SparseCategoricalCrossentropy(),
    metrics=[tnf.keras.metrics.SparseCategoricalAccuracy ()])
.   Training = tnff.learning.build federated_averaging_process(
   model fn,
   client_optimizer_fn=lambda: tnf.keras.optimizers.SGD(0.1))
state = training.initialize()
for _ in range(5):
  state, metrics = training.next (state,data_trained)
  print (metrics['train']['losses'])
```


Let's understand this framework with the help of an example. Suppose a contractor gets a contract for building a statue. He needs to make this within a limited specific period. He is thinking about how to do it within time and without leaking the design; later, he comes up with an idea. He first divides the design into three parts, then he uses his device and hires three sculptors from the internet from different areas and sends them different strategies to build the statue part as given in the design. Although the sculptor is doing the work, they are unaware of the complete work, and since all are doing it together with the same contractor work but a different part of the work, the work will be done soon, and the risk of leaking design is zero. After completing it, sculptors send the statue parts to the conductor by courier service. The courier carries the details and gives them to the conductor quickly. The conductor receives the different parts assembled and fulfills his contract within the time. Here in this example, the man plays the role of the central server, the sculptor plays the role of the devices, and the courier service plays the part of the mini server.

V. CONCLUSION

This paper discussed the concept and workflow of federated learning and homomorphic encryption, the drawbacks of federated learning, and how these drawbacks can be solved by the new framework combining both federated learning and homomorphic encryption. Even we discussed its application. But at present federated learning and homomorphic encryption are in the development stage. Many challenges of machine learning algorithms need to be solved to get a smooth framework of federated learning with homomorphic encryption. This technique must improve to secure our machine learning model training without data breaches. In the future, this technique will improve the multiple fields of technology. Companies like Google and Apple are researching to improve it. This technique has a bright future for the next generation and a significant role in the technological world.

VI. FUTURE USES

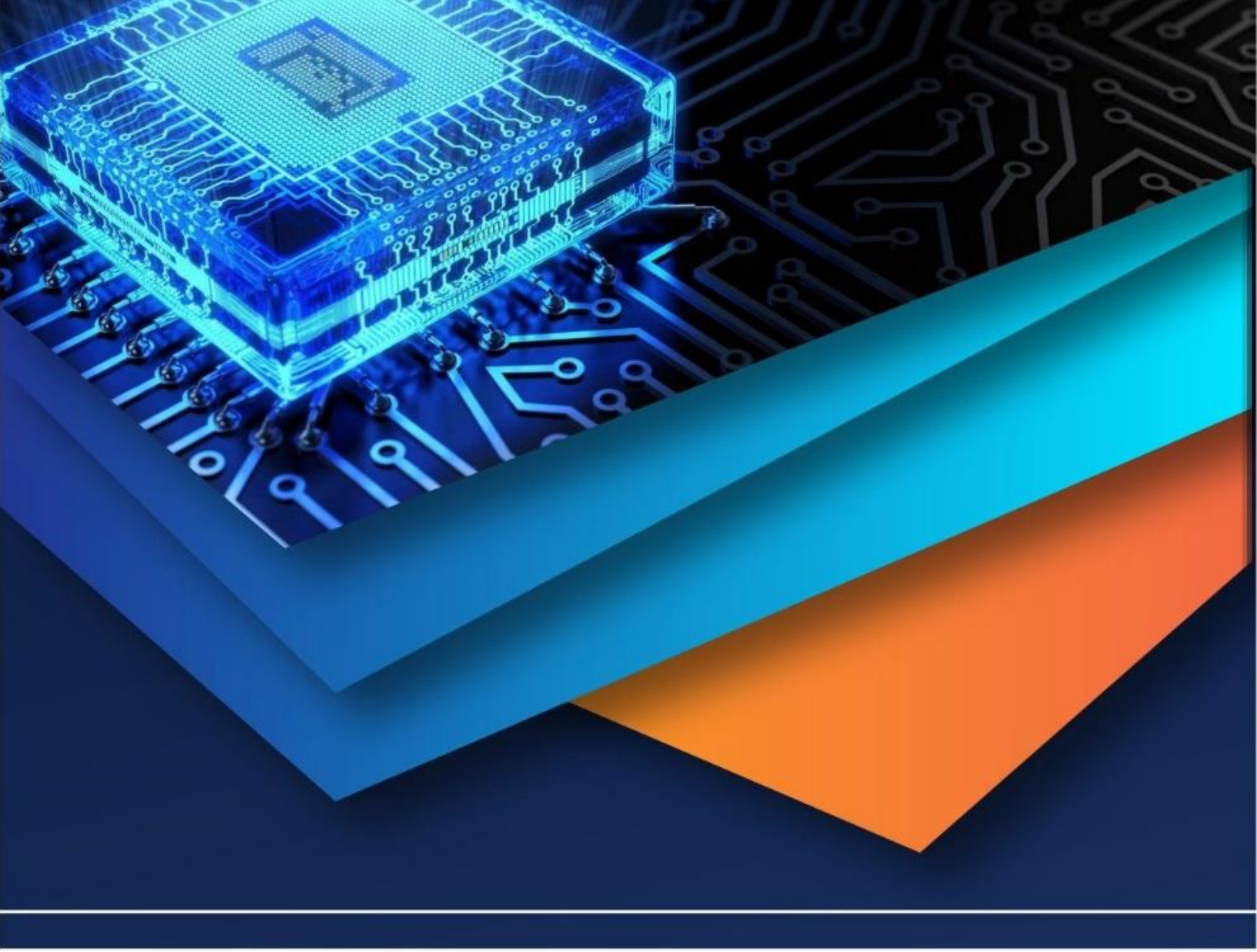
After the technique is developed, it can be used in many fields to improve science and daily human life. Some of the areas are discussed below:

1) *Health Sector:* Once it gets fully secured, hospitals, the medical manufacturing industry, and the health insurance company can take advantage of it. Using federated learning and homomorphic encryption, we can train our model to know the hospital's patients' various disease ratios, amount of medicine used per month, dead ratio, and diagnosis of rare diseases. Medicine companies will determine in what proportion they need to produce various types of medication. When a new condition comes to the hospital, the doctor can treat them. This data can be used to improve digital healthcare services.
