ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN030

Network Intrusion Detection System using Federated Machine Learning Approach

R Padmashani[1]; Harshan R.[2]; Logeshwaran C.[3]; Srikrishna R.[4]; Vijay Sundar[5]
Department of Information Technology,
PSG College of Technology

Abstract:- In the quickly changing digital world of today, protecting oneself from cyberattacks is crucial. This study presents a novel method that uses TensorFlow Federated (TFF) Learning to merge BiLSTM and DNN architectures, improving the precision and effectiveness of intrusion detection systems (IDS). TFF offers a major paradigm change in model training by enabling decentralized learning on several servers or devices. TFF provides IDS with collective intelligence by fostering collaborative learning on remote data sources while protecting data privacy. This improves detection accuracy and strengthens defenses against adversarial attacks. By utilizing TensorFlow Federated methods, IDS may run DNN and BiLSTM models concurrently, maximizing processing speed and resource efficiency. The system's capacity to manage high-throughput data streams is ensured by this concurrent execution, which speeds up threat detection and response. Moreover, information sharing and smooth integration between concurrent processes are made possible via synchronization and communication protocols. The cooperative synergy between the various models improves IDS's dependability and efficacy in thwarting emerging cyberthreats.

I. INTRODUCTION

In the fast-paced world of digital technology, defending against cyber threats is a critical endeavor. Traditional Intrusion Detection Systems (IDS) often struggle to keep pace with the rapidly evolving threat landscape. To address this challenge, this project proposes an innovative approach that integrates BiLSTM and DNN architectures within the IDS framework using TensorFlow Federated (TFF) Learning.

TensorFlow Federated (TFF) represents a paradigm shift in model training, enabling decentralized learning across diverse devices or servers. By fostering collaborative learning on distributed data sources while ensuring data privacy, TFF empowers IDS with collective intelligence. This collaborative approach enhances detection accuracy and fortifies resilience against adversarial attacks. Deep learning architectures, such as Deep Neural Networks (DNN) and Bidirectional Long Short-Term Memory (BiLSTM), play a crucial role in the development of the IDS. DNNs are well-suited for capturing complex patterns and relationships within large-scale datasets, making them ideal for identifying subtle anomalies in network traffic. BiLSTM excels at capturing temporal dependencies and context from both past and future data points, enabling IDSs to detect sophisticated intrusion patterns that evolve over time.

By integrating DNN and BiLSTM models into the IDS framework and leveraging parallel training techniques, a more robust and efficient Intrusion Detection System can be created. These deep learning architectures complement each other, allowing IDSs to effectively capture spatial and temporal features in network traffic data. Additionally, parallel training enables the models to learn concurrently, accelerating the overall training process and enhancing the system's ability to adapt to changing cyber threats. In addition to enhancing detection accuracy and resilience against adversarial attacks, this collaborative approach also optimizes resource utilization and processing speed, ensuring timely threat detection and response. The project also explores potential applications for the enhanced IDS across diverse sectors including Banking, Finance, Healthcare, Defense, and E-commerce, underscoring its versatility and significance in safeguarding critical digital assets within an interconnected digital landscape.

II. EXISTING WORKS

In his research, Muhammad Ashfaq Khan [1] proposed the use of a Convolutional Recurrent Neural Network (CRNN) to develop a deep learning-based hybrid intrusion detection system that can identify and classify potentially dangerous network intrusions. The CSE-CIC- DS2018 [2] intrusion dataset was utilized to train the suggested methodology. Utilizing the HCRNN methodology and a few common classification techniques like Logistic Regression, Decision Tree, XGBoost, etc., the proposed Intrusion Detection system was put into practise.

In a paper by Javed Asharf et al. [3], the researchers sought to give a thorough analysis of the technologies, protocols, architecture, and dangers that arise from hacked Internet of Things devices as well as an overview of intrusion detection methods. The examination of several machine learning and deep learning-based methodologies appropriate to identify IoT devices vulnerable to cyberattacks is also included in this paper. The problem is that there isn't a common mechanism that ensures the suggested systems' or method's validity. The

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN030

majority of research studies provide assessment of the suggested systems using simulated datasets, which may not be applicable to real- world situations with actual data and other challenges.

In a research published by Congyuan Xu et al [4], the traits of the time-related incursion were taken into account. Recurrent neural networks with gated recurrent units (GRU), multilayer perceptrons (MLP), and softmax modules make up a unique IDS that has been presented. The suggested solution was developed using the NSL-KDD [5] and KDD Cup 99 [6] datasets. The theoretical verification is mostly responsible for the suggested system's limitations in this study.

On Coburg Intrusion Detection Datasets (CIDDS), Niraj Thapa et al [7] presented a comparative study of several ML models and DL models. On the CIDDS dataset, various ML and DL-based models have first been contrasted. Second, a model ensemble combining the top ML and DL models is suggested to obtain high- performance metrics. Finally, using the CICIDS2017 dataset, the best models are compared to the most current models. The primary disadvantage would be that different sorts of assaults are not included in the dataset utilized. It does not provide defenses against complex adversarial assaults.

Moving towards dynamically created datasets that not only represent the traffic compositions and intrusions of the moment but are also changeable, expandable, and repeatable is important as network behaviors and patterns change and intrusions grow. Ali Shiravi et al. [8] established a methodical method in this research to create the needed datasets to meet this demand. For HTTP, SMTP, SSH, IMAP, POP3, and FTP, genuine traces are analyzed to establish profiles for agents that produce real traffic.

A profile consists of an abstract representation of various features and events to make it simpler to recreate particular real-world behaviors as seen from the network. Then, agents or human operators use these profiles to create network events. Two broad classes of profiles are and profiles try to describe an attack scenario as precisely as possible. The simplest case is that individuals can comprehend these profiles and subsequently take appropriate action. Compilers and autonomous agents would be employed to interpret and perform these scenarios in a perfect world. Profiles are techniques with pre and post conditions that include mathematical distributions or behaviors of certain entities that have been extracted. Examples include how often a protocol uses different packet sizes, how many packets are in a flow, and certain patterns.

The requirement to initially construct and then run profiles causes complications. Profiles need a specialized understanding of how an assault is put together. Filtered network traces are needed for the production of -profiles but they might not be readily available. Some of this routine traffic may contain hidden signs of assaults, which will have an impact on the final profile that is derived.

Six Machine learning-based IDSs were suggested by Bhavani et al. [9] utilizing the algorithms K Nearest Neighbour, Random Forest, Gradient Boosting, Adaboost, Decision Tree, and Linear Discriminant Analysis. The dataset CSE-CIC-IDS2018 [10] was utilized. Additionally unbalanced is the chosen dataset. Bias towards the dominant class results from unbalanced datasets, and in certain extreme cases, minority classes are overlooked. These minority groups, nevertheless, are often advantageous ones. Therefore, the imbalance ratio is decreased by employing a synthetic data generation model called Synthetic Minority Oversampling Technique (SMOTE) in order to boost the efficiency of the system depending on attack types and to decrease missed incursions and false alarms. The suggested technique significantly boosted the detection rate for infrequent incursions, according to experimental data.

Alqahtani et al [11] employed various popular Machine learning classification algorithms, namely Bayesian Network, Naive Bayes classifier, Decision Tree, Random

Decision Forest, Random Tree, Decision Table, and Artificial Neural Network to detect intrusions. Finally, the effectiveness of various experiments on Cybersecurity datasets having several categories of cyber attacks were tested and evaluated on the effectiveness of the performance metrics, precision, recall, F1 score, and accuracy.

Variational autoencoders (VAE) were suggested as a technique by Jinwon [12] for anomaly identification. A probabilistic graphical model called a variational autoencoder combines DL with variational inference. By taking into consideration the idea of variability, the reconstruction probability combines the variational autoencoder's probabilistic traits. Compared to the reconstruction error of autoencoder and Principal Component Analysis (PCA) based approaches, the reconstruction probability is a probability measure, making it a far more objective and principled anomaly score. The suggested technique outperforms autoencoder and PCA-based algorithms, according to experimental data. It is also feasible to deduce the reconstruction of the data to study the underlying cause of the anomaly because of its generative properties.

In order to create a flexible and effective IDS to identify and categorize unanticipated and unpredictable cyberattacks, Vinayakumar et al.

[13] investigated a Deep Neural Network (DNN), a type of Deep learning model. The fast growth of attacks and the ongoing change in network behaviour need the evaluation of multiple datasets that have been produced over time using both static and dynamic methods. This study makes it easier to identify the best algorithm for reliably identifying upcoming threats. A thorough analysis of DNN and other traditional

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN030

machine learning classifier studies is presented using a variety of publicly accessible benchmark malware datasets, such as the KDDCup 99 dataset. Through the use of hyperparameter selection methods and the KDDCup 99 dataset, the best network parameters and topologies for DNNs are determined.

An agent-based distributed intrusion detection system architecture was put out in this study by Riyad et al [10]. Mobile agents are used by the system for analysis and detection. Since the process now travels to the data for analysis, the network latency is much reduced. Here, agents work together to jointly detect and analyse, enabling informed decision-making. Effective intrusion detection was accomplished using an ensemble data mining methodology. With the help of numerous modules in the design, the system is capable of responding to fresh threats in the future. The agents' autonomy allows them to quickly replace a broken agent, which makes for a good fault tolerance mechanism. The JADE platform for mobile agents was used for the studies, and the outcomes are quite encouraging.

