IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 10 January 2025, accepted 23 January 2025, date of publication 29 January 2025, date of current version 4 February 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3535952

RESEARCH ARTICLE

A Physics-Based Hyper Parameter Optimized Federated Multi-Layered Deep Learning Model for Intrusion Detection in IoT Networks

CHIRAG JITENDRA CHANDNANI, VEDIK AGARWAL, SHLOK CHETAN KULKARNI, ADITYA AREN, D. GERALDINE BESSIE AMALI, AND KATHIRAVAN SRINIVASAN, (Senior Member, IEEE)
School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu 632014, India
Corresponding author: D. Geraldine Bessie Amali (geraldine.amali@vit.ac.in)

ABSTRACT The Internet of Things (IoT) is reshaping our lives with its omnipresence. The sudden uptick in the ubiquitous nature of IoT devices ranging from fitness watches to aircraft has led to a surge of cyber-attacks. Artificial Intelligence powered Intrusion Detection Systems (IDS) are being used recently to combat this increasing surge of attacks in the IoT environment. However, existing solutions lack optimization for training in distributed decentralized environments. A popular solution for training a model in a decentralized environment is Federated Learning. Multiple client models collaboratively train a global model while keeping the individual client's data decentralized and private. This, however, suffers from poor generalization of the individual client data. This work proposes a new Federated Multi-Layered Deep-Learning (Fed-MLDL) model that employs physics-based hyperparameter optimization (HPO) technique FedRIME in a distributed federated learning environment for intrusion detection on the CICIoT23, CICIoT22, ToN_IoT, Edge_IIoT and, IoT-23 datasets. FedRIME ensures good generalization for all clients' data by finetuning the model's hyperparameters according to each client. The experimental results indicate that the Fed-MLDL with Fed-RIME optimization exhibits the highest accuracy for independent and identically distributed datasets with the scores being 99.2% with CICIoT23, 98.1% with CICIoT22, 98.2% with ToN_IoT, 98.5% with Edge_IIoTset and, 98.6% with IoT-23 dataset respectively. Further, the proposed Fed-MLDL with Fed-RIME optimization has demonstrated a significant improvement in the speed of convergence, stability, and client specific customization in federated learning. The study provides a comprehensive comparison with the most recent physics based HPO techniques. This study observes that coupling a Deep-Learning model with HPO techniques results in a much faster convergence requiring only 10-15 communication rounds. The proposed Fed-MLDL with Fed-RIME optimization outperforms existing state-of-the-art models on the CIC-IoT23 dataset.

INDEX TERMS Hyperparameter optimization, federated learning, deep learning, intrusion detection systems.

I. INTRODUCTION

Today's era is driven by internet connectivity. There are more than 5.45 billion internet users in the world and most users are prone to various types of security risks. These risks may often be underappreciated. However, it is estimated that cyber-attacks cost a staggering 10.5 trillion dollars annually and are projected to rise even further [1]. Various common types of cyber-attacks include Denial of Service (DoS) [2], Distributed Denial of Service (DDoS) [3], and, web spoofing [4]. The scale of these attacks necessitates sophisticated IDS solutions that can effectively protect systems and data.

IDSes can be classified into two major categories: signature-based and anomaly-based detection [5]. Anomaly based is further classified into state based and model based.

The associate editor coordinating the review of this manuscript and approving it for publication was Huaqing Li.

Signature-based IDS operate by utilizing a repository of established attack patterns (called signature database) to detect intrusions. Although this strategy is successful in dealing with familiar attacks, it is not as effective when adapting to new and developing intrusions. Anomaly-based IDS [6] monitors network traffic and system activity to detect deviations from established norms. State based IDS tries to generate an alarm when the network veers into an unsafe state modeled by the IDS. This is similar to the safety analysis in usage control models such as Pre UCONA with finite attribute domains [7] and PreUCONidA [8].On the other hand, model based IDS tries to model an unsafe network and tries to find the resemblance between the existing state of the model and the generated model. This method is highly adaptable, as it can identify previously unknown dangers by examining patterns and irregularities. Anomaly-based detection, however, suffers from false alarms, and misclassification of packets, thereby reducing its efficiency.

The capabilities of anomaly-based IDS have been significantly increased by recent breakthroughs in AI. These systems utilize various ML techniques such as supervised methods which include Support Vector Machines (SVM) [9], Decision Trees (DTs), Naive Bayes and unsupervised methods which include DBSCAN, K-Nearest Neighbours, K-Means [10]. They, however, are highly sensitive to changes in feature extraction techniques and selection requiring extensive effort for feature engineering. Deep learning techniques on the other hand show much lower sensitivity to features and thus, deep learning-powered IDS can constantly adjust and evolve in response to emerging threats, hence enhancing their ability to accurately detect and identify new and potential security breaches. Furthermore, the rise of specialised systems such as the Internet of Things (IoT) emphasizes the necessity for a strong and highly adaptable IDS.

Therefore, this article incorporates a deep learning-based approach to develop a highly accurate IDS model. Several researchers have made use of Artificial Neural Networks (ANNs) [11], Convolutional Neural Networks (CNNs) [12], and Recurrent Neural Networks (RNN) [13] for their IDSes. However, these suffer from a major issue which is data privacy. In today's modern world, data privacy remains one of the major issues for any system or institution. That being the case, acquiring and accessing data to train a model for IDS while preserving data privacy is a challenge. Storing this data locally for training also poses an issue, as it is easier for a malicious actor to gain access to a client due to its lower defence capabilities. Federated Learning (FL) [14] based approaches have recently gained attention as being a decentralized approach to solving the problem of data privacy. It's a client-server-based model where the clients are the end nodes and the end nodes are not required to share their sensitive data to the central model for training. Instead, the model trains locally on the client devices and after each communication round, aggregates their weights to update the global model. In such a scenario where data privacy is a global issue, Blockchain and decentralized approaches like FL have gained relevance in recent years in research as well as industry.

This study aimed to use FL for an IDS setup. In a traditional IDS setup the large volume of data that needs to be sent to the server for processing poses an issue, as, this may lead to significant bandwidth consumption and high latency as highlighted by [15]. This is especially important in the context of IoT devices which are resource constrained. FL also improves the scalability and personalization to data, especially in Non-IID tasks, where client models can capture unique patterns specific to their data. This also ensures the robustness of the global model. In addition to its advantages in mitigating communication overheads, FL-based systems also help protect sensitive information on the network as this data is never directly sent to the central server as shown by [16]. This is especially necessary in an IoT-based IDS scenario which deals with sensitive network data. Additionally, in distributed systems, there are a large number of IoT devices (sensors and actuators) in the network and each of these components can't identify the distinction between malicious requests and benign, and in order to ensure the security of data, we can utilize FL by training the data on the device itself or using FOG compute stations to train the models and send the parameters to the server.

Despite its many advantages, FL still has some issues. A major issue arises when working with non-IID data distributions. One of the most significant research areas in the field of FL is the use of Independent and identically distributed (IID) versus non-IID datasets [17]. In an IID dataset, the data samples are assumed to be independent from each other and drawn from the same probability distribution. Whereas in a Non-IID dataset, the data samples across users may have different statistical properties or may not be independent [18]. This more closely resembles a real-world data distribution. However, due to the difference in data distributions of each client device, the global model may generalise poorly to each individual device [19]. In most cases, this is due to a bad choice of hyperparameters which are not optimized for the devices they are used on. In order to maintain the efficacy of the FL technique in such cases various hyperparameter optimization (HPO) algorithms can be used. They ensure that each node in the FL architecture runs with the optimal set of hyparameters which leads to an overall increase in model performance and faster convergence [20]. This is especially important when the distribution of data is non-IID.

The article is structured as follows: In the following section, a comprehensive review of the literature is presented to understand the existing literature, followed by the proposed methodology where each and every aspect of the proposed model has been explained in detail. All of the results of the experimentation are compiled and compared under the results

and discussion section, followed then by the conclusion and future works.

Main Contributions of This Work:

- This work proposes a novel MLP-based IDS, namely, Federated Multi-Layered Deep Learning (FedMLDL) for an FL-based environment, to detect various types of intrusions based on packet information.
- The FedMLDL model is enhanced by employing the RIME HPO technique on individual clients, to increase the performance and convergence rate of the FedMLDL model on multiple datasets.
- The study also empirically proves, with extensive experimentation, the superiority of RIME as an HPO technique for the IDS use case against 4 other physics-based HPO techniques.
- The paper is the first to prove and thoroughly investigate the efficacy of the proposed approach with other physics-based HPO algorithms in an FL environment on the CICIoT'23 dataset [21].

II. RELATED WORKS

This section deals with the current research work in the fields relevant to the methodology of this article. First, this section reviews the past IDSes used, then it explores various works under optimization algorithms. Following that, then explore different works on FL.

A. MACHINE LEARNING AND DEEP LEARNING

ML has been playing a pivotal role in the applications of IDS, be it identifying malicious packets or intrusions within the system or filtering packets based on different data features. Wasnik and Chavhan [22] explore the use of DL-based IDS techniques to recognize complex network traffic patterns. Their proposed IDS achieves higher detection rates, lower false positive rates, and improved resilience against evasion strategies using this DL approach. Their IDS, however, may face difficulties with feature extraction and scalability. Kasongo [23] utilizes Recurrent Neural Networks like LSTM, GRU, and Simple RNN with XGBoost-based feature selection to detect network attacks. The study found that the XGBoost-LSTM model performed best on the NSL-KDD dataset with a test accuracy of 88.13%, while the XGBoost-Simple-RNN model performed the best on the UNSW-NB15 dataset with a test accuracy of 87.07%. However, the IDS network may have issues with detecting new types of attacks, especially, as feature size increases. Rosay et al. [24] have developed an MLP-based IDS for IoT environments. They have used the CICIDS2017 dataset for training and testing the model and have achieved an accuracy score of 99%. They have achieved the highest accuracy and have shown a detailed comparison with other related works. One of the shortcomings in the above literature is the preservation of privacy among IoT devices. As IoT devices don't have a dedicated software to differentiate between a malicious payload and a benign one, it can serve to be an entry point for attackers in the network. Thus this necessitates the need for FL in these scenarios. Arreche et al. [25] have proposed an end-to-end Explainable AI (XAI) framework for IDS. They have benchmarked seven black-box AI models, namely, RF, DNN, AdaBoost, MLP, K-Nearest Neighbours (KNN), SVM and, LightGBM. They have made use of XAI local explanations using Shapley Additive Explanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME) values. These are the most commonly used approaches for giving insights into the model's decision-making process, which enhances the model interpretability. Garcia and Blandon [26] has developed a unique DL-based IDS system named 'Dique', which is a software application utilized for detecting intrusions in real-time. They also state that DL is the most suitable technique for IDS tasks and the MLP based model utilized by them, as the backbone of their software also validates the same. The trained model was able to classify both known and unknown DoS attacks. The model gave an accuracy of 99.4% on the CICDDoS2019 dataset. One of the main shortcomings found in their work was the privacy of individual user data, which can only be made possible by the incorporation of FL for IDS-based tasks

B. HYPERPARAMETER OPTIMIZATION

HPO techniques are utilized to make sure that the hyperparameters being used are fine-tuned to the training environment. This becomes especially important in an FL use case, where training happens across various devices. There are a wide variety of HPO algorithms that exist today which specialise in different scenarios, both centralised and decentralised.