2) *Self-driving car:* It helps to improve the model in real-time decision-making and continual learning. Self-driving cars not only help in driving without a driver but also help to save time by developing real-time traffic information. Through federated education and homomorphic encryption, artificial intelligence-based cars can improve themselves over time with input data from various other's vehicles.
3) *AI-voice recognition command:* Many companies are already developing this technology. Developer trains their model in the user devices. It prepares the model according to the user's behavior without leaking private data. The model can be trained to recognize user sounds to make it more secure. Voice commands can be used to operate a car, tv, phone, laptop, and so on.

REFERENCES

[1] Minelli, Michele. "Fully homomorphic encryption for machine learning." Ph.D. diss., PSL Research University, 2018.
[2] 876ruex S, Baracaldo N, Anwar A, Steinke T, Ludwig H, Zhang R, Zhou Y. A hybrid approach to privacy-preserving federated learning. InProceedings of the 12th ACM Workshop on Artificial Intelligence and Security 2019 Nov 11 (pp. 1-11).
[3] Fang H, Qian Q. Privacy-Preserving Machine Learning with Homomorphic Encryption and Federated Learning. Future Internet. 2021 Apr;13(4):94.
[4] Sukanya Bag. Federated learning – a beginners guide. Analytics Vidhya. 2021. Accessed at: https://www.analyticsvidhya.com/blog/2021/05/federated-learning-a-beginners-guide/
[5] Yang Q, Liu Y, Cheng Y, Kang Y, Chen T, Yu H. Federated learning. Synthesis Lectures on Artificial Intelligence and Machine Learning. 2019 Dec 19;13(3):1-207.
[6] Brendan McMahan and Daniel Ramage. Federated learning: Collaborative machine learning without centralized training data. Google AI Blog. Accessed at: https://ai.googleblog.com/2017/04/federated-learning-collaborative.html
[7] Aledhari M, Razzak R, Parizi RM, Saeed F. Federated learning: A survey on enabling technologies, protocols, and applications. IEEE Access. 2020 Jul 31;8:140699-725.
[8] Pier Paolo lppolito.AI differential privacy and federated learning. Towards data science. 2019. Accessed at: https://towardsdatascience.com/ai-differential-privacy-and-federated-learning-523146d46b85
[9] Yong Cheng, Yang Liu, Tianjian Chen, Qiang Yang Communications of the ACM, December 2020, Vol. 63 No. 12, Pages 33-3610.1145/338710
[10] Kai Hu, Yaogen Li, Min Xia, Jiasheng Wu, Meixia Lu, Shuai Zhang, Liguo Weng, "Federated Learning: A Distributed Shared Machine Learning-Method", Complexity, vol. 2021, Article ID 8261663, 20 pages, 2021.

[11] Sun X, Zhang P, Liu JK, Yu J, Xie W. Private machine learning classification based on fully homomorphic encryption. IEEE Transactions on Emerging Topics in Computing. 2018 Jan 17;8(2):352-64.
[12] Pandey R, Pandey VK. Cryptography & security implementation in network computing environments. In2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom) 2016 Mar 16 (pp. 3136-3140). IEEE.
[13] Chialva D, Dooms A. Conditionals in homomorphic encryption and machine learning applications. arXiv preprint arXiv:1810.12380. 2018 Oct 29.
[14] Kim M, Song Y, Wang S, Xia Y, Jiang X. Secure logistic regression based on homomorphic encryption: Design and evaluation. JMIR medical informatics. 2018 Apr 17;6(2):e8805.
[15] Pandey VK, Singh H. Enhancing the security of mac protocol in manet using the trust-based engine. In2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom) 2015 Mar 11 (pp. 1290-1295). IEEE.
[16] Niknam S, Dhillon HS, Reed JH. Federated learning for wireless communications: Motivation, opportunities, and challenges. IEEE Communications Magazine. 2020 Jul 15;58(6):46-51.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY