

Article

Fusing Localized Network Expertise via Global Meta-Learning

Małgorzata Przybyła-Kasperek ^{1,2}  and Kornel Chromiński ^{1,*} ¹ Institute of Computer Science, University of Silesia in Katowice, Będzińska 39, 41-200 Sosnowiec, Poland; malgorzata.przybyla-kasperek@us.edu.pl² Faculty of Natural Sciences nad Informatics, Constantine the Philosopher University in Nitra, Trieda Andreja Hlinku 1, 949 01 Nitra, Slovakia

* Correspondence: kornel.chrominski@us.edu.pl; Tel.: +48-368-97-63

Abstract

Detecting anomalies in network traffic is a critical challenge due to the increasing risk and potential consequences of cyberattacks. Building accurate intrusion detection systems is complicated by the massive volume of operations, significant class imbalance, and the highly contextual nature of legitimate traffic baselines across different network locations. Consequently, centralizing raw network logs to build a single universal model is often suboptimal and raises severe privacy concerns. To address these limitations, this paper proposes a novel, two-level hierarchical classification architecture tailored for dispersed network environments. At the first level, autonomous, local classifiers act as localized domain experts trained on specific node data. At the second level, a global classifier learns patterns and makes final decisions exclusively by fusing the abstract prediction vectors (probability distributions) generated by the local models. This structural framework circumvents the need for both raw data centralization and the heavy communication overhead typical of iterative approaches. Extensive hyperparameter tuning and experimental validations were conducted using Multilayer Perceptron (MLP) and Random Forest (RF) algorithms on two real-world cybersecurity datasets: CIC-DDoS2019 and CTU-13. The results confirm that the proposed hierarchical architecture achieves high classification quality, matching or surpassing the performance of classical centralized models. This approach ensures robust fusion of localized expertise and optimizes operational efficiency while significantly mitigating privacy risks.

Keywords: dispersed data; hierarchical classification; intrusion detection; meta-learning

1. Introduction

Detecting all types of anomalies in network traffic that may constitute attempts to attack IT systems is currently one of the most significant issues [1], given the ever-increasing risk of attacks and their potential consequences. The proper functioning of security systems is critical, as it can protect an organization from serious financial and reputational losses. However, developing highly accurate intrusion detection systems is challenged by the inherent nature of the underlying data. Data derived from network traffic analysis is quite specific, both in terms of the massive volume of operations on the network and the significant class imbalance [2]. In real-world computer networks, the vast majority of network traffic relates to legitimate operations; traffic that could constitute a cyberattack represents only a small fraction of the total.

Furthermore, the nature of such traffic depends largely on location, or more precisely, on the services operating within a given network [3]. It is these services that generate



Academic Editor: George Drosatos

Received: 28 May 2026

Revised: 25 June 2026

Accepted: 26 June 2026

Published: 1 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

legitimate network traffic; conversely, they may also be the target of potential attacks on the infrastructure. Because the baseline of what constitutes "normal" behavior is highly contextual, attempting to build a single centralized model that works well in every case may be quite difficult [4]. Centralizing raw network logs not only dilutes specialized local traffic signatures but also raises severe operational and privacy concerns.

To address the challenges of decentralized data, the machine learning community has increasingly turned to distributed [5] and federated learning paradigms [6]. In traditional distributed learning, different approaches can be identified, where local models are independently constructed and then combined using fusion techniques for the final decision-making [7]. Federated learning challenges are currently being explored in depth due to the increasing prevalence of decentralized organizational structures in various industries and widespread data collection practices across different domains [8]. Moreover, the significance of safeguarding data privacy has become paramount.

While federated learning effectively mitigates privacy concerns by keeping raw data on local devices, it frequently relies on the iterative convergence of a global model through continuous parameter synchronization. Such iterative communication can be bottlenecked by network latency and assumes a degree of structural homogeneity among the participating nodes. However, in real-world dispersed environments, it is often assumed that the dispersed data are collected separately, without strict control over the dispersion of data or the alignment of local architectures. In this paper, we have focused on a different, non-iterative meta-learning scheme and compared it empirically with strong, centralized baseline models; we leave comparisons with FedAvg, personalized FL, FL based on model ensembles, or low-communication-cost variants of FL to future research.

Given these constraints, the main premise of this research is that since data from network traffic analysis will vary significantly depending on the location where the traffic is monitored, it seems a better idea to create local models and, only on that basis, to create a single global model. Building upon previous concepts of dispersed data classification [9,10], this paper introduces a novel hierarchical architecture specifically tailored for cybersecurity. By training specialized local neural networks at the edge and employing a global meta-learner to interpret their prediction vectors, the proposed approach leverages local domain expertise, circumvents the need for centralized raw data aggregation, and avoids the heavy communication overhead typical of iterative federated learning.

To clearly describe the scope and innovations of this research, the main contributions of this paper are summarized as follows:

- Proposing a hierarchical classification structure specifically tailored for the task of cyberattack and anomaly detection in dispersed network environments.
- Deploying Multilayer Perceptron (MLP) neural networks at both the local and global levels.
- Aggregating localized domain expertise exclusively via abstract prediction vectors to eliminate the need for centralizing sensitive raw logs and circumvent the communication overhead typical of iterative federated learning.
- Conducting extensive experimental evaluation and hyperparameter tuning of the proposed dual-level architecture using real-world cybersecurity datasets, specifically CIC-DDoS 2019 and CTU-13.

The remainder of this paper is organized as follows. Section 2 reviews related work in the field. Section 3 details the proposed hierarchical architecture and mathematical models. Section 4 describes the datasets and formalizes the problem. Section 5 presents the experimental setup and results, followed by a discussion and final conclusion in Section 6.

2. Related Work

The problem of detecting anomalies in network traffic actually dates back to the early days of computer networks. By ‘anomalies’, we mean both errors caused by the malfunctioning of network hardware and far more serious issues relating to the security of the network infrastructure. Given the potential losses associated with attacks on network infrastructure, researchers have long been developing the most effective methods for detecting them. Advances in machine learning and artificial intelligence have led to the development of methods that learn to recognize attacks rather than rely on traditional heuristics. The first systems based on machine learning methods appeared as early as the last century [11]. Since then, there has been significant progress in AI methods, but unfortunately, also a significant increase in the number of potential attack types. Currently, the most popular approaches are based either on classical machine learning, such as random forests [12,13], or on other machine learning algorithms [14]. Network traffic data is quite specific; it very often depends on location, active network services, and network usage patterns, so classical machine learning methods do not always perform well in traffic classification. Increasingly, deep learning-based methods [15] are used for network traffic classification, and models are being trained for specific situations or device types. Examples include models for detecting attacks on IoT devices [16,17], on communication systems [18], and, more recently, on autonomous vehicles. Recently, large language models for attack detection have also begun to emerge; for example, ZeroDay-LLM [19,20]. However, the literature lacks approaches that combine a wide variety of network traffic data across different networks and strive to develop a more universal model for detecting cyberattacks. Such a model is based on distributed data, so correctly constructing it poses a considerable challenge.