Agrawal et al. [27] have developed Genetic Clustered FL (CFL) which is evaluated on the MNIST dataset. The server model is initially trained on a subset of the MNIST dataset. This model is thereafter allocated to the clients according to the client ratio. The models' performance is assessed for 10, 15, and 30 clients. To ensure non-IID data and simulate real-time circumstances, each client device is assigned a random portion of the dataset. Their article reports achieving a 97.9% accuracy on the MNIST dataset and a 76.8% accuracy on the CIFAR-10 dataset. The client devices undergo training for two epochs, while the hyperparameters (learning rate and batch size) are optimized by genetic evolution. Kilichev and Kim [28] have used 1D-CNN for the creation of their Network Intrusion Detection System (NIDS), and have utilized Genetic Algorithm and Particle Swarm Optimization (PSO) to optimize 9 hyperparameters of their model (batch size, dropout rate, kernel size, learning rate, number of dense layers, number of epochs, number of filters, number of neurons and number of pooling layers). They have utilized the UNSW-NB15 dataset. This approach achieved 0.993 and 0.992 test accuracy respectively for both the techniques in a Non-FL based environment. Kundroo and Kim [20] have devised FedHPO, a revolutionary Federated optimization technique, that dynamically adjusts learning rate and epochs for each client during training.

TABLE 1. Comparative analysis of different techniques used on various datasets.

Technique	Dataset Name	Optimized FL	Federated Environment	Centralized
ANNs	NSL-KDD, UNSW-NB15	×	×	✓
RNNs with XGBoost	NSL-KDD, UNSW-NB15	×	×	✓
MLP	CICIDS2017	×	×	✓
ML-based XAI	RoEduNet-SIMARGL2021, CICIDS2017, NSL-KDD	×	×	✓
FedPSO	MNIST	×	✓	×
Deep Feed Forward Neural Network	CICDDoS2019	×	×	✓
FedGWO	CIFAR10	×	✓	×
Genetic-CFL	MNIST	×	✓	×
GA-1D-CNN, PSO-1D-CNN	UNSW-NB15	×	×	✓
FedHPO	FashionMNIST	×	✓	×
FL & ANN	ToN_IoT, CICIDS2017	×	✓	×
DNN & FL	CICIoT23	×	✓	×
DAFL	CIC-IDS2018	×	✓	×
Fed-MLDL	**CICIoT23, ToN_IoT, IoT23, CICIoT22, Edge_IIoTset,**	✓	✓	×

TABLE 2. Summary of symbols used.

Symbol	Description
$P(i)$	Probability of each device being selected
ω_0	Initial model weights/parameters
q	Number of clients selected in each round
R_i	Subset of clients from the total, selected in the i^{th} training round
μ	Momentum
λ	Weight decay
t	Current iteration number in the algorithm
T	Maximum number of iterations for the algorithm
E	Coefficient of adherence
$\tilde{x}$	Synthetic instance created by SMOTE
S_i	Fitness score of the i-th rime agent, used for normalization and selection.
x_i	Instance of minority class
x'	Normalized value of the original data point (x_{org})
κ	random value between 0 and 1 used by SMOTE
x_{zi}	neighbor of the minority class instance
x_{org}	original feature value
x_{min}	min value of feature
x_{max}	max value of feature
z	Standardized value of the original data point(x_{org})
χ	mean value of feature
τ	standard deviation of feature
$\mathcal{K}$	Total number of clients
η	Learning rate
$\mathcal{B}$	Local minibatch size
p	Fraction of clients available in each round
$\mathcal{P}_k$	Random distribution of data to clients
ω	Model weights/parameters
n	Total number of data points
$n(i)$	Number of data points in the data of the i^{th} client
$sl(\omega)$	Server loss in each communication round
$L(i)$	Objective function of the i^{th} client device
TP	True positives - Number of positive instances classified correctly
TN	True negatives - Number of negative instances classified correctly
FP	False positives - Number of instances classified incorrectly as positive
FN	False negatives - Number of instances classified incorrectly as negative

Dynamic tweaking accelerates model convergence without adding client or server complexity. Here they have used the 2D-CNN model on the FashionMNIST dataset for classification and have compared their approach to other algorithms.

Park et al. [29] in their study, presented a new FL technique called FedPSO that uses Particle Swarm Optimization to lower communication costs and improve model accuracy in unstable network settings. FedPSO greatly lowers data transmission amounts by sending score values rather than big weights, which in unstable network trials results in an average accuracy improvement of 9.47% and a 4% reduction in accuracy loss. Abasi et al. [30] have proposed a FedGWO, FL approach using the Grey Wolf optimizer algorithm for hyperparameter tuning. Through modification of the data format used in client-server communication based on the Grey Wolf Optimizer (GWO), it seeks to improve the efficiency of the global model in FL. They have used the CIFAR-10 dataset and the proposed model outperforms FedAvg and FedPSO by reducing accuracy loss on unstable networks by obtaining an average improvement in the accuracy of the global model of 13.55% while lowering data capacity for network communication.

1) PHYSICS-BASED HPO TECHNIQUES

A specific class of HPO algorithms are the physics-based HPO algorithms which take inspiration from the laws of physics and natural phenomena. This study mainly focuses on these HPO algorithms, and utilizes the following for comparison. Azizi et al. [31] introduced the Energy Valley Optimization (EVO) algorithm. It is based on the concept of the existence of stable energy levels where energy is minimum which is similar to an optimal solution in a search space. To make use of the understanding of this concept, the algorithm starts by initializing agents in the solution space with varying levels of stability. Solutions are then updated based on three factors- enrichment bound, position vector, and stability level. Shehadeh [32] proposed the Chernobyl Disaster Optimizer (CDO) algorithm which uses the patterns of movement of alpha, beta, and gamma particles as the basis for its algorithm. To mimic the movement of radioactive particles, the optimizer simulates the movement of such particles from a high-pressure(explosion point) to a low-pressure(human standing point). The algorithm takes into account 2 factors namely, Particle propagation area calculation and Gradient descent factor when searching for solutions. Hashim et al. [33] presented the Fick's Law Algorithm (FLA) which is a physics-based metaheuristic algorithm inspired by Fick's law of diffusion. It includes three stages, namely the diffusion phase (exploration), the equilibrium phase (transfer phase from exploration and exploitation), and the steady state phase (exploitation). FLA utilizes these concepts to find the optimal solution in our search space. Thieu [34] introduced the Tug of War Optimizer (TWO) which takes inspiration from the game of Tug of War where two opposite forces pull a rope towards themselves. The algorithm starts by initializing a population that consists of teams. The weights of teams are then calculated and compared with other teams to move them accordingly. This process is repeated for several generations. Su et al. [35] proposed the RIME optimization algorithm. RIME is an algorithm that takes inspiration from the natural process of rime ice formation, which takes place as super-cooled water droplets freeze onto surfaces. The RIME algorithm consists of three primary strategies. The first strategy is the Soft-Rime Search Strategy. This strategy is utilized when soft rime particles are present when there are no strong external factors like gusts of wind affecting rime ice formation. They, therefore, form a feathery layer and are known to be less structured in formation. This strategy helps balance between large-scale exploration and small-scale exploitation. Another strategy utilized by the algorithm is the Hard-Rime Puncture Mechanism. Hard rime particles are more organized in their formation. They are formed when strong external environmental factors are present such as a strong wind as they require more directional flow to form. Soft rime particles can eventually turn into hard rime particles given the appropriate environmental conditions. Finally, the RIME algorithm utilizes the Positive Greedy Approach to improve upon the greedy selection mechanism. The positive greedy approach weeds out the inferior solutions and regularly introduces sub-optimal solutions to actively avoid local optima.

C. FEDERATED LEARNING

FL is a promising decentralized method that allows ML models to be trained collaboratively across several devices or organizations without requiring the exchange of raw data. Reviewing existing approaches and their applicability, this overview of the literature investigates the junction of FL and IDSes. Lazzarini et al. [36] proposed using a shallow ANN with FedAvg as the aggregation algorithm. The article investigates FL as an alternative to centralized ML for IoT intrusion detection. Experiments on the Ton_IoT and CICIDS2017 datasets show that FL surpasses centralized methods in terms of accuracy, precision, recall, and F1-score. Furthermore, alternative aggregation algorithms such as FedAvgM, FedAdam, and FedAdagrad were assessed, with FedAvg and FedAvgM performing better.

Abbas et al. [37] approach includes FL with IDS in an IoT environment using the CICIoT23 dataset. It helps detect attacks such as spoofing, DDoS, and jamming. It achieves accuracy and loss close to normal centralized learning. They first used various data preprocessing techniques to normalize and remove unnecessary features, then upsample the dataset using SMOTE (Synthetic Minority Over-sampling Technique) and used a deep neural network (DNN) architecture for modeling. Friha et al. [38] have proposed an FL-based Intrusion Detection System (FELIDS) for agricultural IoT infrastructure security. The solution protects Agricultural IoTs by using three deep-learning classifiers: DNNs, CNNs, and RNNs. The work compares the performance of FELIDS with traditional centralized ML models to achieve high attack detection accuracy and secure the privacy of IoT device data. Li et al. [39] propose a Distributed Adaptive FL (DAFL) approach. Here the dynamic aggregation at the server side ensures that the system can improve its overall

performance and can adjust to changing data characteristics over time. DAFL performs competitively or better in metrics like accuracy, precision, recall, and F1-score when compared with conventional and advanced FL systems in this work. The efficiency of the DAFL scheme is demonstrated by experimental results for the centralized scheme and FL-NIDS based on the FedAvg algorithm under various scenarios. It reduces the overhead in the FedAvg algorithm by 33%. The above literature survey is compiled in Table 1.

Based on the existing literature, the article proposes an MLP based IDS named FedMLDL, in an FL environment, enhanced with RIME physics based HPO technique. By extensive experimentation, the work has demonstrated an increase in the performance and convergence speed of FedMLDL using RIME on various IoT based datasets. The subsequent sections explain the same.

III. METHODOLOGY

This section will first give a brief introduction to the dataset used in the study (CICIoT'23) along with other datasets utilized for comparison. In addition, it will explore the proposed FedMLDL model and its enhancement using FedRIME. All symbols used in this work have been defined in Table 2.

A. DATASET DESCRIPTION

The proposed methodology makes use of the open-source CICIoT'23 dataset for training and validation on the classification task. A major factor in selecting this dataset for this study is its relatively recent release. Additionally, no existing work has applied physics-based HPO techniques in a federated learning environment to this dataset. The dataset contains more than 40 million rows. It contains information about 33 attacks in an IoT network [21]. We use 70-30 train test split for our model as this is optimal [40]. This is then distributed to individual clients. This sampling for each client from the training dataset is done in accordance with both IID and non IID scenarios.

