NATURENGS
E-ISSN 2717-8013
https://dergipark.org.tr/en/pub/naturengs

Federated Learning for Attack Prediction on UNSW-NB15 Training Data

Hayriye TANYILDIZ [a,*], Canan BATUR ŞAHİN [a], Özlem BATUR DİNLER [b]

[a] Malatya Turgut Ozal University, Faculty of Engineering and Natural Sciences, Malatya, 44210, Turkey
[b] Siirt University, Faculty of Engineering, Department of Computer Engineering, Siirt, 56000, Turkey

* Corresponding author

ARTICLE INFO

Received 29.07.2024
Accepted 07.10.2024

Doi: 10.46572/naturengs.1524394

ABSTRACT

Traditional centralized machine learning models for network intrusion detection systems (NIDS) face significant challenges related to data confidentiality, scalability, and centralization risks. This study highlights the potential of Federated Learning (FL), a decentralized approach that enables multiple clients to train a shared model while keeping their data local collaboratively. FL is particularly relevant for attack prediction in cybersecurity, where organizations may be reluctant to share sensitive data due to privacy and security concerns. This paper investigates the application of FL on the UNSW-NB15 dataset to predict network attacks. The study demonstrates that FL can achieve high accuracy and minimal false negatives in attack detection while preserving data confidentiality. The results, visualized through a confusion matrix, underscore the effectiveness of FL in distinguishing between normal and malicious network traffic, making it a promising approach for real-world cybersecurity applications. By leveraging FL, organizations can enhance their network security infrastructure while mitigating the risks associated with centralized data processing.

Furthermore, this study's findings suggest that FL has broad application potential in cybersecurity and other domains where data privacy is crucial. However, FL still has limitations such as network latency, heterogeneous data distributions, and communication costs. Future research should address these challenges to optimize the use of FL in various contexts further.

Keywords: Federated Learning, Anomaly Detection, Cyber Attack

1. Introduction

The rapid proliferation of the Internet and connected devices has increased cyber threats and made network security a critical concern for organizations and individuals. Network Intrusion Detection Systems (NIDS) are essential tools for identifying and mitigating potential threats in network traffic. Traditional machine learning models for NIDS are typically centralized, requiring data from multiple sources to be aggregated into a central repository for training. While this approach is practical, it poses significant data privacy, security, and scalability challenges.

Federated Learning (FL) has emerged as a promising solution to these challenges. It enables decentralized model training across multiple nodes without transferring raw data to a central server. This approach addresses privacy concerns, reduces the risk of data breaches, and increases scalability.

In FL, a global model is trained collaboratively by aggregating locally computed updates from multiple clients, ensuring sensitive data remains on local devices. By using federated learning for attack prediction, organizations can collaboratively train machine learning models without centralizing sensitive data. Each participating entity can train the model on its local dataset and only share model updates rather than raw data. This helps maintain privacy and security while allowing collective insights from diverse datasets. Federated learning also enables organizations to benefit from a broader range of threat intelligence without directly exchanging sensitive information. It empowers collaborative predictive modeling across different entities or sectors within the cybersecurity ecosystem, leading to more robust attack predictions. However, challenges exist with federated learning, such as potential communication overhead, ensuring consistency across decentralized models, and addressing issues related to bias and fairness.

* Corresponding author. e-mail address: tanyildiz003@gmail.com
ORCID : 0000-0002-6300-9

A comprehensive benchmark for network intrusion detection, the UNSW-NB15 dataset extracts data from Network traffic, including both standard and malicious activities. It offers various features. This dataset is ideal for evaluating the performance of FL in detecting network attacks. Leveraging federated learning with the UNSW-NB15 dataset for attack prediction has promising potential in cybersecurity research and practice. It enables collaborative model training while maintaining data privacy and security, contributing to enhanced threat intelligence and improved attack prediction capabilities.

This paper investigates the effectiveness of FL in predicting network attacks using the UNSW-NB15 dataset. In order to assess these aspects and ultimately determine the effectiveness of Federated Learning in predicting network attacks using UNSW-NB15, empirical studies and experiments could be conducted comparing Federated Learning against centralized approaches concerning model performance, privacy preservation, and scalability specifically within a cybersecurity context utilizing this dataset for intrusion detection research. Leveraging FL, we aim to show how decentralized learning can achieve high accuracy and low false negative rates in attack detection while preserving data privacy. The work involves implementing an FL framework, training a neural network model on UNSW-NB15 training data, and evaluating its performance using a confusion matrix.