III. PROPOSED METHODOLOGY FOR NETWORK INTRUSION DETECTION SYSTEM

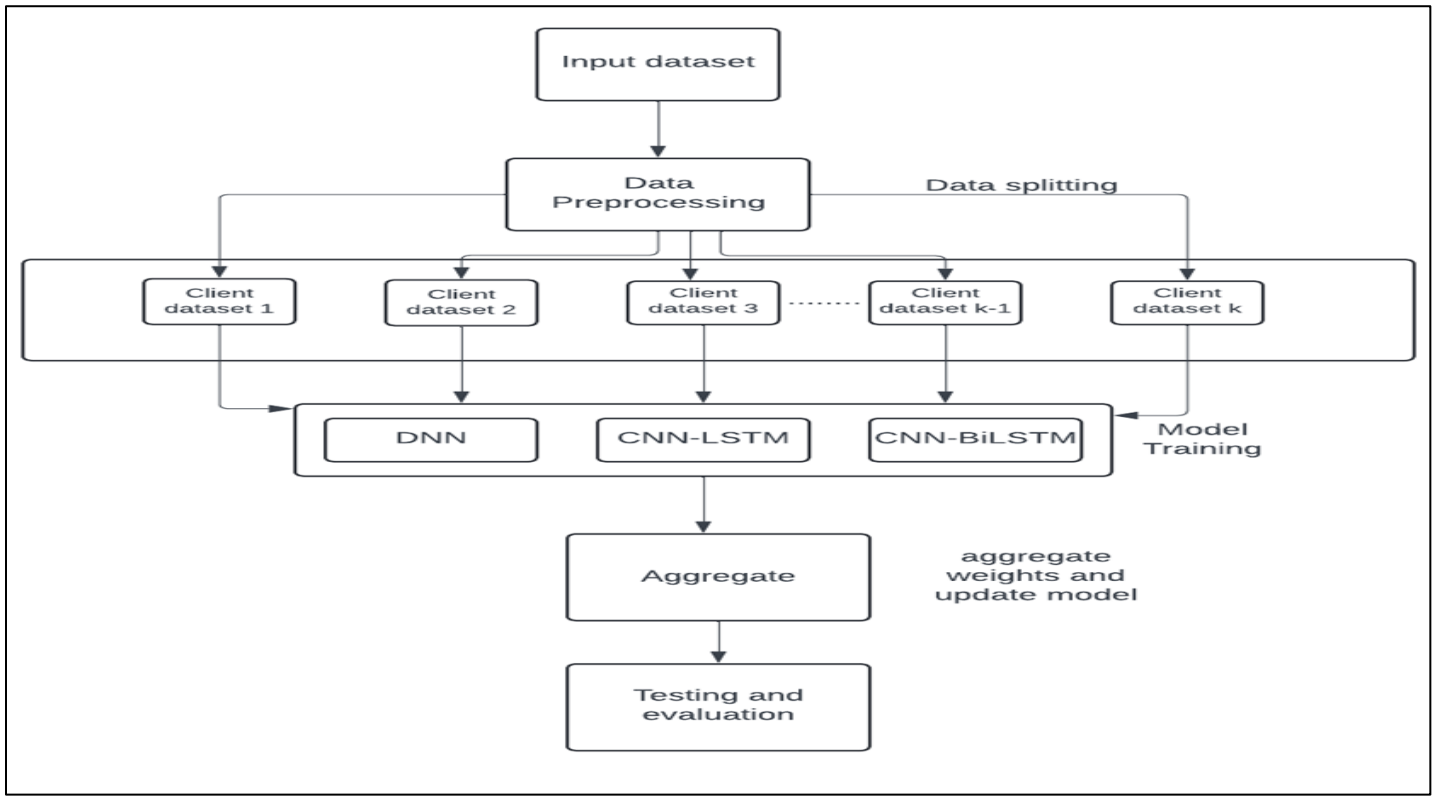

Fig 1: Proposed Methodology

➢ *Dataset Description*

The project aims to enhance network security through the development of robust intrusion detection systems using Federated Machine Learning and techniques. To achieve this goal, the project utilizes two benchmark datasets: the NSL-KDD dataset and the IEC 60870-5-104 dataset. For each traffic record in the NSL-KDD dataset, there are 41 features and one category tag, including basic features, information, and traffic features. Attacks in the database are classified into four types of attacks according to their characteristics: DoS (Denial of Service Attacks), R2L (Root to Local Attacks), U2R (User to Root Attack) and Probe (Test Attacks). For each traffic record in the IEC Dataset, there are 83 features and one label tag. In the dataset there are two types of the labels, which are normal and anomaly. The training data as well as the testing data contains 12 classes of traffic including 11 attack classes and one normal class.

➢ *Dataset Preprocessing*

The preprocess function which is common for both the datasets defines a nested function called map_fn, which maps each dataset element to an Ordered Dictionary with features (x) and labels (y). The features are cast to float64 using TensorFlow operations, and the labels are then reshaped appropriately. From the input DataFrame, the function creates a TensorFlow dataset by repeating it over epochs multiple epochs, batching the data, prefetching batches to increase performance, and rearranging the data with a given buffer size. After applying this preprocessing function to a list comprehension, the function samples data points from the training set to create pre-processed

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN030

datasets for multiple clients. This is common in federated learning, in which every client might have a local training dataset of its own. Overall, the code ensures data consistency and effective processing across multiple clients by establishing a strong preprocessing pipeline appropriate for federated learning scenarios.

- *Intrusion Detection using Federated Learning*

In a federated learning setup, K clients act as gateways for monitored systems, training local models individually. Each of these K models, denoted as *k*=1.., shares an identical structure, meaning they possess the same number of layers and neurons per layer. They are, however, trained on separate datasets provided by their connected clients.

As a result, the available user clients will be used to train the data locally and compute the update

to the server's shared global model, which will aggregate all of the updates from the distributed devices and compute weight using the Federated Averaging algorithm in the Equation (1).

$$w_{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} w^k_{t+1} \quad (1)$$

In this formula, n_k is the size of the partition of client k such that n is the sum of all partitions. Further w^k_{t+1}, is the local weight of client k which is averaged over the summation.