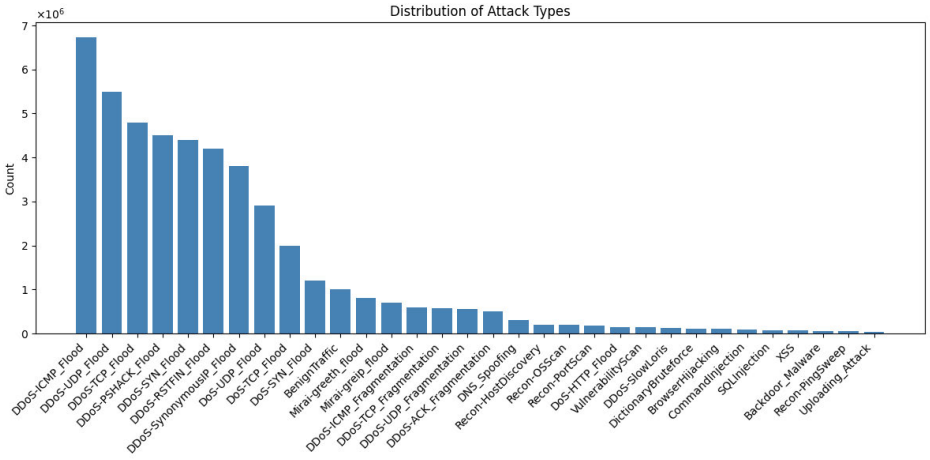
Other publicly available datasets utilized for comparison in our study were selected based on their release date, the more recent the dataset, more up-to-date it is with the attack types and their traffic capture techniques, utilizing IoT based networks. We make use of CICIoT22 which is a generation previous to CICIoT23 discussed above. It was released in the year 2022 and they collected a data from a network of 60 IoT devices in four stages, specifically, powered on, idle, active, and interaction. The dataset consists of two attack types, Flood DoS attack and RTSP brute-force attack. Additionally, the ToN_IoT dataset released by UNSW Sydney, is another widely used IoT based Intrusion Detection dataset. This dataset contains both benign packets and more than 25 attack types such as DoS, DDoS and botnet activities, against web applications, IoT gateways and computer systems across the IoT/IIoT network. It supports a range of deep learning models, including CNNs other deep learning based models, which have been shown to perform well when trained on this data for cybersecurity applications [41]. The Edge-IIoTset dataset is another dataset which contains packets collect in a industrial IoT based environment, with more than 10 types of IoT devices utilized in the whole network. The authors have claimed that this dataset can be used for both centralized machine learning based tasks and federated learning tasks [42]. This dataset consists of more than 60 features and threats classified into five categories, DoS/DDoS attacks, Information gathering, Man in the middle attacks, Injection attacks, and Malware attacks. The Aposemat IoT23 dataset is another cybersecurity analysis dataset released in the year 2020 by The Stratosphere labs and Avast Software, Prague. It contains 20 malware captures executed using IoT devices, and 3 captures for benign IoT devices traffic. It contains attack types such as, Command & Control (C&C), Mirai, DDoS, File Download, Okiru and Port Scan. Data distributions of all the datasets mentioned above is shown in Figure 1.

Data imbalance is one of the most common issues in ML tasks, wherein certain categories are under-represented compared to other classes. A skewed dataset can lead to a bias in the model towards one class or another. The data imbalance problem in the datasets is alleviated with the help of SMOTE [43]. SMOTE is a well-known technique for resolving class imbalance in various datasets and is used by numerous data scientists worldwide.

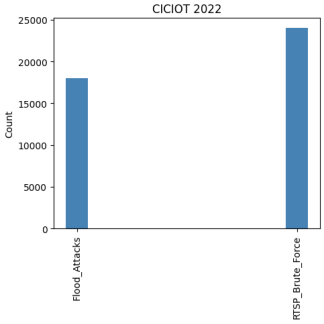
SMOTE uses neighbors of a minority class instance to generate a new synthetic instance of the class. To achieve this, the algorithm interpolates the minority class instance x_i with one of its neighbors x_{zi} to create a new synthetic instance $\tilde{x}$ as shown in equation 1.

Feature scaling is particularly important while training deep learning algorithms as having features with varying magnitudes and ranges will cause different step sizes for each feature, additionally, a lack of feature scaling can result in a lack of convergence during the training. In this work, two methods of scaling the dataset have been incorporated (1) Normalization and (2) Standardization. Normalization is the process of scaling the features in the range [0-1] and standardization is the process of scaling features to a mean of zero and standard deviation of 1. The equations for normalization and standardization are shown in 2 and 3 respectively.

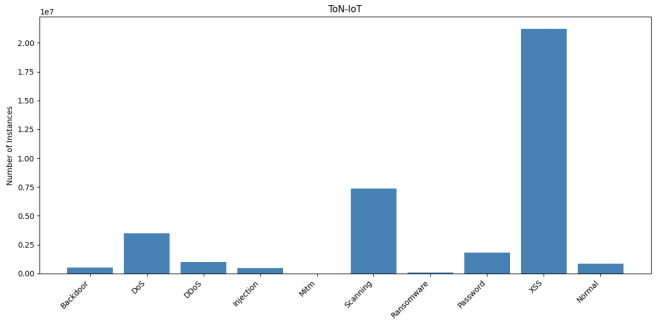
To gain a better understanding of the relationships between different variables, and to aid in reducing the dimensionality of the data, a Pearson correlation matrix representation with all the columns present in the dataset corresponding to attack types is shown in Figure 2. and a table encompassing all the statistical parameters of the dataset is shown in Table 3. Some of the statistical parameters used are Mean, Variance, Range and Inter-Quartile range, which gives us insight into the data distribution and variation. Pearson correlation helps us to analyze and study the relationship between various features in a particular dataset and identify features which are highly correlated. These highly correlated features can then be combined into one composite feature or removed, thus



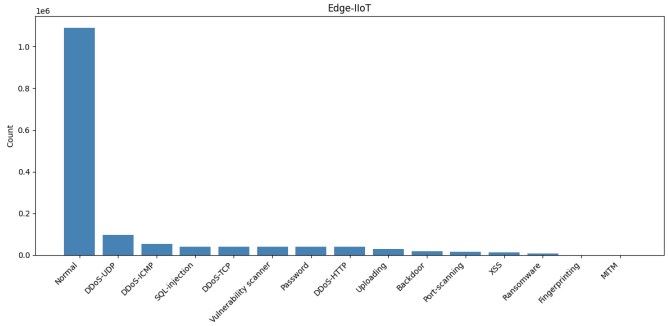
(a) Data distribution for CICIoT23 dataset



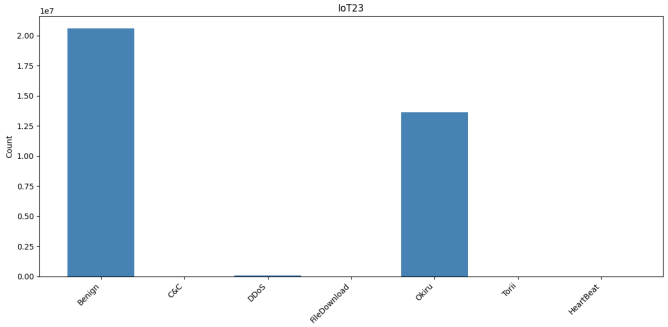
(b) Data distribution for CICIoT22 dataset



(c) Data distribution for ToN-IoT dataset



(d) Data distribution for Edge-IIoTset dataset



(e) Data distribution for IoT-23 dataset

FIGURE 1. **Comparison of data distributions for the datasets used for comparative analysis.**

reducing the input features and aiding in convergence of the model. For instance, in the above dataset, we have removed redundant features fin_flag_number, ack_count, urg_count, using the correlation matrix for analysis. This helps in reducing the curse of dimensionality problem, especially in a dataset like CICIoT'23, which has 40+ features, more than 40 million rows.

$$\tilde{x} = x_i + \kappa \times (x_{zi} - x_i) \tag{1}$$

$$x' = \frac{x_{\text{org}} - x_{\min}}{x_{\max} - x_{\min}} \tag{2}$$

$$z = \frac{x_{\text{org}} - \chi}{\tau} \tag{3}$$

B. PROPOSED FedMLDL ENHANCED WITH FEDRIME

FL-based techniques work on several client devices which perform local training and then aggregate the weights obtained by each of the models to update the global model. These updated weights are then redistributed to all client devices. All these steps are performed in one communication cycle. The objective of each of these rounds is to minimize the objective function as given in 4.

$$\min_{\omega} sl(\omega) = \sum_{i=0}^{n} P(i) \cdot L(i) \tag{4}$$

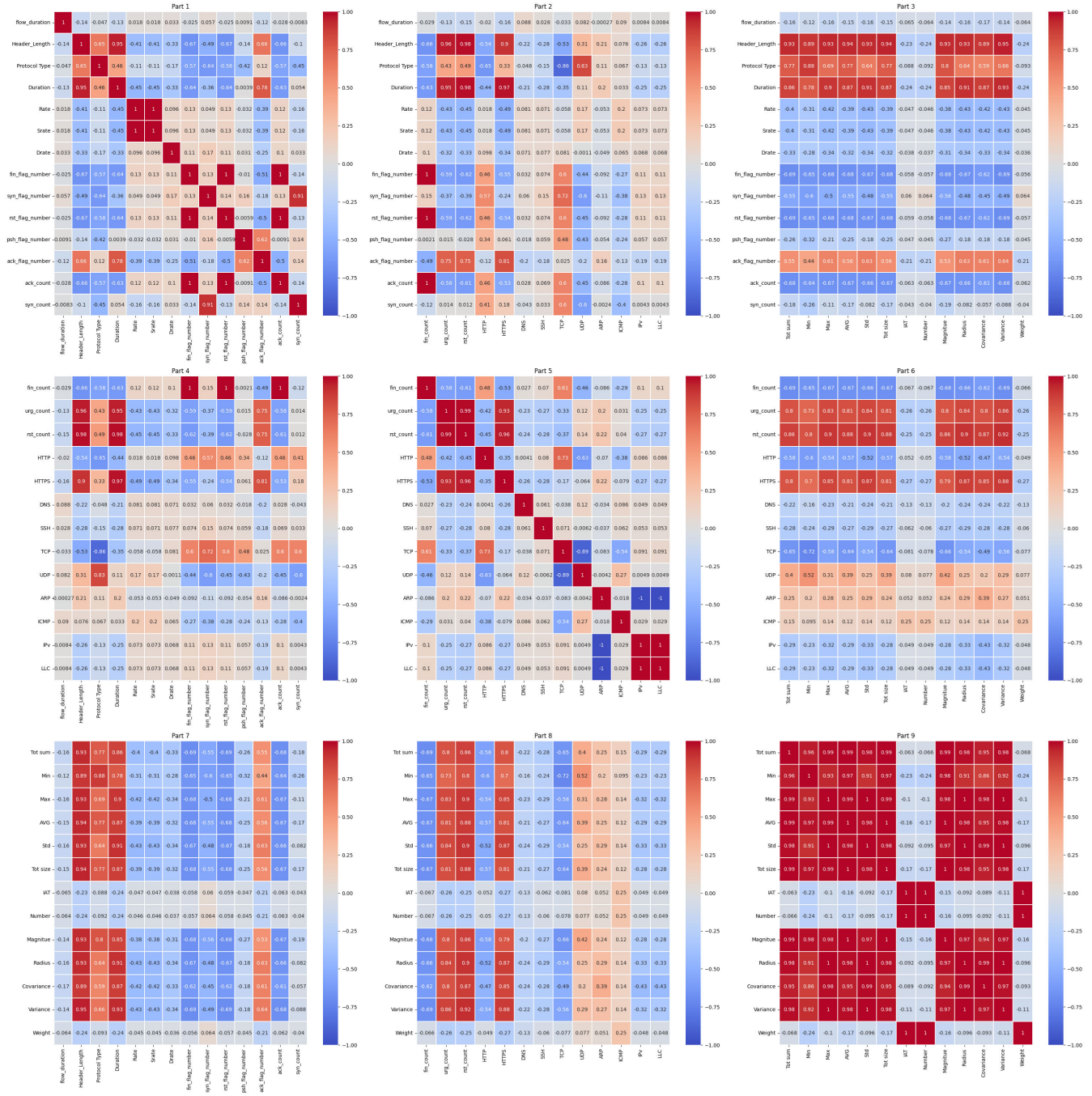


FIGURE 2. Pearson correlation matrix depicting relationships between various features of the dataset.