The challenge of classifying dispersed data has driven significant advancements in distributed machine learning. In traditional distributed learning, various approaches construct local models independently and subsequently combine them using fusion techniques for final decision-making [21]; this process can be carried out either in parallel or hierarchically. While these early methods effectively addressed computational load distribution, they were not originally designed to handle strict data privacy regulations.

As the significance of safeguarding data privacy has become paramount, Federated Learning (FL) has emerged as a dominant paradigm. FL challenges are currently being explored in depth due to the increasing prevalence of decentralized organizational structures in various industries and widespread data collection practices [22]. Unlike traditional centralized machine learning, FL enables edge devices or isolated servers to collaboratively learn a shared prediction model while keeping all training data localized. The foundational algorithm in this space, Federated Averaging (FedAvg), operates by executing local training steps on edge devices [23] and periodically sending only the model updates (such as weights or gradients) to a central server for aggregation. Building upon FedAvg, diverse methodologies have been employed in this field to address statistical heterogeneity and communication efficiency, incorporating algorithms such as decision trees, principal component analysis, and deep neural networks. Recent iterations of FL have introduced proximity-based aggregation, asynchronous updates, and personalized federated models to better accommodate non-IID (Independent and Identically Distributed) data [24]. The ability to efficiently categorize and analyze dispersed data without centralizing raw information is central to unlocking valuable insights across multiple sectors. Key domains that can benefit from this architecture include healthcare (e.g., integrating electronic health records, laboratory results, and medical imaging from distinct facilities), finance (e.g., fraud detection, credit risk assessment, and customer segmentation across isolated units), and environmental monitoring (e.g., combining heterogeneous data from dispersed sensors

and satellites), as well as cybersecurity (e.g., anomaly detection and attack localization in decentralized network traffic).

Despite the robust privacy guarantees of standard Federated Learning, its application to specific cybersecurity contexts presents notable operational challenges. Traditional FL relies heavily on the iterative convergence of a global model through continuous communication rounds between nodes and the central server. In environments where data derived from network traffic analysis involves massive volumes of operations and significant class imbalance, the communication latency and synchronization bottlenecks of FedAvg-like methods can become prohibitive. Furthermore, standard FL attempts to synthesize a single global model. However, the nature of network traffic depends largely on location, or more precisely, on the specific services operating within a given network [25]. These localized services generate legitimate traffic baselines that are highly contextual; attempting to build a single centralized model that works well in every case may be quite difficult. Our research is not aimed at replacing iterative federated optimization schemes, such as FedAvg, but rather at proposing a complementary aggregation layer that can operate on any locally trained models. Although our experimental comparison focuses on strong, centralized baseline models (RF and MLP trained on fully aggregated datasets), we are aware that comparing with classical FL algorithms and standard ensemble techniques (e.g., model averaging, soft voting, and stacking) remains an important avenue for future research. Therefore, alternative approaches [26]—where iterative convergence of the global model is not employed, but rather a single-pass hierarchical fusion of prediction vectors is used—offer a more efficient and context-aware solution. By moving away from continuous parameter synchronization, the hierarchical architecture proposed in this study addresses the latency issues of FL while preserving the crucial specialized knowledge of local network nodes.

In recent years, numerous research papers have been published on privacy-preserving intrusion detection systems [27]. The approaches discussed often combine federated learning, meta-learning and hierarchical architectures [28,29]. Several papers have proposed federated or cross-silo intrusion detection system (IDS) architectures, in which network sensors jointly train anomaly detectors without sharing raw network traffic data, often utilising secure aggregation, differential privacy or split learning to provide additional protection for sensitive data. Other studies explore meta-learning strategies for detecting ‘few-shot’ or ‘zero-day’ attacks, where a global meta-model is trained to adapt rapidly to new network environments or attack types using only a small number of local samples. In the context of cybersecurity, hierarchical and multi-level learning schemes have also been investigated; these approaches typically involved combining local detectors with a global fusion layer [30,31]. In contrast to these approaches, our research focuses on a non-iterative meta-learning mechanism at the prediction level. Rather than synchronizing model parameters across multiple rounds of communication, we treat local IDS models as autonomous experts and train the global meta-classifier exclusively on their test-set prediction vectors. This makes our method conceptually closer to ‘stacking’ and ‘mixture-of-experts’ approaches, but adapted to scenarios involving distributed, privacy-preserving IDS systems, where raw traffic data and model parameters remain local.

3. Methods and Models

We proposed hierarchical classification architecture designed for dispersed data analysis. The framework features a two-level structure consisting of a local level (the first stage) and a global level (the second stage). The primary objective of the global model is to learn and identify meta-patterns within the prediction vectors (probability distributions) generated by the first-level classifiers. This general scheme is highly versatile and can

be effectively deployed across a wide range of applications where data is collected independently and centralized raw data aggregation is unfeasible or restricted due to privacy, legal, or technical limitations. It is important to emphasize that this hierarchical framework is inherently model-agnostic; while Multilayer Perceptron (MLP) neural networks and Random Forest (RF) are chosen as the constituent models in this study, the architecture can flexibly accommodate virtually any machine learning algorithm at either level.

The proposed framework utilizes a two-level hierarchical architecture consisting of local models (the first stage) and a global model (the second stage). Rather than executing raw data synthesis or engaging in iterative decentralized model parameter updates—as seen in standard federated learning regimes—the proposed methodology operates by structurally segregating the classification task into localized and generalized decision spaces. The first level comprises a set of distinct, autonomous classifiers trained independently on localized data distributions. These base models function as localized domain experts. The second level introduces a global meta-classifier whose structural objective is to interpret, weight, and learn complex inter-relational patterns from the prediction vectors (i.e., probability distributions over decision classes) yielded by the first-level models.

Each local node trains a specialized model on its specific data distribution. To form a cohesive final prediction, a separate validation dataset is used to capture the probabilistic outputs of these local models. These outputs are concatenated to train a global model. This strategy leverages the specialised knowledge of local models while inherently protecting sensitive data from exposure or transfer through the central unit. This study specifically adopts Multilayer Perceptrons (MLPs) and Random Forest (RF) for both the local and global levels due to their robust function approximation capabilities. The general model flow is presented in Figure 1, and a detailed description is provided in the following sections.

3.1. First Level—Local Specialized Models

We define a dispersed data environment comprising n independent local data sources. Each source maintains a distinct local decision table formulated as:

$$D_i = (U_i, A_i, d), \quad i \in \{1, \dots, n\} \quad (1)$$

where U_i denotes the universe of localized objects (e.g., distinct network flow instances), A_i represents the set of conditional attributes unique to the i -th monitoring node, and d signifies the common decision attribute representing c mutually exclusive target classes. The attribute sets A_i can be structurally heterogeneous across tables, reflecting the localized configurations and specific services operating at each data collection unit.