- *Federated Aggregation of the Local Weight Updates into the Master Model*

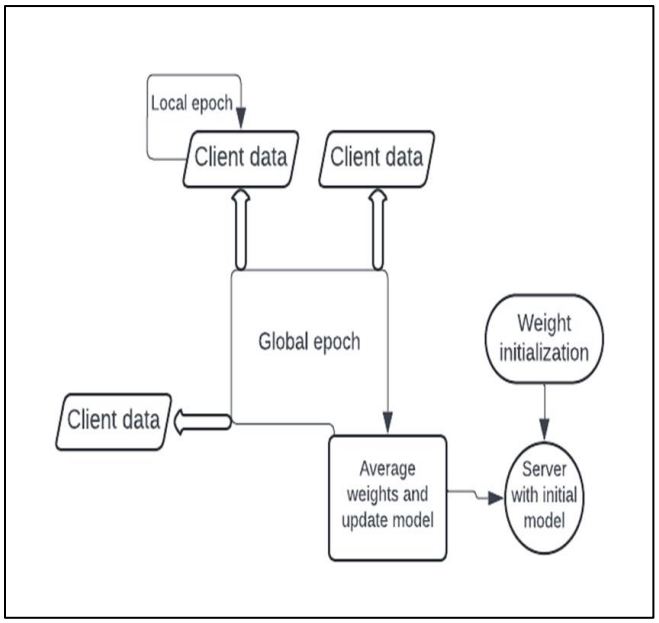

Fig 2: Federated Averaging

In this federated learning process, as illustrated in Figure 2, each client conducts multiple local epochs of training on its data, represented by the dashed arrows. The model parameters are initialized beforehand, ensuring consistency across client devices. Meanwhile, the central server initiates with an initial model, serving as a starting point for the collaborative learning process. After each training round, updates from client devices are transmitted to the server, where they are aggregated by computing the average of model parameters, as depicted by the solid arrow leading to the "Average Weights" step. This aggregation mechanism ensures that individual client contributions are integrated into the global model while preserving data privacy. Subsequently, the updated global model is refined on the server, incorporating the aggregated updates, before being redistributed to all client devices for further training rounds. This iterative exchange of model updates continues until the model achieves the desired performance metrics, facilitating collaborative learning across distributed data sources while addressing privacy concerns and regulatory requirements.

IV. RESULT ANALYSIS

The results obtained evaluation of the federated learning model on both the IEC and NSL-KDD datasets reveals promising performance in network intrusion detection. The purpose of using federated learning in NIDS is to enhance the overall security posture by leveraging distributed data sources without compromising data privacy. The below are the obtained results:

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN030

Table 1: Results of IEC Dataset

Epoch	Accuracy	Loss
1	93.3984	0.1199
2	92.7214	1.1685
3	97.5911	0.3796
4	99.1927	0.1284
5	98.9844	0.1602
6	99.4922	0.0799
7	99.5378	0.0642
8	99.043	0.1129
9	99.7656	0.0357
10	99.7721	0.0358