Here, $P(i)$ may be set manually or may be derived using equation 5

$$P(i) = \frac{n(i)}{n} \tag{5}$$

Here, $n = \sum_{i=0}^{k} n(i)$. The objective function for each client is calculated on its data at the end of each training epoch. The optimization of this loss itself is done using Stochastic Gradient Descent (SGD). The model parameters for each client are then sent to the server for aggregation at the end of a communication cycle.

For training in an FL environment, an aggregation strategy must be utilized to aggregate client weights for each communication round. The proposed FedMLDL model makes use of FedAvg for this task. FedAvg is a popular aggregation strategy utilized in FL applications. It is shown to be an improvement over the previously popular FedSGD as shown in [14]. A major difference between FedSGD and

TABLE 3. Statistical parameters for each of the features utilized in the training of the model in an FL environment.

Column	Mean	Variance	Range	IQR
Protocol Type	9.0096	78.5661	47.0000	7.2000
Duration	66.3505	194.3708	255.0000	0.0000
Drate	0.0000	10.4376	0.0476	0.5421
syn_flag_number	0.2077	0.1646	1.0000	0.0000
psh_flag_number	0.0887	17.2381	1.0000	0.4876
ack_flag_number	0.1247	0.1091	1.0000	0.0000
ece_flag_number	0.0000	23.8572	2.1000	0.7234
cwr_flag_number	0.0000	0.0000	3.7000	0.0000
syn_count	0.3318	0.4423	6.3700	0.0600
fin_count	0.0984	5.4789	17.0000	0.8234
HTTP	0.0478	0.0455	1.0000	1.1200
HTTPS	0.0554	0.0524	1.0000	0.2345
DNS	0.0002	0.0002	1.0000	0.0000
Telnet	0.0000	14.1874	0.0000	0.3951
TCP	0.5748	0.2444	1.0000	1.0000
UDP	0.2112	0.1666	1.0000	0.0000
DHCP	0.0000	20.1235	1.0000	0.5672
ARP	0.0000	0.0000	1.0000	0.0000
ICMP	0.1647	0.1376	1.0000	0.0000
IPv	0.9999	15.8824	1.0000	0.1325
Min	91.1736	19635.4100	3224.0000	4.0000
Max	181.7700	274154.9000	26522.4000	5.2000
AVG	124.1468	58077.3400	8147.7970	4.0452
Std	33.3219	25690.3600	9039.4930	0.3647
Tot size	124.3736	58633.5600	5530.0000	4.0600
Number	9.5011	0.6769	12.5000	0.0000
Magnitue	13.0960	74.0772	116.3554	0.3962
Radius	47.0798	51373.0000	12783.7700	0.5059
Variance	0.0965	0.0544	1.0000	0.0800
Weight	141.5795	447.7223	243.6000	0.0000
label	7.0269	30.7572	33.0000	7.0000

Algorithm 1 Fed-MLDL Algorithm

Fed-MLDL Algorithm (Server Side)

ω_0 is initialized
for each training round $i = 0, 1, 2, 3, \ldots$ **do**
  $q \leftarrow \max(\lfloor \mathcal{K} \cdot p \rfloor, 1)$
  $R_i =$ random set of q clients
  for each client $k \in R_i$ in parallel **do**
    $\omega_{i+1}^k, \eta_k, \mu_k, \lambda_k \leftarrow \text{ClientUpdate}(\omega_i, k)$
  end for
  $\omega_{i+1} \leftarrow \sum_{k=1}^{\mathcal{K}} \frac{n_k}{n} \omega_{i+1}^k$
end for

ClientUpdate Function

function ClientUpdate(ω, k)
  $\mathcal{B} \leftarrow$ Split $\mathcal{P}_k$ into batches of size $\mathcal{B}$
  $(\eta, \mu, \lambda) \leftarrow$ RIME() ▷ Call HPO using RIME
  for each local epoch i from 1 to $\mathcal{E}$ **do**
    for each batch $b \in \mathcal{B}$ **do**
      $\omega \leftarrow \omega - \eta(\nabla \ell(\omega; b) + \lambda\omega) + \mu m$
      $m \leftarrow \mu m + (1 - \mu)\nabla \ell(\omega; b)$
    end for
  end for
  return $\omega, \eta, \mu, \lambda$
end function

RIME Algorithm

function RIME
  Initialize the rime population R
  Get the current optimal agent and optimal fitness
  $t \leftarrow 0$
  while $t \leq T$ **do**
    Compute coefficient of adherence $E = (t/T)^{0.5}$
    if $r_2 < E$ **then**
      Update rime agent location by the **soft-rime search strategy**
    end if
    if $r_3 <$ NormalizeFitness(S_i) **then**
      Cross update agents using the **hard-rime puncture mechanism**
    end if
    if $F(R_i^{\text{new}}) < F(R_i)$ **then**
      Replace suboptimal solution using **positive greedy selection mechanism**
    end if
    $t \leftarrow t + 1$
  end while
  return updated η, μ, and λ
end function

FedAvg is that in FedAvg you can perform multiple rounds of training on each client before you send an update to the server. This results in more stable updates and reduces the number of communication rounds required. The algorithm

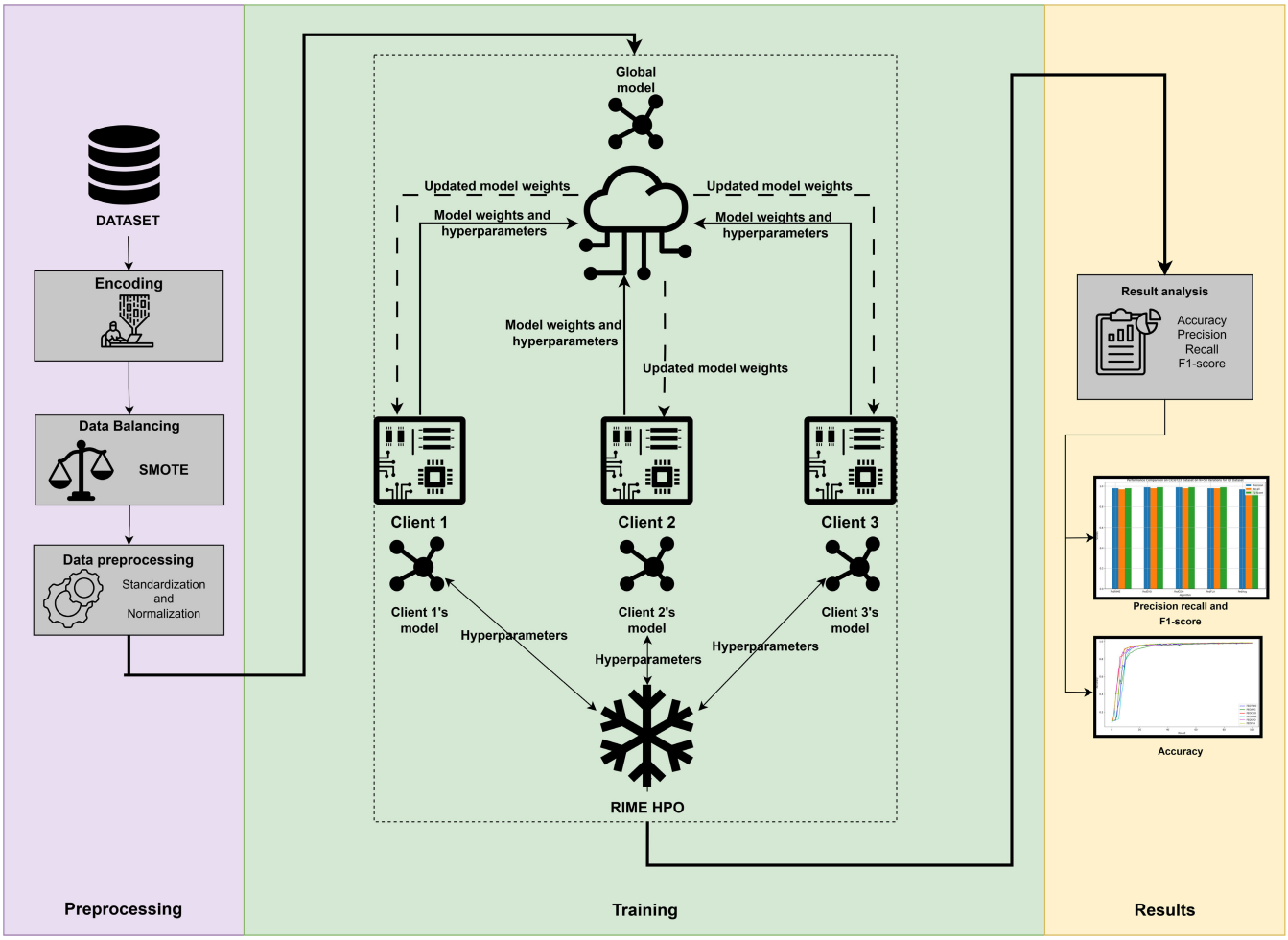


FIGURE 3. FedMLDL model methodology with RIME.

starts by randomly initializing weights on the server. Then for a predefined number of communication rounds, it will update the server. First for each communication round random clients must be selected which will participate in training. Now, for each client, training will occur using minibatches and for a certain number of epochs using SGD as an optimizer. The server then aggregates all the weights and performs an update. This is continued till convergence or the stipulated number of communication cycles is reached.

In the proposed FEDMLDL model, this article uses a distributed structure of clients and servers, in line with federated learning, where the model for each client and the central server is exactly the same in architecture. Keeping in mind the FL principles, this article assumes that each of these clients are edge devices that may not be available at all times and the model must be capable of carrying on without all of them being active. To achieve this, each client is provided different data but, the same starting weights, which are copies of the global model weights. Data is preprocessed and upsampled using the SMOTE technique to deal with class imbalance. Inside each of these clients, a Multilayer Perceptron (MLP) based encoder model is initialized using the Xavier Uniform Initialization, which is the default way of initializing weights in Pytorch 2.3.1.

In each client, the model makes use of a sequential neural network of fully connected layers to effectively handle multidimensional input data. It begins with an input size equal to the number of input features. The model begins by mapping this data to a 64-dimensional space using a linear layer which is followed by a batch normalization layer to stabilize activations. After this, a LeakyReLU layer is utilized to introduce some non-linearity. This is followed by a linear layer which maps the data from 64 to 32 dimensions followed by a Batch normalization and LeakyReLU layer as before. The data is then mapped to a 16-dimensional space using a linear layer. Finally, the data is passed through either a sigmoid layer for binary classification or a Softmax layer for multi-class classification to provide class probabilities, which are then utilized for the classification task. This architecture is well suited for robust feature extraction and handling complex data effectively.