The rest of the paper is organized: Section 1 reviews related work on applying machine learning and FL for intrusion detection in various fields. Section 2 describes the methodology, including the FL framework and experimental setup, and details the dataset used. Section 3 presents the results and analysis of the model's performance. Chapter 4 discusses the implications of the findings and potential challenges. Finally, By examining the application of FL on the UNSW-NB15 dataset, this paper contributes to the growing body of knowledge on privacy-preserving machine learning techniques for cybersecurity. It highlights the potential of FL in enhancing network security in decentralized environments.

This work makes several significant contributions to network intrusion detection and privacy-preserving machine learning. It demonstrates the use of Federated Learning (FL) as an effective method to decentralize the training process, enabling privacy-preserving and improved scalability when dealing with sensitive network traffic data. It also highlights the comparative advantages of FL over traditional centralized machine learning approaches, including reduced privacy risks, improved scalability, and a broader range of threat intelligence.

1.1 Literature Review

Li et al. [1], the SELSTM model combining NSENet and LSTM provided a robust solution for IoT attack detection by improving feature extraction capabilities, optimizing model convergence, and providing effective detection with limited computational resources. Experimental results confirmed its superiority over traditional models in terms of sensitivity, accuracy, and generalizability in attack detection, and they achieved an accuracy value of 82.14%. Sharma [2] proposed a method that works in three stages. First, they used the ExtraTrees classifier (ELM) to individually select relevant features for each attack type. They then used an ELM ensemble to detect each attack type individually. Finally, the results of all ELMs were combined using a softmax layer to improve the results and further increase the accuracy. The accuracy rate obtained was 91.26%. Gharaee et al. [3] proposed an anomaly-based IDS with a new feature selection method using a genetic algorithm and a Support Vector Machine (SVM). The new model used a Genetics-based feature selection method including an innovation in the fitness function. It reduces the data size, aiming to increase true positive detection and simultaneously reduce false positive detection. Salman et al. [4] investigated detecting and categorizing anomalies. They used two supervised machine learning techniques: linear regression (LR) and random forest (RF). As a result, they achieved 93.6% accuracy. Zhang et al. [5] implemented a filter-based feature reduction technique using Artificial Neural Networks (ANN) and Decision Trees (DT). Our experiments considered both binary and multi-class classification configurations. The results showed that the XGBoost-based feature selection method enabled methods such as DT to increase the testing accuracy from 88.13% to 90.85% for the binary classification scheme. Salim et al. [14] focused on securing IIoT environments by using a Federated Learning-based CTI framework (FL-CTIF) to detect anomalous traffic patterns, including ARP poisoning tool attacks, SSL-based attacks using encrypted traffic, and DNS flood-based DDoS traffic. They achieved successful results. Boabalan et al. [15] proposed the FusionFedBlock framework, which combines Blockchain with Federated Learning in Industry 5.0. This approach addresses the dual challenges of secure data sharing and collaborative learning, leading to more resilient and intelligent industrial systems.

2. Materials and Methods

2.1 Data Set

The UNSW-NB15 dataset was developed by the Australian Center for Cyber Security (ACCS) and is widely used in research on intrusion detection systems. This dataset includes nine different attack types along with regular network traffic. It was created using the IXIA PerfectStorm tool to create a hybrid of actual modern normal activities and synthetic contemporary attack behaviors. The dataset comprises 49 features, including the class label and 2,540,044 records. The table below provides an overview of the features of the UNSW-NB15 dataset:

Table 1 Dataset Column Description

No.	Feature	Description	No.	Feature	Description
1	scrip	Source IP address	26	res_bdy_len	Length of the response body
2	sport	Source port number	27	sjit	Source jitter
3	dstip	Destination IP address	28	djit	Destination jitter
4	sport	Destination port number	29	time	Source time
5	proto	Protocol type	30	ltime	Destination time
6	state	State of the connection	31	sintpkt	Source inter-packet arrival time
7	dur	Duration of the connection	32	dintpkt	Destination inter-packet arrival time
8	bytes	Source to destination bytes	33	tcprtt	TCP round-trip time
9	bytes	Destination to source bytes	34	synack	Time between SYN and SYN-ACK packets
10	sttl	Source to destination time to live	35	backdate	Time between SYN-ACK and ACK packets
11	dttl	Destination to source time to live	36	is_sm_ips_ports	If source and destination IP addresses and port numbers are equal
12	sloss	Source packets retransmitted or dropped	37	ct_state_ttl	Number of connections with the same state and time-to-live
13	dloss	Destination packets retransmitted or dropped	38	ct_flw_http_mthd	Number of connections with the same HTTP method
14	service	Network service (e.g., HTTP, FTP, ssh)	39	is_ftp_login	If the FTP session is authenticated
15	load	Source bits per second	40	ct_ftp_cmd	Number of FTP commands issued
16	dload	Destination bits per second	41	ct_srv_src	Number of connections to the same service from the source IP
17	spots	Source to destination packet count	42	ct_srv_dst	Number of connections to the same service to the destination IP
18	dpkts	Destination to source packet count	43	ct_dst_ltm	Number of connections to the same destination IP
19	swin	Source TCP window advertisement	44	ct_src_ltm	Number of connections from the same source IP
20	Edwin	Destination TCP window advertisement	45	ct_src_dport_ltm	Number of connections from the same source IP and destination port
21	stcpb	Source TCP base sequence number	46	ct_dst_sport_ltm	Number of connections to the same destination IP and source port
22	dtcpb	Destination TCP base sequence number	47	ct_dst_src_ltm	Number of connections to the same source and destination IP
23	smeansz	Mean packet size transmitted by the source	48	attack_cat	Category of the attack
24	dmeansz	Mean packet size transmitted by the destination	49	Label	Binary label of the attack (0 for normal, 1 for attack)
25	trans_depth	Number of layers traversed in a single connection			

2.2 Federated Learning Framework

Federated Learning (FL) is a machine learning paradigm that allows models to be trained on multiple decentralized devices or servers, each holding local data samples and not modifying them. This framework is beneficial for applications where data privacy and security are paramount. FL is an evolving machine learning scheme that addresses the data island problem while preserving data privacy. Decentralized machine learning settings refer to multiple clients coordinated with one or more central servers. Google introduced it in 2016 to predict user text input on tens of thousands of Android devices while keeping the data on them [6].

Federated learning is a setup where multiple clients collaborate to solve machine learning problems under the coordination of a central aggregator. This setting also allows training data to be decentralized to ensure data privacy for each device. Federated learning depends on two main ideas: local computing and model transmission, reducing some of the systematic privacy risks and costs introduced by traditional centralized machine learning methods. The client's original data is stored locally and cannot be modified or moved. With the implementation of federated learning, each device uses local data for local training and then uploads the model to the server for collection. Finally, the server sends the model update to the participants to achieve the learning goal [5].

This approach offers several advantages:

Privacy Protection: FL reduces the risk of data disclosure and protects user privacy by ensuring sensitive data remains on local devices.

Enhanced Security: By keeping data distributed across multiple nodes, FL minimizes the risk of a single point of failure and reduces the attractiveness of targets for cyber attacks.

Scalability: FL leverages the computing power of multiple devices, making it easy to train large-scale models without the need for centralized data processing.

2.3 The Proposed Method

To prepare the training data for federated learning in the proposed model, it was divided into multiple subsets, each representing data held by different devices. A TensorFlow dataset was created for each device subset.

The model architecture in this study includes three essential layers:

Input Layer: This layer defines the shape of the input data, corresponding to the number of features in the dataset. It serves as the entry point for data into the neural network.

First Dense Layer: A fully connected layer with 10 neurons, where each neuron is connected to every input feature. It applies a linear transformation followed by a ReLU (Rectified Linear Unit) activation function, enabling the network to model complex patterns.

Second Dense Layer: Another fully connected layer with a single neuron. This layer outputs a value between 0 and 1 using a sigmoid activation function, making it suitable for binary classification tasks.