In the assumptions of this work, a local table represents different network locations or monitored services. Each source contains U_i , the set of network traffic records, A_i features derived from network flow analysis, and d , a specific attack type. In general, the sets of features may vary across locations due to the specific characteristics and configurations of the monitored services. But due to the data used in this publication, all local tables have the same set of features. In other words, the overall architecture has been designed to accommodate heterogeneous local feature spaces and even different model families at each node, as the global classifier utilizes only abstract prediction vectors. However, due to limitations in the publicly available test datasets used in the experiments, we focused on an architecture with homogeneous features, in which all local tables share the same set of attributes and differ primarily in class distribution and traffic profiles. Consequently, the presented experimental results should be interpreted as confirmation of the hypothesis that the proposed meta-learning scheme is effective even in this more limited, homogeneous environment, whilst full verification under real-world conditions with a varied set of features remains a goal for future research.

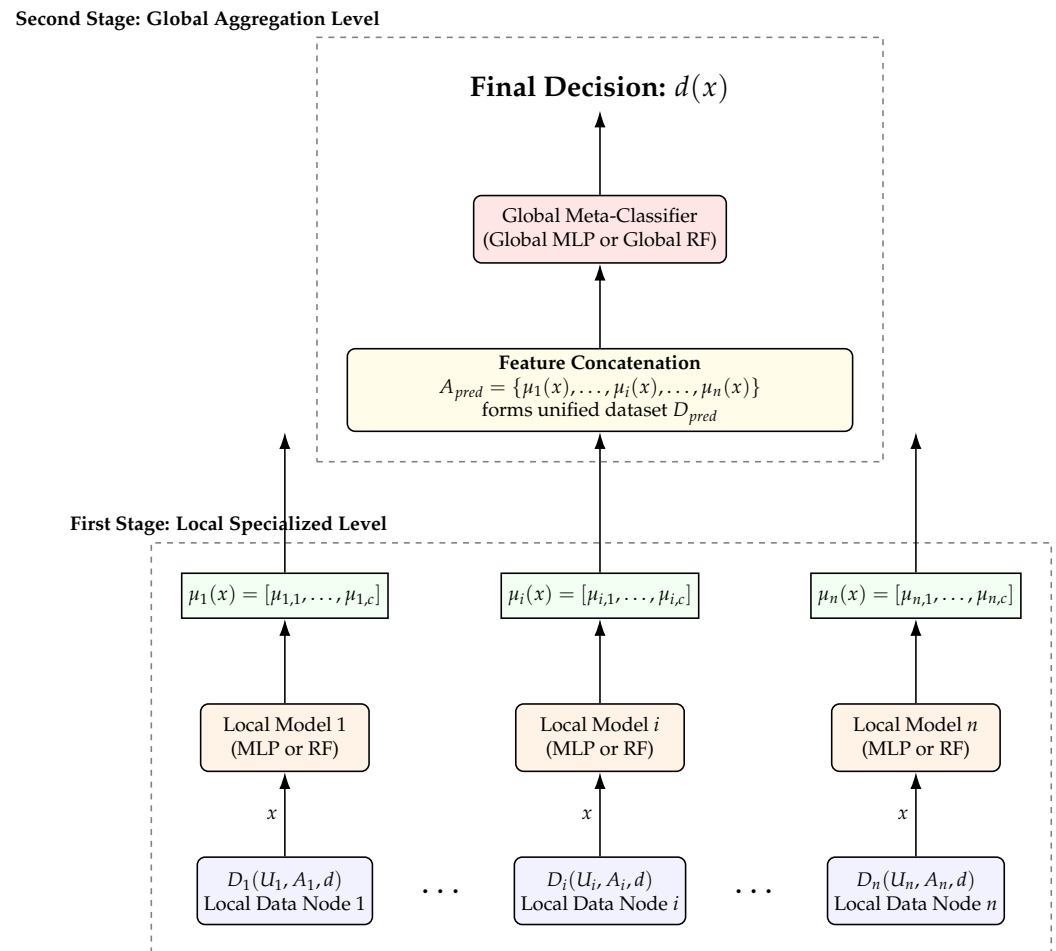


Figure 1. The proposed two-level hierarchical classification architecture.

For each local table D_i , an independent model is trained. For any given object x , the trained i -th local model outputs a localized prediction vector representing a continuous probability distribution over the decision space:

$$\mu_i(x) = [\mu_{i,1}(x), \mu_{i,2}(x), \dots, \mu_{i,c}(x)] \quad (2)$$

where each coordinate $\mu_{i,j}(x) \in [0, 1]$ represents the localized model's confidence or probability estimate that the object x belongs to the j -th decision class, subject to the constraint $\sum_{j=1}^c \mu_{i,j}(x) = 1$. While the proposed dual-level architecture shares structural similarities with classical ensemble stacking and mixture-of-experts paradigms found in distributed learning, its unique contribution lies in its conceptual adaptation for heterogeneous, non-iterative intrusion detection. Unlike standard stacking, which typically requires a uniform, centrally available pool of training data, our framework treats each first-level model as a completely autonomous edge expert bound to a distinct physical node. The global model is explicitly designed to resolve the contextual domain differences and heavy class imbalances inherent to dispersed network security applications without the continuous weight-synchronization overhead seen in federated learning setups.

3.2. Second Level—Global Aggregation Model

To synthesize the outputs of the specialized local models, a distinct validation set $D_{val} = (U_{val}, A_{val}, d)$ is employed, containing objects for which all necessary conditional attributes are available. In this paper, since all local tables have the same set of features, the validation set features are the same as those from the local tables. For every object $x \in U_{val}$,

prediction vectors from all n local models are generated. To eliminate any risk of data leakage and avoid overfitting of the global classifier, a strict Out-of-Fold (OOF) prediction protocol within a stratified K -fold cross-validation framework is employed. The localized datasets D_i are partitioned into K disjoint folds. In each iteration, $K - 1$ folds are utilized to train the local specialized models. The remaining fold acts as the local validation set.

The trained local models generate prediction vectors exclusively for the instances in this unseen validation fold. These out-of-sample prediction vectors are subsequently collected across all local nodes and concatenated to construct the unified meta-training dataset D_{pred} . Formally, the validation dataset $D_{val} = \bigcup_{i=1}^n D_{val,i}$ is strictly disjoint from the data chunks used for the concurrent optimization of the first-level local parameters.

These vectors are then concatenated to form a single, unified feature vector for the global model. Consequently, a new training dataset D_{pred} is constructed:

$$D_{pred} = (U_{pred}, A_{pred}, d) \quad (3)$$

where $U_{pred} = U_{val}$ and the new attribute set A_{pred} is defined as the concatenation of all local prediction vectors:

$$A_{pred} = \{\mu_{1,1}, \dots, \mu_{1,c}, \dots, \mu_{n,1}, \dots, \mu_{n,c}\} \quad (4)$$

For each object, the tuple stored in D_{pred} takes the form:

$$[\mu_{1,1}(x), \dots, \mu_{1,c}(x), \dots, \mu_{n,1}(x), \dots, \mu_{n,c}(x), d(x)]$$

where $d(x)$ is the ground-truth class label.