The novelty of the proposed model is the integration of FL and physics-based RIME HPO technique. At the beginning of each communication cycle, a population is initialized for each client. Each of these populations has its own unique value for the following hyperparameters: learning rate, momentum, and weight decay. The initialization values of these hyper-parameters along with their updates are handled by RIME and they are updated for a fixed number of times as decided by the number of tuning epochs which can be set by the user. The best set of hyper-parameters are identified at the end of each communication round and are sent to the server along with weights for aggregation according to the FedAvg strategy. The Fed-MLDL with FedRIME algorithm is provided in algorithm 1. The entire process of the proposed methodology is summarised in Figure 3.

The algorithm deals with client updates, server updates and the HPO algorithm used which, in this case, is RIME. At the server side, the weights are at first initialized. Then, for each training/communication round, a random set of q clients is chosen(R_i). For each client in that round, we do an update of weights using the **ClientUpdate Function**. In the **ClientUpdate function**, we use minibatches of size $\mathcal{B}$ and acquire our learning rate(η), weight decay(λ) and momentum(μ) using the provided RIME function. These hyperparameters and data are then used to train the model using gradient descent with momentum for a number of epochs. After training is complete, the client returns its updated weights and chosen hyperparameters to the server. The server, upon receiving all the client's weights, updates the global weights using a weighted average.

This article compares four other physics-based HPO techniques in addition to RIME with the proposed model and compares their performance on the dataset as discussed in the section I.

The model has been implemented using the Flower (version 2.0.0.1) framework [44] which was chosen due to its flexibility and robustness. Additionally, MealPy [45] was also utilized to simulate the HPO techniques.

IV. RESULTS

In this section, the results of the experiment are presented. This section evaluates the result by using various quantitative factors such as loss, precision, recall, and F1-Score. These metrics are explained in IV-A. Overall the proposed Fed-MLDL with Fed-RIME optimization gives the highest accuracy of 99.7% for 2 classes, 99.5% for 8 classes, and 99.3% for all 34 attack types for IID distributed dataset, and Fed-MLDL with Fed-FLA gives the highest accuracy of 99.4% for 2 class, 99.3% for 7 class and 99.1% for all 34 attack type for Non-IID distributed dataset.

A. METRICS

In machine learning and deep learning model evaluation, some common metrics are used to evaluate the model's effectiveness. This article makes use of accuracy, recall, precision, and F1-score. Each of these metrics examines a unique aspect of the model's performance.

1) ACCURACY

Accuracy represents the ratio of correctly predicted values to the total number of instances. It is calculated according to equation 6

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

2) RECALL

Recall measures how frequently a model identifies a positive instance from the pool of all positive instances. It is calculated according to equation 7.

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

3) PRECISION

Precision is a measure of how often the model's predicted positive instances are correct. It is calculated according to equation 8

$$\text{Precision} = \frac{TP}{TP + FP} \tag{8}$$

4) F1-SCORE

The F1-score is the harmonic mean of precision and recall and represents the balance between precision and recall. It is calculated as shown in equation 9

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{9}$$

Figure 4 and, Figure 11 show the accuracy of the aggregation algorithm in different optimization techniques in a non-IID Distribution over 100 communication rounds for 100 clients in binary and 34 classes respectively. These figures show that on non-IID data sets, all optimization-based algorithms (FedTWO, FedCDO, FedRIME, FedEVO, and FedFLA) demonstrate superior performance compared to FedAvg. FedAvg shows slower convergence, taking about 40 rounds to reach high accuracy levels, which indicates its inefficiency in rapid learning scenarios. In contrast, FedTWO, FedRIME, and FedFLA demonstrate rapid convergence thereby stabilizing within the first 10 rounds and achieving high accuracy. FedCDO and FedEVO converge faster than FedAvg but slower than FedTWO, FedRIME, and FedFLA taking about 15 rounds to stabilize. In terms of stability, FedTWO, FedRIME, and FedFLA exhibit minimal fluctuations once they reach high accuracy, indicating high stability while FedCDO and FedEVO show some fluctuations before stabilizing. Figure 5 and, Figure 8 shows the accuracy of the aggregation algorithm in different optimization techniques in an IID distribution. FedTWO, FedCDO, FedRIME, FedEVO, and FedFLA show rapid convergence, reaching over 80% accuracy within the first 10-15 rounds and stabilizing close to 100% accuracy shortly after. FedAvg, on the other hand, shows a significantly slower

TABLE 4. Performance comparison of Fed-MLDL on different hyper-parameter optimization algorithms in IID data distribution for binary classification.

Model Used	HPO Algorithm	Accuracy	Precision	Recall	F1-Score
Fed-MLDL	**FedRIME**	**0.997**	**0.994**	**0.995**	**0.995**
Fed-MLDL	FedEVO	0.993	0.992	0.992	0.993
Fed-MLDL	FedCDO	0.992	0.992	0.991	0.992
Fed-MLDL	FedFLA	0.994	0.993	0.993	0.994
Fed-MLDL	FedTWO	0.991	0.991	0.990	0.993
Fed-MLDL	None	0.978	0.977	0.974	0.979

TABLE 5. Performance comparison of Fed-MLDL on different hyper-parameter optimization algorithms in IID data distribution for 8 class classification.

Model Used	HPO Algorithm	Accuracy	Precision	Recall	F1-Score
Fed-MLDL	**FedRIME**	**0.995**	**0.993**	**0.994**	**0.994**
Fed-MLDL	FedEVO	0.994	0.991	0.993	0.992
Fed-MLDL	FedCDO	0.993	0.990	0.992	0.993
Fed-MLDL	FedFLA	0.991	0.992	0.991	0.990
Fed-MLDL	FedTWO	0.992	0.991	0.993	0.991
Fed-MLDL	None	0.975	0.974	0.973	0.976

TABLE 6. Performance comparison of Fed-MLDL on different hyper-parameter optimization algorithms in IID data distribution for 34 class classification.

Model Used	HPO Algorithm	Accuracy	Precision	Recall	F1-Score
Fed-MLDL	**FedRIME**	**0.993**	**0.992**	**0.993**	**0.992**
Fed-MLDL	FedEVO	0.993	0.993	0.994	0.992
Fed-MLDL	FedCDO	0.991	0.990	0.992	0.992
Fed-MLDL	FedFLA	0.990	0.991	0.991	0.989
-Fed-MLDL	FedTWO	0.991	0.990	0.992	0.990
Fed-MLDL	None	0.971	0.972	0.971	0.974

TABLE 7. Performance comparison of Fed-MLDL on different hyper-parameter optimization algorithms in Non-IID data distribution for binary classification.

Model Used	HPO Algorithm	Accuracy	Precision	Recall	F1-Score
Fed-MLDL	FedRIME	0.993	0.993	0.991	0.993
Fed-MLDL	FedEVO	0.991	0.992	0.993	0.991
Fed-MLDL	FedCDO	0.990	0.991	0.994	0.992
Fed-MLDL	**FedFLA**	**0.994**	**0.995**	**0.994**	**0.995**
Fed-MLDL	FedTWO	0.992	0.993	0.992	0.994
Fed-MLDL	None	0.972	0.973	0.974	0.975

TABLE 8. Performance comparison of Fed-MLDL on different hyper-parameter optimization algorithms in Non-IID data distribution for 8 class classification.

Model Used	HPO Algorithm	Accuracy	Precision	Recall	F1-Score
Fed-MLDL	FedRIME	0.992	0.991	0.992	0.992
Fed-MLDL	FedEVO	0.990	0.992	0.992	0.993
Fed-MLDL	FedCDO	0.991	0.993	0.992	0.991
Fed-MLDL	**FedFLA**	**0.993**	**0.994**	**0.993**	**0.994**
Fed-MLDL	FedTWO	0.991	0.992	0.993	0.993
Fed-MLDL	None	0.972	0.971	0.973	0.973

convergence rate, taking approximately 50 rounds to reach its maximum accuracy level. This indicates that while FedAvg may eventually reach similar performance levels, it is less efficient in terms of convergence speed. The results suggest that FedTWO, FedCDO, FedRIME, FedEVO, and FedFLA are more effective in quickly achieving high accuracy under IID conditions compared to FedAvg.

Figure 6 and, Figure 10 show the average loss across the selected clients at each round in an IID distribution. All the optimization algorithms exhibit a sharp decline in loss within the first 20 rounds. FedTWO and FedRIME consistently show the lowest loss values, indicating their superior performance in reducing error quickly. However, FedAvg lags significantly behind other algorithms, resulting

TABLE 9. Performance comparison of Fed-MLDL on different hyper-parameter optimization algorithms in Non-IID data distribution for 34 class classification.

Model Used	HPO Algorithm	Accuracy	Precision	Recall	F1-Score
Fed-MLDL	FedRIME	0.990	0.992	0.993	0.992
Fed-MLDL	FedEVO	0.989	0.990	0.991	0.991
Fed-MLDL	FedCDO	0.990	0.991	0.992	0.990
Fed-MLDL	**FedFLA**	**0.991**	**0.993**	**0.992**	**0.992**
Fed-MLDL	FedTWO	0.988	0.989	0.990	0.989
Fed-MLDL	None	0.970	0.969	0.971	0.971

in higher loss values throughout the training rounds. This suggests that FedAvg struggles to adapt as quickly as the other algorithms in an IID setting. FedCDO, FedEVO, and FedFLA show rapid convergence, with their loss values stabilizing near the minimum after 20 rounds. Figure 7 and, Figure 13 illustrate the average loss across the selected clients at each round in a non-IID distribution. A similar trend is observed, with all algorithms showing a sharp decline in loss within the first 20 rounds. However, the initial variance is slightly higher compared to the IID scenario, reflecting the increased difficulty of learning from non-IID data. Despite this, FedTWO and FedRIME maintain the lowest loss values throughout, confirming their robustness and effectiveness even in more challenging data distributions. FedAvg continues to underperform, showing not only a pronounced initial lag but also higher loss values throughout the training process compared to other algorithms. This persistent underperformance of FedAvg in both IID and non-IID distributions highlights its limitations in adapting to diverse data scenarios. The performance of FedCDO, FedEVO, and FedFLA remains consistent, with all algorithms achieving near-zero loss by the end of the training, illustrating their adaptability and efficiency in non-IID settings.

Figure 14 and, Figure 9 show the server loss at each round in IID distribution. All algorithms display a rapid decrease in server loss within the first 20 rounds. FedTWO and FedRIME show the lowest server loss value. In contrast, FedAvg shows a slower convergence and higher loss value. FedCDO, FedEVO, and FedFLA show rapid convergence with their loss values stabilizing near zero after 20 rounds. Figure 15 and, Figure 12 show that there is a sharp decline in server loss within the first 25 rounds. The initial variance is slightly higher compared to the IID scenario, reflecting the greater challenge of learning from non-IID data. FedAvg continues to underperform, exhibiting a significant initial lag and higher loss values throughout the training process. This highlights FedAvg's limitations in diverse data scenarios. The consistent performance of FedCDO, FedEVO, and FedFLA, with all achieving near-zero loss by the end of the training, illustrates their adaptability and efficiency in both IID and non-IID environments.