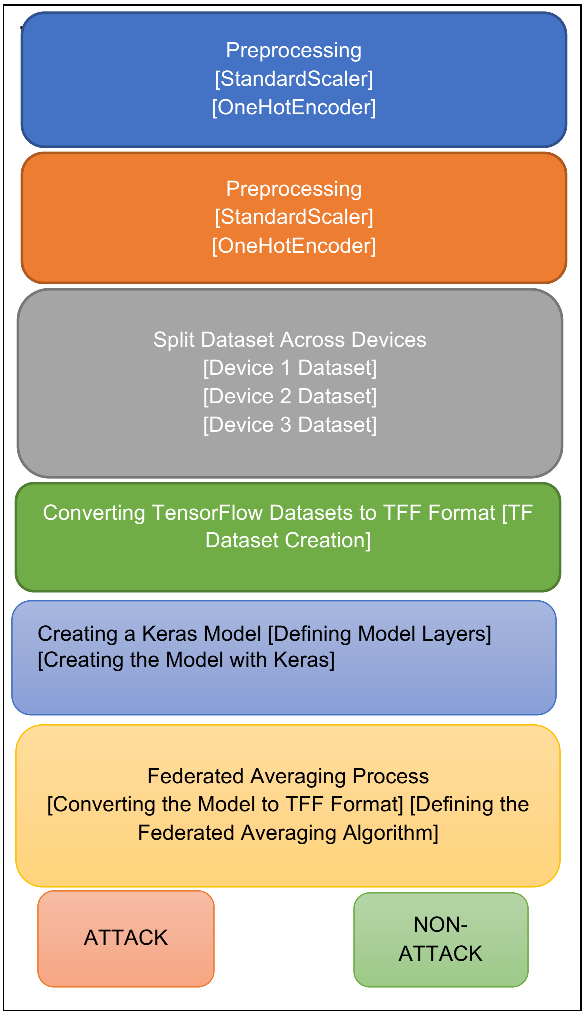
Preprocessing
[StandardScaler]
[OneHotEncoder]
Preprocessing
[StandardScaler]
[OneHotEncoder]
Split Dataset Across Devices
[Device 1 Dataset]
[Device 2 Dataset]
[Device 3 Dataset]
Converting TensorFlow Datasets to TFF Format [TF Dataset Creation]
Creating a Keras Model [Defining Model Layers]
[Creating the Model with Keras]
Federated Averaging Process
[Converting the Model to TFF Format] [Defining the Federated Averaging Algorithm]
ATTACK
NON-ATTACK

Figure 1 Architecture of the Proposed Model

The preprocessing step involved transforming both numeric and categorical features. Numeric properties were standardized using StandardScaler, and categorical properties were encoded using OneHotEncoder. A simple neural network was designed using TensorFlow Keras. The model consisted of an input layer, a hidden dense layer with ReLU activation, and an output dense layer with sigmoid activation. The federation average algorithm was used to train the model. An iterative process was created in which the global model was updated in multiple rounds of federation learning. The process model is shown in detail in Figure 1

3. Results and Discussion

The FL framework was implemented using the PySyft library. Each node in the FL setup was trained with a subset of the UNSW-NB15 training data. The model architecture included a simple neural network with two hidden layers. The training process was conducted over ten communication rounds, each consisting of local training followed by global model collection.

Specific hyperparameters were used to train the federated learning model. Table 2 states the following parameters during the training process.

Table 2 Parameters of FL

Batch Size	20
Num Devices	3
Learning Rate	0.02
Number of Rounds	10

These parameters have been carefully chosen to train the model efficiently and optimize its performance. The batch size value was determined as 20, which provided sufficient data to update the model at each training step. Model training was distributed using three devices, thus shortening the training time. The learning rate was determined to be 0.02, and the aim was to make balanced progress in the model's learning process. The training continued for ten rounds, and the model's performance was observed in each round.

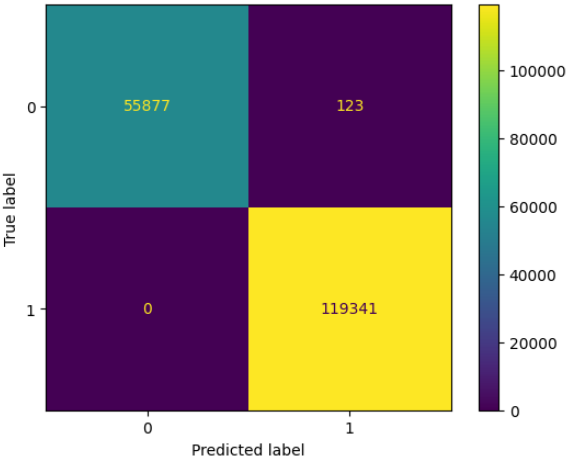

Figure 2 Confusion Matrix of FL

When the confusion matrix is examined in Figure 2 the following findings were derived:

- True Negatives (TN): The model correctly classified 55,877 negative examples as unfavorable. This indicates that the model can accurately describe average data.
- False Positives (FP): The model incorrectly classified 123 negative samples as positive. This indicates that the model perceives some average data as an attack.
- False Negatives (FN): The model has 0 false negative examples. This means the model correctly classified all positive examples and did not miss any attacks.
- True Positives (TP): The model correctly classified 119,341 positive examples as positive. This shows that the model can describe attack data quite successfully.