A global model is subsequently trained on D_{pred} . This global network learns to weigh and interpret the confidence of each local model based on the validation data, effectively operating as a sophisticated meta-learner. In this paper, the MLP model or RF model is used as the global model.

Mathematically, the global classifier operates as a non-linear mapping function $H : \mathcal{M} \rightarrow \mathcal{Y}$, where $\mathcal{M} \in [0, 1]^{n \times c}$ denotes the continuous meta-feature space, and $\mathcal{Y} = \{1, \dots, c\}$ represents the discrete decision space of target cyberattack classes. The input to the global model for any instance x is a comprehensive vector of soft labels representing the following combined conditional probabilities:

$$\Phi(x) = [\mu_{1,1}(x), \dots, \mu_{1,c}(x), \dots, \mu_{n,1}(x), \dots, \mu_{n,c}(x)] \quad (5)$$

The objective function optimized by the global model (e.g., via cross-entropy minimization in the case of the Global MLP) effectively learns the latent correlation and dynamic reliability weights of each local expert. Instead of assuming static uniform voting, the decision logic computes the posterior probability distribution $P(d(x) = j \mid \Phi(x))$, allowing the global model to automatically down-weight ambiguous or uncalibrated local probability distributions and prioritize high-confidence expertise relevant to specific structural contexts.

The global model operates on the combined soft labels $\mu_i(x) \in [0, 1]^c$ originating from each node. This meta-feature space has a fixed dimension of $n \cdot c$, which is independent of the dimensions of the local raw features. This approach ensures that the heterogeneity of the raw features manifests solely as variability in the local reliability of the model (as reflected in the shape and calibration of μ_i). Consequently, the use of simple methods, such as local probability calibration or likelihood-based weighting of local forecasts, enables the global model to adapt to heterogeneous local representations without modifying the input data schema. The architecture is therefore model-independent about local feature spaces.

3.3. Inference and Operational Advantages

During the operational inference phase, the architecture utilizes a decentralized feed-forward communication protocol. When a new, unclassified network flow instance x arrives at a specific local gateway, its raw feature vector is distributed to the n active local nodes (or evaluated locally if nodes are structurally aligned). Crucially, the raw data vector x never leaves the secure boundaries of the local network segment where it was captured to be centralized in a master database.

Each local node processes x internally and transmits only its c -dimensional abstract prediction vector $\mu_i(x)$ to the central coordinator. While recent literature highlights theoretical vulnerabilities of probability vectors to advanced membership inference or model inversion attacks, the abstract nature of $\mu_i(x)$ provides a significantly more robust privacy-preserving barrier than the raw aggregation of detailed network packets, payload information, or structural logs.

The hierarchical architecture detailed above is presented as a generalized framework, making it highly adaptable to a wide array of classification tasks. However, as outlined earlier in this paper, our specific application focuses on cyberattack detection, for which we currently utilize MLP networks or RF at both levels of the model. This dual-level model is highly advantageous for cybersecurity applications. Because the legitimate data generated by network services is heavily dependent on the location and specific functionality of the node, building a single centralized model is often suboptimal and practically challenging. By maintaining localized MLPs or RF, the system respects physical data segregation and eliminates the overhead of continuous data harmonization. Furthermore, it allows for a robust fusion of localized expertise during the prediction phase without ever exchanging raw, potentially sensitive data.

3.4. Communication and Computational Considerations

Let n represent the number of local nodes, and c the number of decision classes. In the proposed approach, each local model generates a c -dimensional prediction vector $\mu_i(x) \in [0, 1]^c$. The global model thus receives, for each instance x , a combined meta-feature vector of dimension $n \cdot c$. In contrast to federated averaging, which requires multiple rounds of communication and the transfer of full model parameters or gradients between the central server and each client, our approach involves a single, non-iterative collection of out-of-fold prediction vectors from local models. The communication cost associated with creating the global training set is therefore proportional to:

$$\mathcal{O}(|U_{\text{pred}}| \cdot n \cdot c),$$

where $|U_{\text{pred}}|$ denotes the number of validation instances used to construct D_{pred} . The proposed approach does not require exchanging model parameter information.

During inference, each participating local node sends only its c -dimensional prediction vector for each evaluated instance, resulting in a communication size of order $\mathcal{O}(n \cdot c)$ real numbers. In typical intrusion detection system (IDS) configurations, where c is small (e.g., a binary distinction between a normal state and an attack, or a small number of attack types), this results in lower communication overhead than transmitting raw flow features or model weights.

The global model stores parameters only for a single classifier operating in the space $\mathbb{R}^{n \cdot c}$. In the experiments conducted, both n and c remain at a relatively low level, so the resulting input dimensionality is manageable for RF and MLP models. The requirements for local memory and computational power are identical to those for standalone RF/MLP classifiers trained on the relevant subsets.

The solution we propose does not require strict synchronization of model parameters. At inference time, if one or more local nodes are temporarily unavailable, the system can continue to operate using predictions from the subset of available nodes or fall back to a purely local decision.

3.4.1. Practical Communication Costs

In the proposed approach, each node sends c real numbers (soft-labels) to the coordinator for each instance; therefore, the per-instance communication cost is

$$\text{bytes}_{\text{per_instance}} = n \cdot c \cdot b,$$

where n is the number of local models, c is the number of classes, and b is the size of a floating-point number (typically 4 B for float32).

For example, with $n = 10$ and $c = 5$ we have $10 \cdot 5 \cdot 4 = 200$ B per instance. At a throughput of 1000 instances/s this corresponds to approximately 200 kB/s (about 1.6 Mb/s), which is easily achievable in typical networks.

The standard FedAvg protocol requires model weights/gradients to be transferred between the server and each client multiple times. Consider a model with S parameters (e.g., $S = 10^6$) and float32 representation ($b = 4$ B). A single exchange of weights amounts to $S \cdot b \approx 4$ MB; with $R = 100$ rounds the per-client communication is approximately $R \cdot S \cdot b \approx 400$ MB (with n clients the total traffic is $n \cdot R \cdot S \cdot b$). Even for smaller models ($S = 10^5$) and fewer rounds ($R = 20$) the costs remain on the order of tens of megabytes per client, significantly exceeding the soft-label transmission proposed here.