Table 4, Table 5 and Table 6 shows the performance of the HPO techniques on the Fed-MLDL model. The Fed-MLDL on Fed-RIME optimization achieved the highest accuracy of 99.7%, with precision, recall, and F1-score values all above 99%. There is a decrease in the values of the metrics for the 8-class and 34-class tasks because of the increased

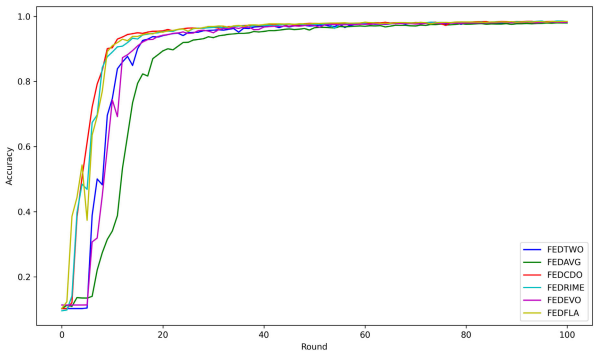


FIGURE 4. Comparison of accuracy in different HPO technique used in Fed-MLDL in Non-IID data distribution on binary classification.

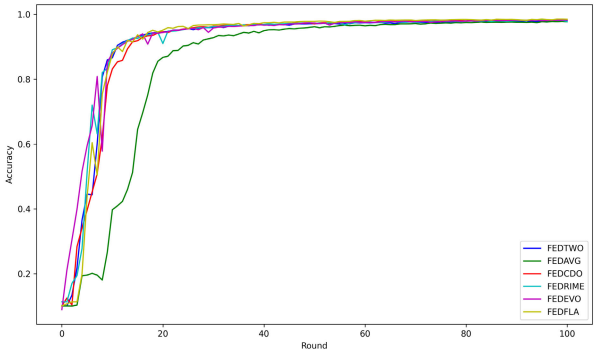


FIGURE 5. Comparison of accuracy in different HPO technique used in Fed-MLDL in IID data distribution on binary classification.

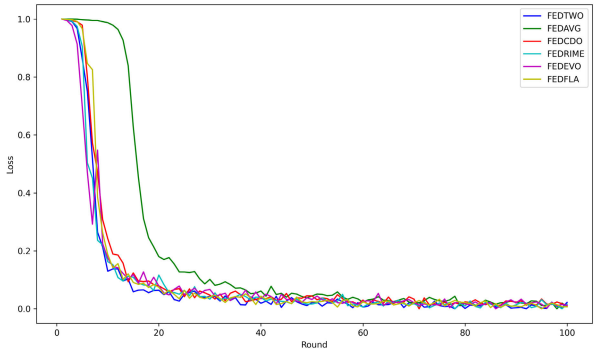


FIGURE 6. Comparison of distributed loss in different HPO techniques used in Fed-MLDL in IID data distribution on binary classification.

TABLE 10. Performance of various models on IoT-based datasets across different classification tasks.

Dataset	Model	Classification Task	Accuracy	Precision	Recall	F1-Score
CICIoT23	LSTM	2-Class	0.98	0.98	0.976	0.972
		8-Class	0.976	0.98	0.975	0.972
		34-Class	0.97	0.973	0.98	0.973
	Random Forest	2-Class	0.99	0.99	0.993	0.992
		8-Class	0.955	0.955	0.955	0.955
		34-Class	0.963	0.962	0.961	0.96
	Federated Learning	2-Class	0.98	0.98	0.98	0.982
		8-Class	0.975	0.976	0.974	0.977
		34-Class	0.97	0.97	0.97	0.969
	FedMLDL with Bayesian HPO	2-Class	0.996	0.993	0.994	0.994
		8-Class	0.994	0.992	0.993	0.993
		34-Class	0.992	0.99	0.991	0.991
	FedMLDL with RIME	**2-Class**	**0.997**	**0.994**	**0.995**	**0.995**
		8-Class	**0.995**	**0.993**	**0.994**	**0.994**
		34-Class	**0.993**	**0.991**	**0.993**	**0.992**
CICIoT22	LSTM	2-Class	0.965	0.967	0.964	0.966
	Random Forest	2-Class	0.971	0.972	0.974	0.973
	Federated Learning	2-Class	0.969	0.968	0.967	0.968
	FedMLDL with Bayesian HPO	2-Class	0.985	0.981	0.978	0.978
	FedMLDL with RIME	**2-Class**	**0.987**	**0.982**	**0.981**	**0.981**
ToN_IoT	LSTM	10-Class	0.974	0.978	0.973	0.976
	Random Forest	10-Class	0.979	0.979	0.975	0.977
	Federated Learning	10-Class	0.972	0.969	0.967	0.968
	FedMLDL with Bayesian HPO	10-Class	0.981	0.979	0.98	0.98
	FedMLDL with RIME	**10-Class**	**0.981**	**0.983**	**0.983**	**0.982**
Edge_IIoTset	LSTM	15-Class	0.969	0.97	0.969	0.969
	Random Forest	15-Class	0.978	0.976	0.974	0.975
	Federated Learning	15-Class	0.974	0.972	0.971	0.972
	FedMLDL with Bayesian HPO	15-Class	0.982	0.982	0.98	0.981
	FedMLDL with RIME	**15-Class**	**0.986**	**0.984**	**0.984**	**0.985**
IoT-23	LSTM	7-Class	0.968	0.969	0.971	0.97
	Random Forest	7-Class	0.973	0.975	0.978	0.977
	Federated Learning	7-Class	0.971	0.972	0.974	0.972
	FedMLDL with Bayesian HPO	7-Class	0.975	0.978	0.981	0.979
	FedMLDL with RIME	**7-Class**	**0.979**	**0.981**	**0.983**	**0.986**

complexity of the problem. Table 7, Table 8 and Table 9 shows the performance of the HPO techniques on Fed-MLDL model. The Fed-MLDL on FedFLA optimization achieved the highest accuracy of 99.4% and precision, recall, and

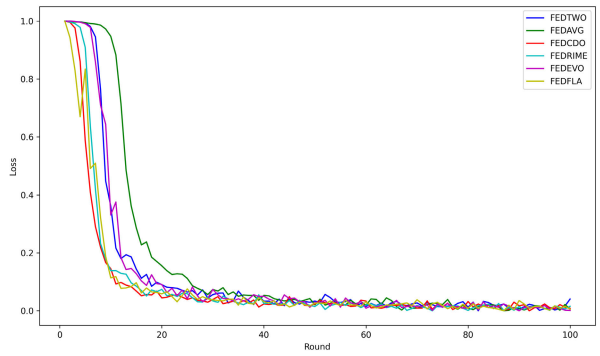


FIGURE 7. Comparison of distributed loss in different HPO techniques used in Fed-MLDL in Non-IID data distribution on binary classification.

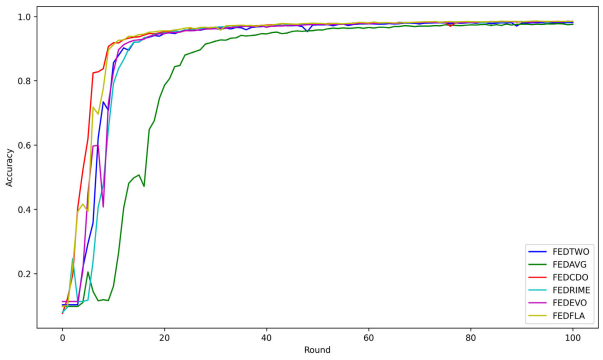


FIGURE 8. Comparison of accuracy in different HPO techniques used in Fed-MLDL in IID data distribution on 34 class classification.

F1-score values all above 99%. Similar to the IID distribution there is a decrease in the value of the metrics in the 8 class and 34 class tasks. FedAvg showed the lowest performance with an accuracy of 97.8% for binary classification and showed the lowest performance in the 8-class and 34-class tasks.

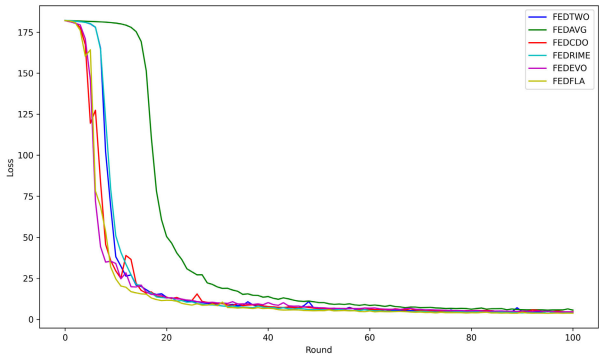


FIGURE 9. Comparison of server loss in different HPO techniques used in Fed-MLDL in IID data distribution on 34 class classification.

V. DISCUSSION

The research described in this article underscores the transformative potential of physics-based HPO algorithms in FL environments, particularly when compared to traditional methods such as FedAvg. By evaluating the performance

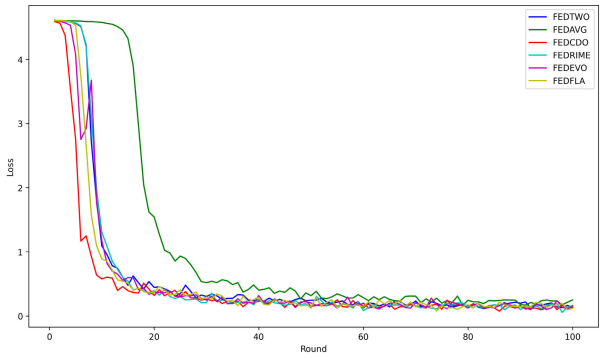


FIGURE 10. Comparison of distributed loss in different HPO techniques used in Fed-MLDL in IID data distribution on 34 class classification.

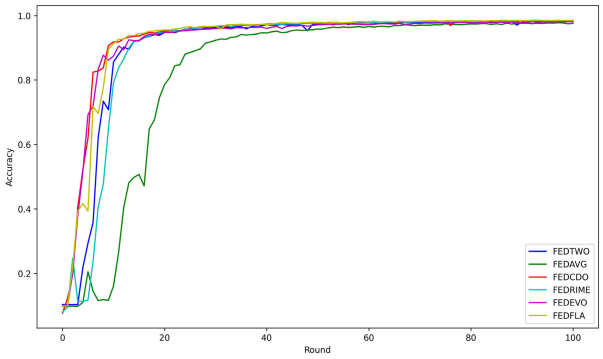


FIGURE 11. Comparison of accuracy in different HPO techniques used in Fed-MLDL in Non-IID data distribution on 34 class classification.

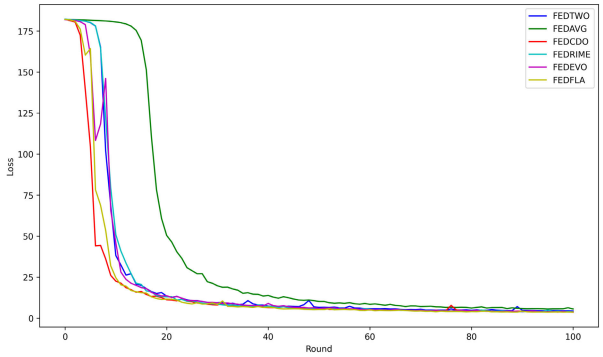


FIGURE 12. Comparison of server loss in different HPO techniques used in Fed-MLDL in Non IID data distribution on 34 class classification.

of various HPO algorithms (FedTWO, FedCDO, FedRIME, FedEVO, and FedFLA) across IID and Non-IID data distributions, the study reveals several compelling advantages that can be leveraged to enhance FL.