When Table 3 is examined the following conclusions were highlighted:

Table 3 Performance of the proposed model

Acc. (%)	Spec. (%)	Sens. (%)	Pre. (%)	F1 (%)
99.93	99.80	100	99.90	99.95

The model achieved an accuracy of 99.93%. The specificity of the model is 99.80%. Sensitivity (or recall) was 100%, meaning it correctly identified all positive samples in the test set. This excellent sensitivity score indicates that the model has no false negatives, making it highly reliable in detecting the presence of the target class. The accuracy of the model is 99.90%. The F1 score, the harmonic mean of precision and sensitivity, is 99.95%. This high F1 score reflects a balanced performance of the model, indicating that it maintains a high level of precision and sensitivity.

As shown in Table 4 and Figure 3, the performance of the proposed model was compared with other models from the literature, and the results were highly promising. The accuracy rates of other studies in the literature are as follows: [1] 82.14%, [2] 91.26%, [3] 99%, [4] 93.6%, [5] 90.95%. Our study's model exceeded the best results in the existing literature by reaching an accuracy rate of 99.93%.

Even the slight difference in our accuracy rate (about 0.93%) compared to work [3] demonstrates the superior performance of our model. This difference shows that significantly improved results can be achieved in real-world applications.

Table 4 Performance of relative studies

Researches	Acc (%)
Li at al. [1]	82.14
J. Sharma at al. [2]	91.26
H. Gharaee at al. [3]	99
T. Salman at al. [4]	93.6
C. Zhang at al. [5]	90.95
Our Study	99.93

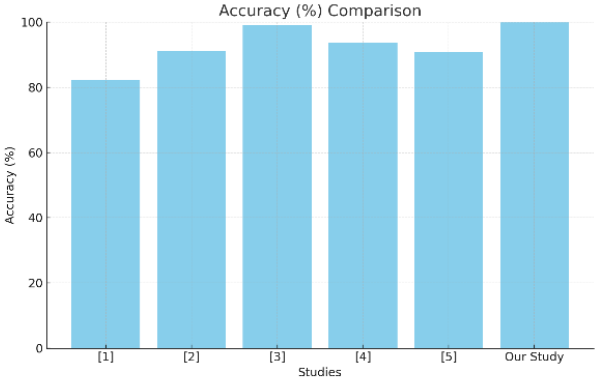

Figure 3 Accuracy of relative studies

4. Conclusion

As a result, our developed model has achieved a significantly higher accuracy rate than existing methods, making it a crucial reference point for future research in this domain. This remarkable success is a testament to the model's ability to generalize more effectively across the dataset, highlighting the robustness of the approach and the substantial improvements introduced during the training process. The enhancements made in model architecture, optimization techniques, and preprocessing steps have collectively contributed to this superior performance, underscoring the potential of our approach to set new benchmarks in the field.

Moreover, Federated Learning (FL) has emerged as a promising solution to the challenges associated with traditional centralized learning methods. FL allows decentralized model training across multiple nodes or devices, enabling data to remain on the local devices while only sharing model updates with a central server. This approach not only enhances data privacy by keeping sensitive information localized but also improves the scalability and adaptability of the model across diverse and distributed data environments. By leveraging FL, our model can harness multiple nodes' computational power and data diversity, leading to a more robust and generalized model that can perform effectively across various scenarios. This innovation addresses vital concerns in data security, privacy, and computational efficiency, paving the way for more secure and efficient machine learning applications in the future.

The combination of our model's high accuracy and the advantages of Federated Learning establishes a strong foundation for subsequent studies, suggesting that this approach could become a standard practice in related fields. The success of this model not only demonstrates the feasibility of using FL in complex scenarios but encourages further exploration and refinement of these techniques to tackle even more challenging problems. The positive outcomes from our study serve as an encouraging indicator that similar methodologies could yield comparable benefits in other applications, thereby broadening the impact and applicability of Federated Learning and advanced machine learning models.

5. Limitation and Future Work

Communication Overhead: FL introduces significant communication overhead, especially in scenarios with large numbers of participating devices or frequent model updates. Strategies to mitigate this overhead will be critical in the experimental setup.

Data Heterogeneity: The variability in data across different clients (i.e., non-iid data) can lead to challenges in model convergence and performance consistency. The study must account for these discrepancies when evaluating FL performance.

Bias and Fairness: Biases in local datasets can propagate through the FL model, leading to fairness issues. The study will analyze potential biases in the UNSW-NB15 dataset and their impact on FL performance.