Practical features of the proposed approach:

- Latency: Our solution requires a single transmission of $n \cdot c$ values per instance (which can be sent asynchronously or batched), whereas FedAvg typically requires synchronization across multiple rounds, increasing total convergence time and sensitivity to network delays.
- Transmission frequency: When rapid real-time decisions are needed, sending short soft-label vectors is far more practical than repeatedly transmitting full model weights.
- Privacy and security: Gradient/weight exchange in FL carries known privacy risks (e.g., gradient attacks). Exchanging compact probabilistic vectors generally limits disclosure of raw data details, though a full adversarial/privacy analysis is planned for future work.

Numerical examples:

- Scenario A (light): $n = 10, c = 5, b = 4$ B $\Rightarrow$ 200 B per instance. For 1000 instances/s $\Rightarrow \approx 200$ kB/s ≈ 1.6 Mb/s.
- Scenario B (medium): $n = 50, c = 10, b = 4$ B $\Rightarrow 50 \cdot 10 \cdot 4 = 2000$ B per instance. For 100 instances/s $\Rightarrow \approx 200$ kB/s.
- FedAvg (comparison): $S = 10^6$ parameters, $b = 4$ B $\Rightarrow \approx 4$ MB per exchange; with $R = 50$ rounds $\Rightarrow \approx 200$ MB per client (total).

A formal framework for node failure resilience and strategies for handling missing prediction vectors (e.g., through masking, imputation, or robust meta-models) are beyond the scope of this article and form part of future work.

3.4.2. Ablation Study—OOF vs. Non-OOF

To assess the risk of leakage when constructing the global meta-training set, we performed an ablation comparing the strict Out-Of-Fold (OOF) protocol used throughout this paper with a naïve non-OOF variant (training the global meta-classifier on prediction vectors produced on the same examples used to train local models). The OOF protocol prevents

information leakage from local training into the global training set; the non-OOF variant represents a common but potentially biased alternative. The results of this comparison are reported in Section 5.1.

3.4.3. Computational Cost Model

To facilitate the estimation of computational requirements, we have used a simplified analytical model. Let S_{local} denote the number of parameters in the local model, S_{global} the number of parameters in the global model, and F_{local} and F_{global} the approximate number of floating-point operations (FLOPs) per single forward pass for the local and global models, respectively. The training cost per epoch for a local node with N_i samples can be approximated as $O(N_i \cdot F_{local})$ FLOP; the global training cost on a meta-dataset of size N_{meta} is $O(N_{meta} \cdot F_{global})$. During inference, the computations per instance amount to approximately F_{local} (local pass) + F_{global} (global pass on combined soft labels). The relationships described allow for approximate estimates and comparisons of the size of edge deployments without the need to conduct full, profiled experiments. To explain this in more detail, we can use an example based on an MLP architecture: a local MLP network $\approx$ with 1×10^5 parameters ($S_{local} \approx 1 \times 10^5$) and a global MLP network $\approx$ with 8×10^4 parameters ($S_{global} \approx 8 \times 10^4$). The typical number of FLOPs in a dense network per forward pass is $\approx 2 \cdot S$ (a multiply–accumulate operation counts as two FLOPs). Therefore:

$$\text{FLOP local_forward} \approx 2 \times S_{local} \approx 2 \times 10^5,$$

$$\text{FLOP global_forward} \approx 2 \times S_{global} \approx 1.6 \times 10^5.$$

In an example ARM core capable of performing $\sim 10^8$ FLOPs/s, the total inference per instance (local + global) takes $\approx (3.6 \cdot 10^5) / (10^8) \approx 3.6$ ms, which falls within the real-time requirements for many IDS system deployments. The number of FLOPs per epoch during training can be estimated similarly ($epochs \cdot N \cdot FLOP$). These calculations are indicative and are intended to help assess feasibility. A precise performance analysis depends on the model architecture, the framework, and the hardware.

3.4.4. Scalability of the Global Meta-Feature Dimension

The global meta-feature vector $\Phi(x)$ is a concatenation of n local soft-label vectors of length c ; hence, $\dim(\Phi) = n \cdot c$. For a global model (e.g., an MLP network with an input layer of size $n \cdot c$ and a first hidden layer of size h), the number of parameters in the first layer alone scales as $(n \cdot c) \cdot h$, i.e., it grows linearly with n . Consequently, both the memory requirements and the number of FLOPs per epoch during the training of a global model increase at least linearly with the number of nodes. The practical impact on training time and latency depends on c , the chosen sizes of the hidden layers, and whether dimension reduction or sparsity techniques are employed. To illustrate this with a practical example, let us assume that $c = 10$ classes and a global MLP network with a first hidden layer of $h = 256$ units. Increasing the number of nodes from $n = 5$ to $n = 50$ causes the size of the input data to increase from $5 \cdot 10 = 50$ to $50 \cdot 10 = 500$, so the number of parameters in the first layer increases from $50 \cdot 256 = 12,800$ to $500 \cdot 256 = 128,000$ ($\approx 10 \times$). The number of FLOPs during training and the memory requirements increase at similar rates, resulting in longer training times. To address this, the following methods can be used:

- Dimensionality reduction of meta-features (PCA, trained autoencoder);
- Grouping/coalition of local experts (hierarchical aggregation) to reduce the effective value of n ;
- The use of sparse or regularised global models;

- The use of very lightweight global learning algorithms (logistic regression or small ensembles of trees) for large values of n .

By applying these techniques, it is possible to control the increase in computational load.

4. Experimental Data and Environment Setup

This section describes the datasets used in the experiments and outlines the problem the article's authors set out to solve. Detecting all types of anomalies in network traffic that may constitute attempts to attack IT systems is currently one of the most significant issues, given the ever-increasing risk of attacks and their potential consequences. Data derived from network traffic analysis is quite specific, both in terms of the volume of operations on the network and the significant imbalance. In real-world computer networks, the vast majority of network traffic relates to legitimate operations; traffic that could constitute a cyberattack represents only a small fraction of the total. Furthermore, the nature of such traffic depends largely on location, or more precisely, on the services operating within a given network. It is these services that generate legitimate network traffic; conversely, they may also be the target of potential attacks on the infrastructure. The proper functioning of security systems can protect an organization from serious financial and reputational losses. The main premise of the research was that, since data from network traffic analysis will vary significantly depending on the location where the traffic is monitored, attempting to build a single centralized model that works well in every case may be quite difficult; it seems a better idea to create local models and, only on that basis, to create a single global model.

The study used two commonly used datasets containing information on both normal network traffic and various types of cyberattacks. The datasets selected were CIC-DDoS2019 and CTU-13.

In the experiments we have conducted, we use scenario-based classifications derived from publicly available datasets. Ultimately, the two-tier architecture we propose has been designed with deployment in multi-branch environments in mind; in our approach, the global model is based exclusively on calibrated forecast vectors provided by local experts. In practice, differences between locations—including, amongst other things, traffic composition, protocol sets and log formats—primarily affect the quality of local forecasts rather than the dimensionality of the meta-feature space. Consequently, the global model utilizes heterogeneous local knowledge, provided that appropriate calibration is ensured for each node.