An important finding from this study is the significant increase in convergence speed and stability offered by physics-based HPO algorithms as compared to traditional hyperparameter optimizers, for instance, Bayesian HPO [46]. Empirically, the experimentation demonstrates that FedMLDL enhanced with RIME consistently performs better than other techniques and reaches convergence within

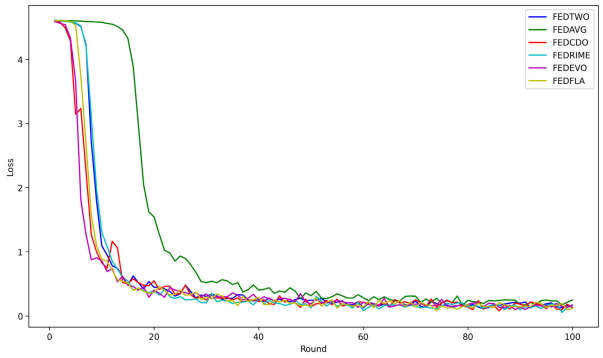


FIGURE 13. Comparison of distributed Loss in different HPO techniques used in Fed-MLDL in Non-IID data distribution on 34 class classification.

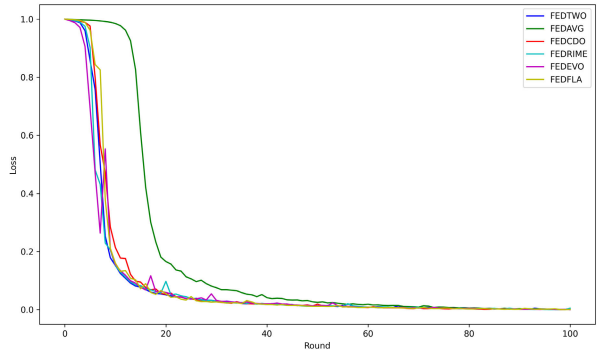


FIGURE 14. Comparison of server loss in different HPO techniques used in Fed-MLDL in IID data distribution on binary classification.

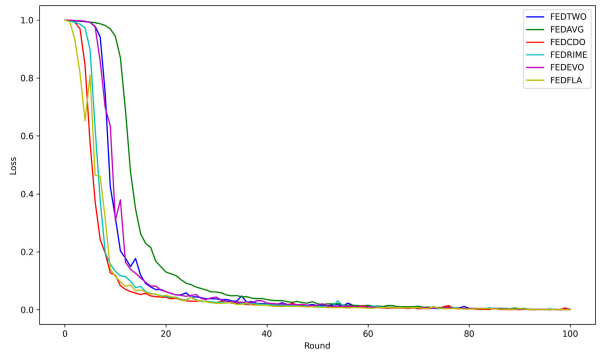


FIGURE 15. Comparison of server loss in different HPO techniques used in Fed-MLDL in Non-IID data distribution on binary classification.

10-15 rounds of communication. This hence increases efficiency in FL settings, where communication overhead and latency are often critical concerns, especially in IoT based scenarios, where devices are compute constrained. Another of the inherent challenges in FL is handling the variability in data distributions (Non-IID) where data is highly heterogeneous and requires numerous rounds of training for convergence. In such settings, FedAvg, which depends on the averaging of gradients from all clients, frequently proves to be inadequate and requires substantial amount of training time. The averaged global model may not effectively capture the unique characteristics of each client's data, resulting in sub-optimal performance and low accuracy. One of the few limitations for the incorporation of HPO techniques in IoT devices is the lack of compute power in these devices for calculating the best hyperparameters for the model. These devices would require an intermediate computing environment, preferably a FOG node, which can do the computations on behalf of the IoT device and compute the best hyperparameters for the model and train them on the dataset. This ensures that resource constrained devices are given adequate priority.

In contrast, the Fed-MLDL model is distinguished by its ability to customize the training process to meet the unique requirements of each client. By optimizing hyperparameters such as learning rate, momentum, and weight decay on a per-client basis, the model can better adapt to each client's data distribution. The empirical results of the research, as illustrated in Figure 14, 9, 15, and 12 demonstrate this assertion. The graphs indicate that optimization techniques substantially diminish the variability in training loss among clients. Amongst the Physics-based HPO techniques tested FedRIME and FedFLA algorithms demonstrate minimal loss fluctuation, suggesting that they can rapidly and consistently obtain optimal solutions. In comparison, FedAvg shows more pronounced fluctuations, which indicates that sub-optimal solutions are more prevalent during the initial training cycles.

The Fed-MLDL coupled with FedRIME's advantages have been shown to hold true even in comparison with state-of-the-art (SoTA) models in the field. Table 10 shows the performance of the Fed-MLDL model with SOTA models, and also Bayesian HPO [46] on the various publicly available datasets as mentioned in the previous sections.

The results show that the Fed-MLDL model enhanced with FedRIME significantly outperforms the SoTA models with 99.7& accuracy for 2 class classification, 99.5% accuracy for 8 class classification, and 99.3% accuracy for 34 class classification. We can observe that the model's performance is consistently the highest among the public IoT based datasets utilized in the study.

To the best of our knowledge, this study is the first to comprehensively investigate the efficacy of a diverse array of physics-based HPO algorithms in an FL environment. The algorithms demonstrate superiority over conventional methods such as FedAvg in both IID and Non-IID data distributions, which substantially contributes to the field. These results open the door to additional research on utilizing HPO techniques in the field of FL.

VI. CONCLUSION AND FUTURE WORKS

This study proposes a new FedMLDL model enhanced with FedRIME HPO. The article also presents a comprehensive comparison with most recent physics based HPO techniques, coupled with FedMLDL in an IoT environment. The goal of this study is to assess the performance and generalization capabilities of these combined approaches on the recently published CICIoT23 dataset and other IoT network traffic

datasets such as CICIoT22, ToN_IoT, Edge_IIoTset and IoT23 which includes both IID and non-IID data distributions, using standard performance metrics, providing insightful comparison. To address the inherent data imbalances in these datasets, the work makes use of the SMOTE algorithm.

The experimental results indicated that the integration of metaheuristic algorithms with FL significantly enhances the model's robustness and scalability by allowing decentralized data processing and personalized training. Among hyperparameter tuning algorithms, FedRIME and FedFLA achieved better accuracy, precision, recall, and F1 scores in IID and non-IID distributions (for CICIoT23), respectively. These findings highlight the enhancement in efficacy and performance of FL-based environments. Coupling FL with physics-based HPO techniques, offers a significant increase in convergence speed which hints at the adaptability and personalization of the client models to heterogeneous and Non-IID Datasets, which is one of the major concerns of FL [47]. This is crucial in reducing disparities among the client models in FL, thus ensuring that the weight updates in the global model are smooth and steady. Model personalization in FL ensures that the client-model is able to adapt and learn the data distributions of each client [48]. This is a promising avenue of research as FL shifts towards the use of personalized models for individual clients to increase the efficacy of the global model. This study's alignment with emerging trends in FL and IoT based environments reinforce its potential in the long run.

With the ability to combine like clients, Federated clustering is a promising and innovative method to improve the effectiveness of training in FL systems. Model performance may be improved and training times can be greatly accelerated by this method as the algorithm clusters clients with similar behavior and trains them together. Creating novel Federated clustering algorithms that efficiently group clients according to computing capacity, data similarity, or other pertinent criteria can greatly increase the efficiency of model training time in an FL environment and also be a use case for Vehicular ad-hoc Networks (VANETs) [49]. Ensuring the scalability and efficiency of Federated clustering techniques, especially for large-scale installations with thousands or millions of clients. Another possible improvement in the future would be the use of more advanced aggregation strategies according to requirements [50]. Aiming to train models for several tasks at once inside an FL framework is Federated multi-task learning (FMTL). Better use of data from many sources and more generalized models can result from this method. Developing new FMTL algorithms that can make use of shared representations among several tasks while allowing for task-specific changes. Some other approaches to FMTL include looking at ways to improve knowledge transfer between tasks in a Federated environment, maybe raising the performance of each task. Also, to ensure that FMTL-based algorithms are robust and scalable.

CONFLICT OF INTEREST

The authors declare that they have no competing interests.

REFERENCES

[1] Cobalt. (2024). *Top Cybersecurity Statistics for 2024*. Accessed: Sep. 2, 2024. [Online]. Available: https://www.cobalt.io/blog/cybersecurity-statistics-2024

[2] Z. Chao-yang, "DOS attack analysis and study of new measures to prevent," in *Proc. Int. Conf. Intell. Sci. Inf. Eng.*, Aug. 2011, pp. 426–429.

[3] M. K. Hasan, A. K. M. A. Habib, S. Islam, N. Safie, S. N. H. S. Abdullah, and B. Pandey, "DDoS: Distributed denial of service attack in communication standard vulnerabilities in smart grid applications and cyber security with recent developments," *Energy Rep.*, vol. 9, pp. 1318–1326, Oct. 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352484723009447

[4] Y. Li and Q. Liu, "A comprehensive review study of cyber-attacks and cyber security; emerging trends and recent developments," *Energy Rep.*, vol. 7, pp. 8176–8186, Nov. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352484721007289

[5] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: Techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, p. 20, Dec. 2019, doi: 10.1186/s42400-019-0038-7.

[6] D. E. Denning, "An intrusion-detection model," *IEEE Trans. Softw. Eng.*, vol. SE-13, no. 2, pp. 222–232, Feb. 1987.

[7] P. Rajkumar and R. Sandhu, "Safety decidability for pre-authorization usage control with finite attribute domains," *IEEE Trans. Dependable Secure Comput.*, vol. 13, no. 5, pp. 582–590, 2016.

[8] P. V. Rajkumar and R. Sandhu, "Safety decidability for pre-authorization usage control with identifier attribute domains," *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 3, pp. 465–478, May 2020.

[9] M. Mohammadi, T. A. Rashid, S. H. T. Karim, A. H. M. Aldalwie, Q. T. Tho, M. Bidaki, A. M. Rahmani, and M. Hosseinzadeh, "A comprehensive survey and taxonomy of the SVM-based intrusion detection systems," *J. Netw. Comput. Appl.*, vol. 178, Mar. 2021, Art. no. 102983. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804521000102

[10] Z. Azam, M. M. Islam, and M. N. Huda, "Comparative analysis of intrusion detection systems and machine learning-based model analysis through decision tree," *IEEE Access*, vol. 11, pp. 80348–80391, 2023.

[11] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bull. Math. Biophys.*, vol. 5, no. 4, pp. 115–133, Dec. 1943, doi: 10.1007/bf02478259.

[12] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jan. 1998, pp. 2278–2324.

[13] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986, doi: 10.1038/323533a0.

[14] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, vol. 54., 2017, pp. 1273–1282.

[15] J. Alsamiri and K. Alsubhi, "Federated learning for intrusion detection systems in Internet of Vehicles: A general taxonomy, applications, and future directions," *Future Internet*, vol. 15, no. 12, p. 403, Dec. 2023. [Online]. Available: https://www.mdpi.com/1999-5903/15/12/403

[16] A. Alazab, A. Khraisat, S. Singh, and T. Jan, "Enhancing privacy-preserving intrusion detection through federated learning," *Electronics*, vol. 12, no. 16, p. 3382, Aug. 2023, doi: 10.3390/electronics12163382.

[17] H. Wang, Z. Kaplan, D. Niu, and B. Li, "Optimizing federated learning on non-IID data with reinforcement learning," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Jul. 2020, pp. 1698–1707.