Model Synchronization: Ensuring consistent model updates across decentralized nodes is challenging. The research will explore mechanisms for synchronizing model updates and maintaining consistency in a distributed setting.

Privacy-Preservation Trade-offs: While FL inherently provides privacy benefits, there is often a trade-off with model accuracy. The study will evaluate this trade-off and seek to quantify the impact of privacy-preserving measures on model performance.

6. Acknowledgement

The present paper does not include any research conducted by authors with human participants.

Ethical Approval

This manuscript does not contain any studies with human participants carried out by any of the authors.

Conflict of Interest

There is no conflict of interest for authorship.

REFERENCES

[1] **Shaoqin Li**, Zhendong Wang, Shuxin Yang, Xiao Luo, Daojing He, Sammy Chan, (2024), Internet of Things intrusion detection: Research and practice of NSENet and LSTM fusion models, Egyptian Informatics Journal, 26, 100476.

[2] **Sharma, J**., Giri, C., Granmo, O. C., & Goodwin, M. (2019). Multi-layer intrusion detection system with ExtraTrees feature selection, extreme learning machine ensemble, and softmax aggregation. EURASIP Journal on Information Security, 2019(1), 1-16.

[3] **H. Gharaee**, H. Hosseinvand, in 2016 8th International Symposium on Telecommunications (IST). A new feature selection is based on genetic algorithm and SVM, (2016), pp. 139–144.

[4] **T. Salman**, D. Bhamare, A. Erbad, R. Jain and M. Samaka, (2017), "Machine Learning for Anomaly Detection and Categorization in Multi-Cloud Environments," 2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud), New York, NY, USA, 2017, pp. 97-103.

[5] **Zhang, C**., Xie, Y., Bai, H., Yu, B., Li, W., & Gao, Y. (2021). A survey on federated learning. Knowledge-Based Systems, 216, 106775.

[6] **Li L**i, Yuxi Fan, Mike Tse, Kuo-Yi Lin, (2020), A review of applications in federated learning, Computers & Industrial Engineering, 149,106854.

[7] **Konečný, J.,** McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., & Bacon, D. (2016). Federated learning: Strategies for improving communication efficiency.

[8] **Tanyıldız, H.,** Batur Şahin, C., & Batur Dinler, Ö. (2024). Disrupting Downtime: Different Deep Learning Journeys into Predictive Maintenance Anomaly Detection. NATURENGS, 5(1), 47-53.

[9] **Tanyıldız, H.,** Batur Şahin, C., & Batur Dinler, Ö. (2024). Enhancing Cybersecurity through GAN-Augmented and Hybrid Feature Selection Machine Learning Models: A Case Study on EVSE Data. NATURENGS, 5(1), 61-70.

[10] **C. B. Şahin,** "DCW-RNN: Improving Class Level Metrics for Software Vulnerability Detection Using Artificial Immune System with Clock-Work Recurrent Neural Network," 2021 International Conference on Innovations in Intelligent Systems and Applications (INISTA), Kocaeli, Turkey, 2021, pp. 1-8.

[11] **Ulah, A.,** Aznaoui, H., Batur S¸ahin, C., Sadie, M., Dinler, O.: Cloud computing and 5G challenges and open issues. Int. J. Adv. Appl. Sci. (2022).

[12] **Ullah, A.,** Şahin, C. B., Dinler, O. B., Khan, M. H., & Aznaoui, H. (2021). Heart disease prediction using various machine learning approaches. Journal of Cardiovascular Disease Research, 12(3), 379–391.

[13] **Ozlem Batur Dinler,** Canan Batur Şahin, & Hanane Aznaoui. (2024). HYBRID MODEL USED FOR REDUCING LATENCY IN SMART HEALTHCARE SYSTEMS. Journal of Advancement in Computing, 2(1), 10–20.

[14] **Mikail Mohammed Salim,** Abir El Azzaoui, Xianjun Deng, Jong Hyuk Park, (2024). FL-CTIF: A federated learning based CTI framework based on information fusion for secure IIoT, Information Fusion, 102, 102074.

[15] **Parimala Boobalan,** Swarna Priya Ramu, Quoc-Viet Pham, Kapal Dev, Sharnil Pandya, Praveen Kumar Reddy Maddikunta, Thippa Reddy Gadekallu, Thien Huynh-The, (2022). Fusion of Federated Learning and Industrial Internet of Things: A survey, Computer Networks, 212,109048.

.