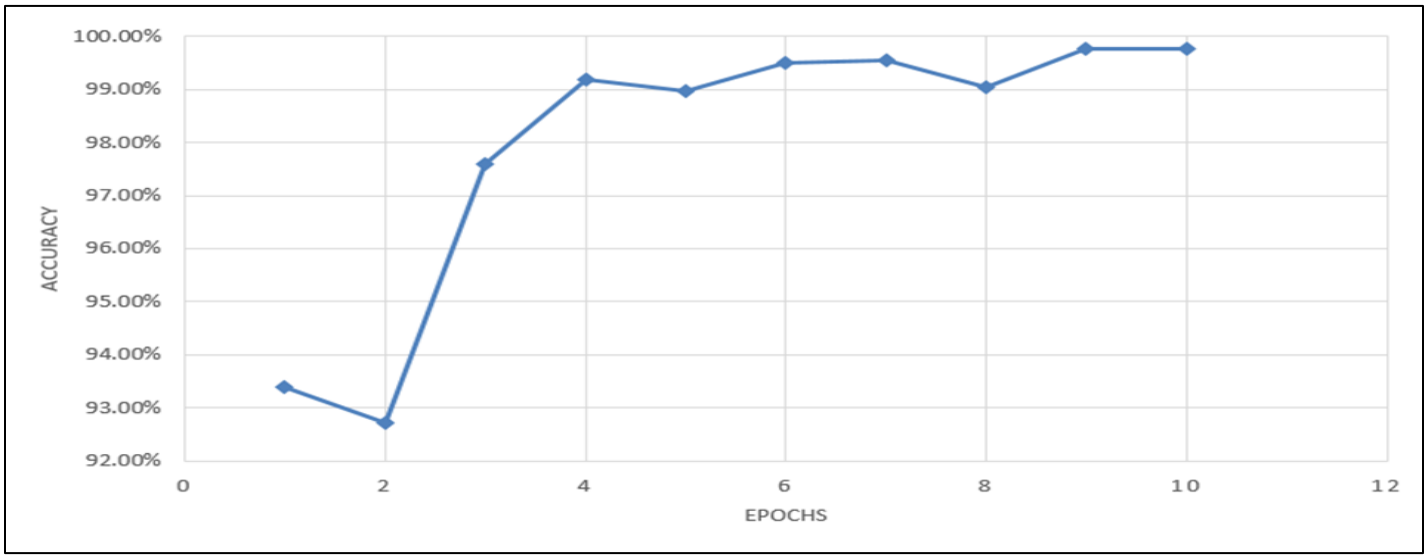

Fig. 3 Accuracy of IEC Dataset

Table 2: Training Results on NSL KDD Dataset

Epoch	Accuracy	Precision	Recall
10	95.41%	94.14%	97.50%
15	96.20%	94.61%	98.51%
20	96.27%	94.49%	98.80%
25	96.57%	94.61%	99.24%
30	96.49%	94.44%	99.30%
35	96.53%	94.42%	99.40%
40	96.89%	94.87%	99.57%
45	96.97%	94.98%	99.61%
55	96.99%	95.02%	99.59%
75	96.38%	94.18%	99.38%