4.1. CIC-DDoS2019 Dataset

The first dataset is the CIC-DDoS2019 dataset [32] (Canadian Institute for Cybersecurity Distributed Denial-of-Service 2019). This dataset was developed for research into the detection of distributed denial-of-service (DDoS) attacks. The authors' aim in developing this dataset was to create a new benchmark dataset that would include new types of attacks and provide a realistic representation of network traffic. The dataset contains information on both normal traffic (labeled as 'benign') and many types of modern DDoS attacks carried out mainly at the application layer using TCP/UDP protocols. The dataset includes, amongst others, reflection attacks utilizing protocols and services such as PortMap, NetBIOS, LDAP, MSSQL, NTP, DNS, SNMP, SSDP, and TFTP, as well as exploit-based attacks, including SYN flood, UDP flood, and the UDP-Lag variant characteristic of, amongst others, online games (disrupting client-server communication through artificial delay or link saturation). Additionally, WebDDoS traffic and port scanning (PortScan) were included. An important element of the dataset was obtaining network traffic as similar as possible to real-world network traffic. For the research in this article, processed network flows

were used, one of the three data types provided by the dataset's authors. This dataset contains over 80 statistical features describing traffic behavior (including packet count, size distribution, inter-arrival times, flow direction, and session-level parameters). For each flow, labels are available specifying the traffic type (benign vs. a specific attack class), as well as basic metadata: timestamps, source and destination IP addresses, source and destination ports, and the protocol used. For this article, we decided to adapt the CIC-DDoS2019 dataset to additionally simulate different locations (network nodes). Based on the full dataset, we created new subsets, organized so that each file contained both normal network traffic and all attack types. Differences in the number of individual attack types for a given subset are a distinguishing feature of that subset (the given node). In the approach for the CIC-DDoS2019 dataset, the proposed division is a simulation rather than a direct representation of separate network locations. All local subsets have the same set of attributes and contain all classes; they differ only in the distribution of attack types. Our approach simplifies by introducing variation in the number of classes at individual nodes, but does not fully capture possible differences in service configuration and feature distributions.

4.2. CTU-13 Dataset

The second dataset used in the experiments is the CTU-13 dataset [33], one of the fundamental datasets used in research on the detection of malicious network traffic. The dataset was developed in 2011 at the Czech Technical University in Prague (CTU, Czech Republic). The main objective of creating the dataset was to obtain a large collection of real botnet traffic, mixed with normal user traffic and background traffic. The CTU-13 dataset consists of 13 scenarios in which various types of botnets were deployed, utilizing different protocols and performing different activities (including C&C communication, scanning, spam, and DoS/DDoS attacks). Each scenario reflects a different use case for a botnet, with its own traffic characteristics and a different ratio of malicious to legitimate traffic. Consequently, each scenario differs from the others in both the types of attacks and their share of total network traffic. Importantly, the data in this dataset originates from monitoring the university network where the dataset was created, so it very accurately reflects real-world network traffic. Furthermore, it has been designed to include both simple cases and complex ones that are harder to detect due to their specific nature or their small share of the total traffic. In our experiments, we treated each scenario as local data originating from different locations. Each scenario corresponds to a different type of botnet, a background traffic profile, and a ratio of malicious to legitimate traffic. This approach provides a reasonable equivalent to distinct logical network segments, where different services and attack vectors are observed. However, this mapping still reflects a more logical concept of a node, rather than traffic recorded by physically separate sensors at different locations across the network.

4.3. Problem Definition

The proposed approach involves a two-level architecture, with the first stage consisting of building local models. Instead of the traditional approach, where we have a single central model trained on aggregate data, our approach assumes multiple nodes (in different locations), each with its own dataset and training its own local models. Only in the next step do we build a global model trained on predictions from the local models. The motivation for this approach is that it better suits real-world environments, which have varying structures and network segmentations, leading to challenges in data integration and in constructing a centralized model. Furthermore, this may also involve constraints related to legal regulations or privacy protection. Network traffic information

can be helpful in planning an attack, as it reveals which services are running on the network. Another advantage of this approach is that different network nodes may generate distinct types of traffic and, consequently, face distinct types of attacks, enabling local models to learn distinct information. Formally, we have n local nodes. Node i has a set $\mathcal{D}_{(i)} = \{(x_k^{(i)}, y_k^{(i)})\}$, where $x_k^{(i)}$ is a flow feature vector and $y_k^{(i)}$ is a label indicating whether the traffic is legitimate or constitutes a specific attack. For the CIC-DDoS2019 dataset, local models are built using artificially generated locations. Within a single file, the dataset itself contains only information on legitimate traffic and one specific type of attack. For the experiments, we created local datasets so that each dataset contained all classes, but in different proportions (one type of attack dominates in one file, another in the next). This approach allowed us to simulate network traffic for different network nodes. For the CTU-13 dataset, the original partition was used to build local models, treating each day of data collection as a dataset from a different node. At the local model level, we train a classifier h_i , which, for an object x , returns the probability distribution across classes (the vector $\mu_i(x)$). Next, for the global model, instead of providing raw features, we construct a feature vector:

$$\mu(x) = [\mu_{1,1}(x), \dots, \mu_{n,c}(x)],$$

We concatenate the prediction vectors of all local models into a single vector. The global model H is trained on the ‘opinions’ of the local models, rather than on the raw features. This approach has several advantages from the perspective of cybersecurity and privacy protection. Firstly, the global model does not need direct access to packets or traffic details. Instead, it receives predictions. The second advantage is the ability to add custom features to each node and even different types of models. In this article, the term ‘distributed’ should therefore be understood primarily to refer to heterogeneously distributed data sources (with different class distributions and traffic contexts), rather than solely to physically isolated network nodes. Research into the proposed approach, based on truly distributed network monitoring points, remains an important area for future work.

5. Experiments Results

In this section, we presented and discussed the results of the experiments conducted. The experiments aimed to evaluate the effectiveness of an approach based on a global model constructed from local models; the results were compared with those of the classic centralized approach. In the first part, we presented the results obtained on the CIC-DDoS2019 dataset, starting with hyperparameter tuning, followed by testing random forest and MLP neural network models.

5.1. Ablation: The OOF Protocol Versus the NON-OOF Protocol

As part of our research, we investigated the impact of the Out-Of-Fold (OOF) protocol on a global MLP meta-classifier, using the CIC-DDoS2019 dataset. The experimental setup reflects the main processing pipeline: local MLP models were trained on synthetic local partitions, whilst the global MLP model was trained on aggregated local forecast vectors. We compared the following:

- OOF_5fold: global training data constructed using exact 5-fold OOF forecasts from local models (no leakage);
- NON_OOF (train==test): a global model trained on forecast vectors generated by local models trained on the same instances (possible leakage).

Table 1 presents the aggregated metrics obtained on the CIC-DDoS2019 dataset.

Table 1. OOF vs NON-OOF—CIC-DDoS2019 (global MLP model).