[18] K. Hu, Y. Li, S. Zhang, J. Wu, S. Gong, S. Jiang, and L. Weng, "FedMMD: A federated weighting algorithm considering non-IID and local model deviation," *Exp. Syst. Appl.*, vol. 237, Mar. 2024, Art. no. 121463. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417423019656

[19] X. Ma, J. Zhu, Z. Lin, S. Chen, and Y. Qin, "A state-of-the-art survey on solving non-IID data in federated learning," *Future Gener. Comput. Syst.*, vol. 135, pp. 244–258, Oct. 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X22001686

[20] M. Kundroo and T. Kim, "Federated learning with hyper-parameter optimization," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 35, no. 9, Oct. 2023, Art. no. 101740. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S131915782300294X

[21] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, "CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment," *Sensors*, vol. 23, no. 13, p. 5941, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/13/5941

[22] P. Wasnik and N. Chavhan, "Designing intelligent intrusion detection system using deep learning," in *Proc. 14th Int. Conf. Comput. Commun. Netw. Technol. (ICCCNT)*, Jul. 2023, pp. 1–9.

[23] S. M. Kasongo, "A deep learning technique for intrusion detection system using a recurrent neural networks based framework," *Comput. Commun.*, vol. 199, pp. 113–125, Feb. 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366422004601

[24] A. Rosay, F. Carlier, and P. Leroux, "MLP4NIDS: An efficient MLP-based network intrusion detection for CICIDS2017 dataset," in *Machine Learning for Networking*, S. Boumerdassi, É. Renault, and P. Mühlethaler., Cham, Switzerland: Springer, 2020, pp. 240–254.

[25] O. Arreche, T. Guntur, and M. Abdallah, "XAI-IDS: Toward proposing an explainable artificial intelligence framework for enhancing network intrusion detection systems," *Appl. Sci.*, vol. 14, no. 10, p. 4170, May 2024. [Online]. Available: https://www.mdpi.com/2076-3417/14/10/4170

[26] J. F. C. Garcia and G. E. T. Blandon, "A deep learning-based intrusion detection and preventation system for detecting and preventing denial-of-service attacks," *IEEE Access*, vol. 10, pp. 83043–83060, 2022.

[27] S. Agrawal, S. Sarkar, M. Alazab, P. K. R. Maddikunta, T. R. Gadekallu, and Q.-V. Pham, "Genetic CFL: Hyperparameter optimization in clustered federated learning," *Comput. Intell. Neurosci.*, vol. 2021, no. 1, Jan. 2021, Art. no. 7156420, doi: 10.1155/2021/7156420.

[28] D. Kilichev and W. Kim, "Hyperparameter optimization for 1D-CNN-based network intrusion detection using GA and PSO," *Mathematics*, vol. 11, no. 17, p. 3724, Aug. 2023. [Online]. Available: https://www.mdpi.com/2227-7390/11/17/3724

[29] S. Park, Y. Suh, and J. Lee, "FedPSO: Federated learning using particle swarm optimization to reduce communication costs," *Sensors*, vol. 21, no. 2, p. 600, Jan. 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/2/600

[30] A. K. Abasi, M. Aloqaily, and M. Guizani, "Grey wolf optimizer for reducing communication cost of federated learning," in *Proc. IEEE Global Commun. Conf.*, Dec. 2022, pp. 1049–1154.

[31] M. Azizi, U. Aickelin, H. A. Khorshidi, and M. B. Shishehgarkhaneh, "Energy valley optimizer: A novel metaheuristic algorithm for global and engineering optimization," *Sci. Rep.*, vol. 13, no. 1, p. 226, Jan. 2023.

[32] H. A. Shehadeh, "Chernobyl disaster optimizer (CDO): A novel meta-heuristic method for global optimization," *Neural Comput. Appl.*, vol. 35, no. 15, pp. 10733–10749, May 2023.

[33] F. A. Hashim, R. R. Mostafa, A. G. Hussien, S. Mirjalili, and K. M. Sallam, "Fick's law algorithm: A physical law-based algorithm for numerical optimization," *Knowl.-Based Syst.*, vol. 260, Jan. 2023, Art. no. 110146. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705122012424

[34] T. Nguyen, B. Hoang, G. Nguyen, and B. M. Nguyen, "A new workload prediction model using extreme learning machine and enhanced tug of war optimization," *Proc. Comput. Sci.*, vol. 170, pp. 362–369, Jan. 2020.

[35] H. Su, D. Zhao, A. A. Heidari, L. Liu, X. Zhang, M. Mafarja, and H. Chen, "RIME: A physics-based optimization," *Neurocomputing*, vol. 532, pp. 183–214, May 2023, doi: 10.1016/j.neucom.2023.02.010.

[36] R. Lazzarini, H. Tianfield, and V. Charissis, "Federated learning for IoT intrusion detection," *AI*, vol. 4, no. 3, pp. 509–530, Jul. 2023. [Online]. Available: https://www.mdpi.com/2673-2688/4/3/28

[37] S. Abbas, A. A. Hejaili, G. A. Sampedro, M. Abisado, A. S. Almadhor, T. Shahzad, and K. Ouahada, "A novel federated edge learning approach for detecting cyberattacks in IoT infrastructures," *IEEE Access*, vol. 11, pp. 112189–112198, 2023.

[38] O. Friha, M. A. Ferrag, L. Shu, L. Maglaras, K.-K.-R. Choo, and M. Nafaa, "FELIDS: Federated learning-based intrusion detection system for agricultural Internet of Things," *J. Parallel Distrib. Comput.*, vol. 165, pp. 17–31, Jul. 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0743731522000570

[39] J. Li, X. Tong, J. Liu, and L. Cheng, "An efficient federated learning system for network intrusion detection," *IEEE Syst. J.*, vol. 17, no. 2, pp. 2455–2464, Jun. 2023.

[40] I. Muraina, "Ideal dataset splitting ratios in machine learning algorithms: General concerns for data scientists and data analysts," in *Proc. 7th Int. Mardin Artuklu Sci. Res. Conf.*, Feb. 2022, pp. 496–504.

[41] I. Tareq, B. M. Elbagoury, S. El-Regaily, and E.-S.-M. El-Horbaty, "Analysis of ToN-IoT, UNW-NB15, and edge-IIoT datasets using DL in cybersecurity for IoT," *Appl. Sci.*, vol. 12, no. 19, p. 9572, Sep. 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/19/9572

[42] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and H. Janicke, "Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning," *IEEE Access*, vol. 10, pp. 40281–40306, 2022.

[43] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.

[44] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, J. Fernandez-Marques, Y. Gao, L. Sani, K. Hei Li, T. Parcollet, P. P. B. de Gusmão, and N. D. Lane, "Flower: A friendly federated learning research framework," 2020, *arXiv:2007.14390*.

[45] N. Van Thieu and S. Mirjalili, "MEALPY: An open-source library for latest meta-heuristic algorithms in Python," *J. Syst. Archit.*, vol. 139, Jun. 2023, Art. no. 102871.

[46] Y. N. Kunang, S. Nurmaini, D. Stiawan, and B. Y. Suprapto, "Improving classification attacks in IoT intrusion detection system using Bayesian hyperparameter optimization," in *Proc. 3rd Int. Seminar Res. Inf. Technol. Intell. Syst. (ISRITI)*, Dec. 2020, pp. 146–151.

[47] S. Ji, Y. Tan, T. Saravirta, Z. Yang, Y. Liu, L. Vasankari, S. Pan, G. Long, and A. Walid, "Emerging trends in federated learning: From model fusion to federated X learning," *Int. J. Mach. Learn. Cybern.*, vol. 15, no. 9, pp. 3769–3790, Sep. 2024, doi: 10.1007/s13042-024-02119-1.

[48] Z. Li, Z. Zhong, P. Zuo, and H. Zhao, "A personalized federated learning method based on the residual multi-head attention mechanism," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 36, no. 4, Apr. 2024, Art. no. 102043. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1319157824001320

[49] X. Chen, W. Qiu, L. Chen, Y. Ma, and J. Ma, "Fast and practical intrusion detection system based on federated learning for VANET," *Comput. Secur.*, vol. 142, Jul. 2024, Art. no. 103881. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404824001822

[50] V. Agarwal, C. J. Chandnani, S. C. Kulkarni, A. Aren, and K. Srinivasan, "A comparative analysis of aggregation methods in federated learning on mnist," in *Artificial Intelligence and Knowledge Processing*, H. K, R. V. Rodriguez, M. Rege, A. Ade-Ibijola, K.-L. Ong, and V. Piuri., Cham, Switzerland: Springer, 2025, pp. 225–238.

CHIRAG JITENDRA CHANDNANI is currently pursuing the Bachelor of Technology (B.Tech.) degree with Vellore Institute of Technology (VIT), Vellore, India. Throughout his academic journey, he has actively participated in numerous research initiatives at the institute, gaining significant hands-on experience. He has had the opportunity to collaborate with renowned professors, contributing to various projects in the fields of deep learning and optimization algorithms. His research interests include machine learning, deep learning, natural language processing (NLP), computer vision, and decentralized models.

VEDIK AGARWAL was born in India. He is currently pursuing the B.Tech. degree with Vellore Institute of Technology (VIT), Vellore. He is also an Undergraduate Researcher by collaborating with distinguished professors in the domains of cybersecurity and NLP. His research interests include multimodal language models and the creation of cross-domain applications in the realms of reinforcement learning and computer vision.

SHLOK CHETAN KULKARNI is currently pursuing the B.Tech. degree with Vellore Institute of Technology, Vellore. He has previously taken part in research at Vellore Institute of Technology. He has carried out research in the fields of machine learning, deep learning, and LLMs. His research interests include federated learning, LLMs, machine learning algorithms, and sentiment analysis.

ADITYA AREN is currently pursuing the B.Tech. degree with Vellore Institute of Technology (VIT), Vellore. He is actively engaged in the field of application development, with a keen interest in exploring machine learning technologies. He is also involved in research, focusing on applying machine learning techniques to real-world problems. In addition to his academic pursuits, he is an Active Member of several development societies at VIT, where he collaborates on projects and hones his skills in building innovative solutions.

D. GERALDINE BESSIE AMALI received the M.Tech. (Hons.) and Ph.D. degrees in computer science and engineering from VIT Vellore, in 2014 and 2019, respectively. She is currently an Associate Professor. She has published more than 30 papers in various international journals and conferences and has authored book chapters in the fields of machine learning and natural language processing. Her research interests include machine learning, natural language processing, and biologically inspired optimization algorithms.

KATHIRAVAN SRINIVASAN (Senior Member, IEEE) received the B.E. degree in electronics and communication engineering, the M.E. degree (Hons.) in communication systems engineering, and the Ph.D. degree in information and communication engineering from Anna University, Chennai, India. He is currently a Full Professor with the School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India. He was previously a Faculty Member/ Lecturer with the Department of Computer Science and Information Engineering and the Deputy Director of the Office of International Affairs, National Ilan University, Taiwan. His research interests include artificial intelligence, machine learning, deep learning, communication systems and devices, computer vision, feature engineering, and intelligent sensing. He is one of the Top 2% Most Influential Scientists in the Stanford University List for two consecutive years 2022 and 2023. In 2016, he received the Best Service Award as the Deputy Director at the Office of International Affairs, National Ilan University. He was awarded the IEEE Access Outstanding Associate Editor in 2019, 2020, 2021, 2022, and 2023. He is also serving as the Associate Editor for IEEE Access, *IET Networks*, *Array* (Elsevier), and *Journal of Internet Technology*.