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN030

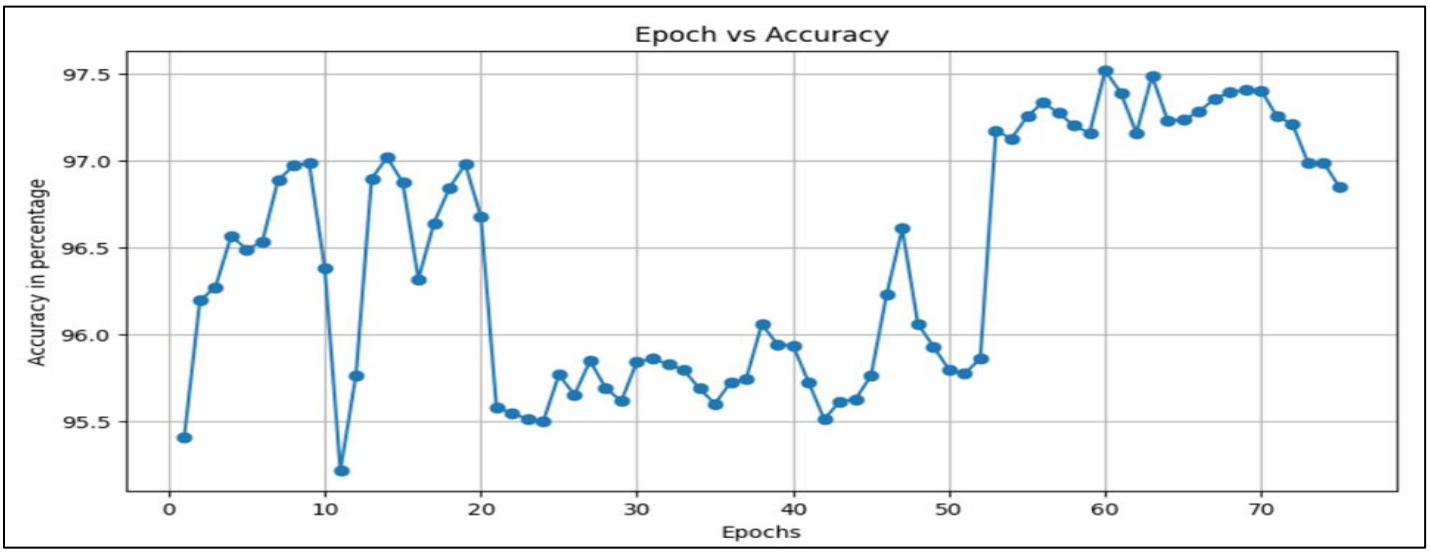

Fig 4: Training Accuracy of NSL KDD Dataset

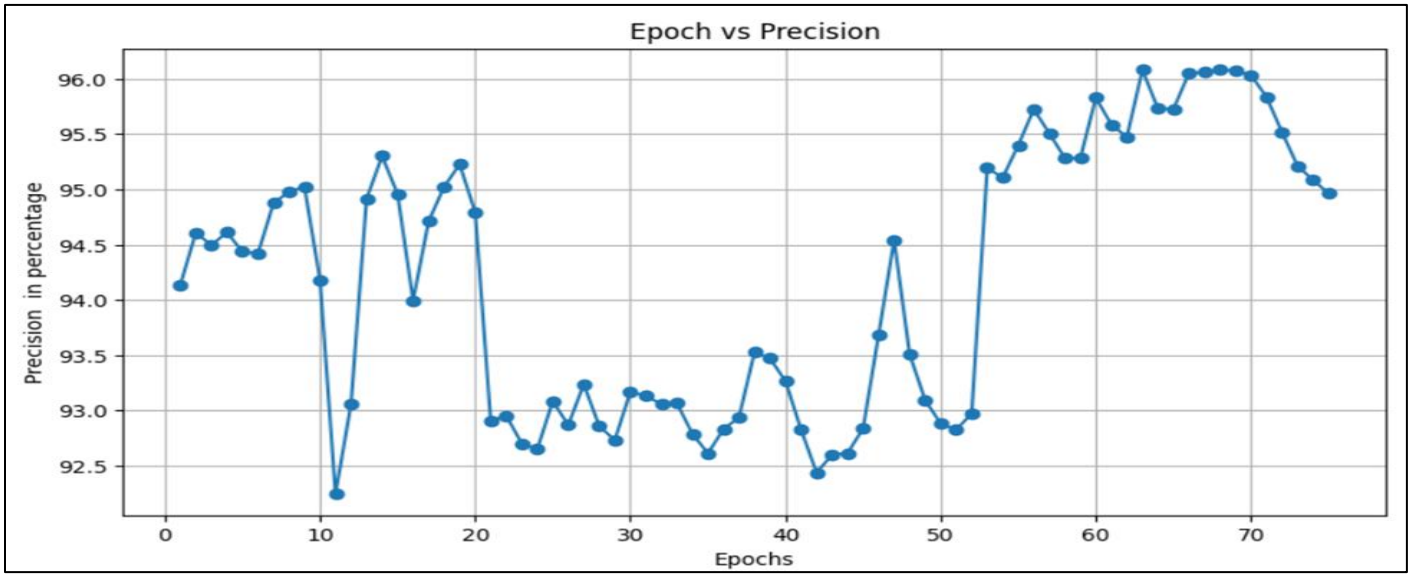

Fig 5: Training Precision of NSL KDD Dataset

Table 3: Testing Results on NSL KDD Dataset

Epoch	Accuracy	Precision	Recall
10	85.69%	83.25%	83.74%
20	90.13%	93.58%	82.85%
30	90.60%	94.80%	82.78%
40	90.71%	95.90%	82.02%
50	91.21%	96.17%	82.95%
60	94.38%	97.91%	88.90%
70	93.61%	97.81%	87.17%
80	94.30%	96.74%	89.84%
90	93.76%	97.27%	88.02%
100	94.20%	97.60%	88.77%

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN030

Table 4: Overall Result Analysis

Dataset	No. of Clients	No. of Epochs	Accuracy	Recall	Precision	F1 Score
IEC	2	10	99.71%	91.65%	99.85%	93.72%
NSL-KDD	2	100	95.43 %	88.76%	97.59%	92.85%
NSL-KDD	5	100	88.38%	97.74%	79.71%	87.81 %
NSL-KDD	10	100	90.31%	95.74%	83.89%	89.42%

The model achieved high accuracy rates, reaching 99.71% on the IEC dataset and a 95.43% on the NSL-KDD dataset in a binary classification setting with two classes. Accuracy is a critical metric in NIDS as it measures the overall correctness of the model's predictions. When considering the impact of client numbers on the NSL-KDD dataset, the accuracy slightly decreased from 95.43% with two clients to 88.38% with five clients and 90.31% with ten clients, highlighting potential challenges with increased client heterogeneity. Recall, which measures the ability of the model to identify all relevant instances, remained consistently high, demonstrating the model's effectiveness in detecting intrusions. Precision, which measures the proportion of true positives among all positive predictions, and F1 Score, which combines precision and recall, are crucial for assessing the model's performance in minimizing false alarms while maximizing the detection of actual intrusions.