Protocol	Weighted F1	Accuracy	Weighted Precision	Weighted Recall
OOF_5fold	0.9242	0.9348	0.9355	0.9348
NON_OOF_train_equals_test	0.9227	0.9362	0.9394	0.9362

The difference in the aggregated, weighted F1 score between these two protocols is small (0.0015) for the CIC-DDoS2019 dataset. An advantage of the OOF protocol is that it provides a stable, verified cross-validation method for assessing generalization (mean F1 score = 0.9236, standard deviation = 0.0016). In contrast, the configuration without OOF exhibits higher apparent accuracy on the training data but does not provide objective statistics for individual local components. Consequently, although the numerical differences for the CIC-DDoS2019 dataset are small, the OOF protocol protects against optimistic bias and overfitting. Based on the results, we decided to retain the OOF protocol for all global model training processes described in this paper.

5.2. DDoS Dataset

This section presents the results of experiments conducted on the CIC-DDoS2019 dataset. This dataset was selected because it contains real-world network traffic featuring DDoS/DoS attacks of varying intensity, service types, and traffic generation scenarios. The large number of samples and the presence of many attack classes allow for a realistic assessment of the models' behavior under severe imbalance and in situations where different types of attacks dominate.

5.2.1. Models Based on Random Forests

This section presents the results of experiments on constructing random forest models; subsequent sections will present and discuss the results of hyperparameter tuning and the model-building process itself, covering both the global approach—based on local models—and the centralized approach.

Tuning the Model's Hyperparameters

Table 2 presents the results of hyperparameter tuning for the random forest method applied to a centralized model on the CIC-DDoS2019 dataset using the RandomSearchCV method (the best configuration is shown in bold); similarly, Table 3 shows the results of hyperparameter tuning for the global model based on local models. In both cases, the objective function was the weighted F1 score calculated using 3-fold cross-validation. The results are sorted by descending F1 score, with the score and its standard deviation shown alongside.

The result obtained for the best configuration was $meanF1 = 0.934 \pm 0.001$. It can also be observed that other configurations achieve F1 scores very similar to those of the best configuration, despite differences in hyperparameter settings. These results suggest that the centralized method is relatively robust to small changes in hyperparameters. Another conclusion is that there are no significant benefits from increasing the depth or number of trees; the model achieves similar results with 400 trees as with 100. In practice, it is possible to reduce complexity and training time without any significant loss of quality.

Table 2. Results of hyperparameter tuning for the random forest method on the DDoS dataset—centralised model.

No. Estim.	Max Depth	Min Samples Split	Min Samples Leaf	Max Features	Bootstrap	Mean F1 $\pm$ Sd
400	40	2	4	sqrt	True	0.934 $\pm$ 0.001
200	20	10	1	sqrt	True	0.934 $\pm$ 0.001
100	40	10	2		True	0.934 $\pm$ 0.001
50	20	5	2	sqrt	True	0.934 $\pm$ 0.001
50		2	4	sqrt	False	0.934 $\pm$ 0.001
200	40	10	2	log2	False	0.933 $\pm$ 0.001
200		5	2	log2	True	0.933 $\pm$ 0.001
200	40	2	2	sqrt	True	0.933 $\pm$ 0.001
50	20	10	4	log2	False	0.933 $\pm$ 0.001
50	20	2	2	log2	False	0.933 $\pm$ 0.001
400	20	2	1		True	0.932 $\pm$ 0.001
200	40	5	2	sqrt	False	0.932 $\pm$ 0.001
200		2	2	log2	False	0.931 $\pm$ 0.001
100	40	5	1	log2	False	0.930 $\pm$ 0.001
200	40	2	1	log2	False	0.927 $\pm$ 0.001
50	40	2	1	log2	False	0.928 $\pm$ 0.001
200	10	5	2	sqrt	False	0.924 $\pm$ 0.001
200	10	5	4	sqrt	True	0.924 $\pm$ 0.001
100	10	5	2	log2	True	0.923 $\pm$ 0.001
400	10	10	2	log2	True	0.923 $\pm$ 0.001

For the global model based on local models (Table 3), the best result obtained was $meanF1 = 0.940 \pm 0.001$. As with the centralized model, the results for the best configurations do not differ significantly; they have similar F1 scores despite differing configurations. However, the results obtained are better than those for the centralized model. Based on the results, the global model, which operates on features derived from local model predictions, achieves a very high weighted F1 score. This means that the local model's prediction vectors contain sufficient information for the global model to distinguish between classes effectively. Similarly to the centralized model, a fairly wide range of optimal hyperparameters can be observed—different configurations yield similar results. Another important point is that the best configuration has shallower depth and fewer trees than the centralized model. For both the centralized and global RF models, the hyperparameter space contains many good solutions, meaning that very precise hyperparameter tuning is not required. Furthermore, both approaches achieve very high F1 scores, indicating that both models can be effectively used in predicting cyberattacks.

Main RF Experiments on CIC-DDoS2019 Dataset

This section presents the results of experiments designed to verify whether the proposed hierarchical model (a combination of a global and local model) achieves better model quality than the classic centralized model. Figure 2 shows the distribution of weighted F1 scores across the individual cross-validation folds.

From Figure 2, it can be seen that the difference between the approaches is small—both methods achieve scores above 0.93—but the hierarchical model has a median that is slightly shifted upwards and a slightly higher upper quartile. The hierarchical model also shows greater stability (lower standard deviation). Table 4 presents the values of the metrics used to evaluate the models for the centralized and hierarchical models.

Table 3. Results of hyperparameter tuning for the random forest method on the CIC-DDoS2019 dataset—global model.

No. Estim.	Max Depth	Min Samples Split	Min Samples Leaf	Max Features	Bootstrap	Mean F1 $\pm$ Sd
100	30	10	4	0.5	True	0.940 ± 0.001
300	30	10	4	0.5	True	0.940 ± 0.001
500	30	10	4	0.5	True	0.940 ± 0.001
200	30	10	2	0.5	True	0.939 ± 0.001
100	40	10	4	sqrt	True	0.939 ± 0.001
300	10	2	4	1	True	0.939 ± 0.001
200	40	10	4	sqrt	True	0.939 ± 0.001
200	10	2	4	1	True	0.939 ± 0.001
500	10	2	4	1	True	0.939 ± 0.001
500	30	10	4	sqrt	True	0.939 ± 0.001
200	40	10	4	0.5	False	0.938 ± 0.001
500	40	10	4	0.5	False	0.938 ± 0.001
100	10	2	4	1	True	0.938 ± 0.001
300	10	2	2	sqrt	False	0.938 ± 0.001
100	40	2	4	0.5	False	0.938 ± 0.001
300	40	10	4	0.5	False	0.938 ± 0.001
300	40	2	4	0.5	False	0.938 ± 0.001
100	10	2	2	sqrt	False	0.938 ± 0.001
300	40	10	2	0.5	False	0.938 ± 0.001
500	40	10	2	0.5	False	0.938 ± 0.001
500	10	2	2	sqrt	True	0.937 ± 0.001
300	10	5	2	sqrt	False	0.937 ± 0.001
300	10	10	4	sqrt	True	0.937 ± 0.001
200	10	10	2	log2	False	0.937 ± 0.001
300	40	10	2	sqrt	True	0.937 ± 0.001
100	10	10	4	log2	True	0.937 ± 0.001
500	40	10	2	sqrt	True	0.937 ± 0.001
100	10	5	4	log2	True	0.937 ± 0.001
500	10	10	4	log2	True	0.937 ± 0.001
500	30	2	2	sqrt	False	0.935 ± 0.001
100	40	5	2	log2	False	0.935 ± 0.001
300	10	10	4	log2	False	0.935 ± 0.001