V. CONCLUSION AND FUTURE ENHANCEMENT

The integration of BiLSTM and DNN architectures through TensorFlow Federated learning in this project represents a significant breakthrough in Intrusion Detection Systems (IDS). By leveraging decentralized learning and collaborative intelligence, this approach enhances the accuracy and efficacy of detecting cyber threats. The BiLSTM component captures bidirectional dependencies, enabling the system to analyze past and future contextual information, while DNNs excel in identifying complex patterns within datasets. Through TensorFlow Federated learning, the IDS efficiently processes data across distributed sources, enhancing scalability and enabling timely threat detection.

The system's ability to adapt and learn from decentralized data sources ensures continuous improvement in threat detection capabilities, making it a valuable asset for safeguarding digital infrastructures against evolving security challenges. Major area to focus on for advancing this project involves refining the model's architecture to integrate more sophisticated deep learning techniques. Exploring innovative approaches for feature extraction and representation learning can enhance the IDS's ability to detect subtle anomalies in network traffic data. Additionally, optimizing the federated learning process by fine tuning communication protocols and implementing strategies for efficient model aggregation will further improve the system's scalability and performance. Moreover, integrating real-time threat intelligence feeds and implementing adaptive learning mechanisms would enable the IDS to dynamically adjust its detection capabilities in response to emerging cyber threats. These enhancements collectively strengthen the effectiveness and reliability of the Intrusion Detection System, ensuring robust cybersecurity defenses for organizations in the face of evolving security challenges.

REFERENCES

[1]. Muhammad Ashfaq khan,” HCRNNIDS: Hybrid Convolutional Recurrent Neural Network-Based Network Intrusion Detection System”,pp 6-8
[2]. Zhang, Chen, et al. "A survey on federated learning." *Knowledge-Based Systems* 216 (2021): 106775.
[3]. Javed Asharf ,Nour Moustafa , Hasnat Khurshid ,Essam Debie ,Waqas Haider ,Abdul Wahab,"A Review of Intrusion Detection Systems Using Machine and Deep Learning in Internet of Things: Challenges, Solutions and Future Directions",pp 12-26
[4]. Conguyan Xu, Jizhong Shen ,Xin Du, Fan Zhang,”An Intrusion Detection System Using a Deep Neural Network With Gated Recurrent Units”,pp 4-9
[5]. Li, Li, et al. "A review of applications in federated learning." *Computers & Industrial Engineering* 149 (2020): 106854.
[6]. Mammen, Priyanka Mary. "Federated learning: Opportunities and challenges." arXiv preprint arXiv:2101.05428 (2021).
[7]. AL-barakati, Niraj Thapa, Saigo Hiroto , Kaushik Roy , Robert H. Newman , Dukka KC, “RF-MaloSite and DL-Malosite: Methods based on random forest and deep learning to identify malonylation sites”, pp 8-20
[8]. Ali Shiravi, Hadi Shiravi, Mahbod Tavallaee, Ali A. Ghorbani,”Toward developing a systematic approach to generate benchmark datasets for intrusion detection”, pp 2-14

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24JUN030

[9]. Lei Wang, Latifur Khan and Bhavani Thuraisingham,"An Effective Evidence Theory based K-nearest Neighbor (KNN) classification",pp 3-12
[10]. Rieke, Nicola, et al. "The future of digital health with federated learning." NPJ digital medicine 3.1(2020): 17.
[11]. Hamed Alqahtani, Iqbal H. Sarker,Asra Kalim,Syed Mohammod ,Minhaz Hossain,"Cyber Intrusion Detection Using Machine Learning Classification Techniques",pp 4-10
[12]. Jinwon An, Sungzoon Cho, "Variational Autoencoder based Anomaly Detection using Reconstruction Probability",pp 5
[13]. Li, Qinbin, Bingsheng He, and Dawn Song. "Model-contrastive federated learning." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021