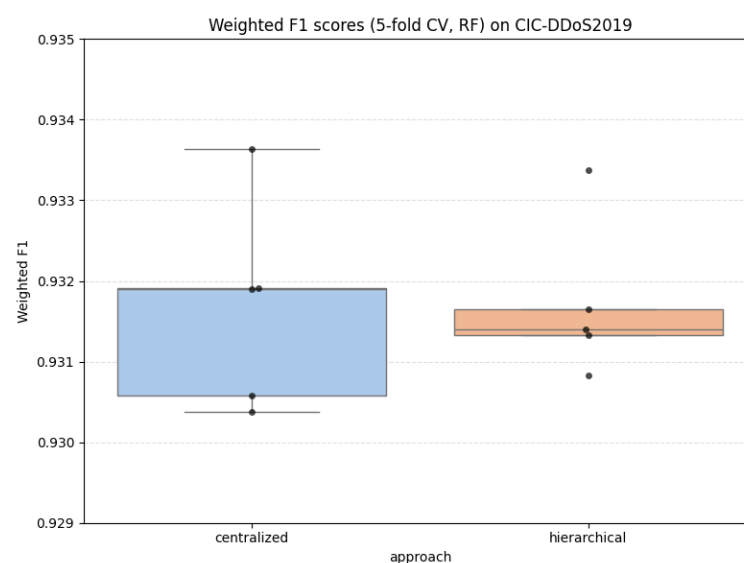
**Figure 2.** The average weighted F1 score for the centralised and hierarchical models based on local models using random forest.

Table 4. Quality assessment metrics for models (a centralised model and hierarchical model based on local models) for the DDoS dataset and the random forest method.

Approach	Accuracy Mean $\pm$ Sd	Precision Weighted Mean $\pm$ Sd	Recall Weighted Mean $\pm$ Sd	F1 Weighted Mean $\pm$ Sd
centralize	0.934 $\pm$ 0.001	0.933 $\pm$ 0.001	0.934 $\pm$ 0.001	0.932 $\pm$ 0.001
hierarchical	0.938 $\pm$ 0.001	0.937 $\pm$ 0.002	0.938 $\pm$ 0.001	0.932 $\pm$ 0.001

From Table 4, it can be seen that both approaches achieve very high values for the classical quality metrics. The hierarchical model achieves slightly higher values for accuracy, recall-weighted, and precision-weighted, whilst the F1-weighted value is very similar for both models. This indicates that both RF models perform well in classifying the CIC-DDoS2019 dataset; although the difference is not significant, the proposed hierarchical approach achieved better model evaluation metrics.

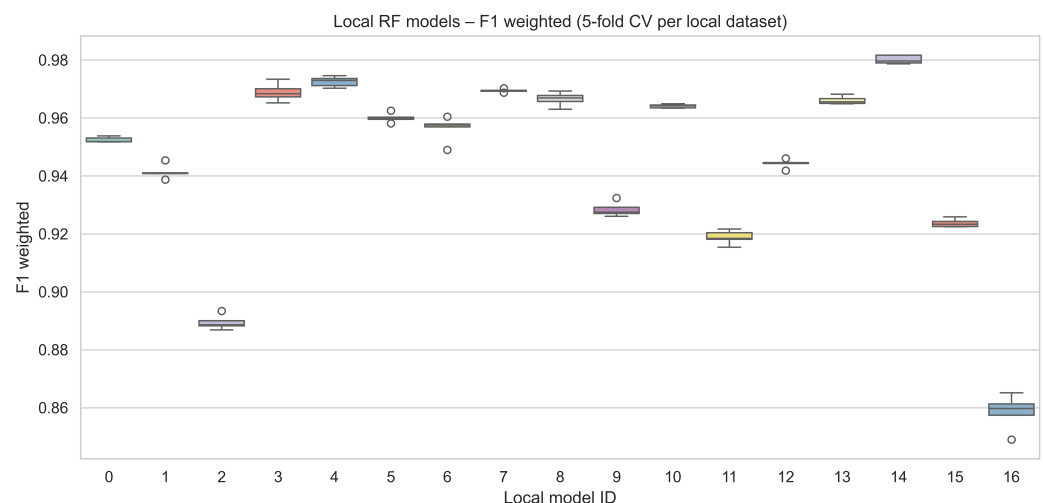
Table 5 shows the values of Cohen’s kappa coefficient and Matthews’ correlation coefficient.

Table 5. Cohen’s kappa and Matthews correlation coefficient (centralised and hierarchical, based on local models) for the DDoS dataset and the random forest method.

Approach	Cohen’s Kappa Mean $\pm$ Sd	Matthews Corrcoef Mean $\pm$ Sd
centralized	0.917 $\pm$ 0.001	0.917 $\pm$ 0.001
hierarchical	0.922 $\pm$ 0.001	0.923 $\pm$ 0.001

For both approaches—centralized and hierarchical—the Kappa coefficient is very high and practically equal to the accuracy, indicating that the agreement between the predictions and the true labels is not random. The same applies to the interpretation of the MCC value: its high value for both models indicates that they perform well.

Figure 3 shows the distribution of the weighted F1 score for the local models used to construct the hierarchical model.

**Figure 3.** The F1 score for local models in a hierarchical RF model.

In the graph shown in Figure 3, it can be seen that individual local models achieve varying performance levels. Some locations have an F1 score significantly above the average, whilst others are clearly lower. This is to be expected, given the different class distributions in the local datasets (in some, a specific type of attack dominates, whilst in

others it is less prevalent). From the perspective of the global model, it is important that these local models perform correctly on local scenarios and that their predictions provide diverse information that the hierarchical model can utilize. This explains why a hierarchical RF based on local models can achieve at least the same quality as a centralized model, even though it does not operate directly on raw features but only on probability vectors.

Based on the results obtained from the RF model on the CIC-DDoS2019 dataset, it can be seen that the centralised model achieves very high performance, confirming that RF is a sound baseline method for this dataset; the hierarchical model, based on local models, performs equally well on this dataset, achieving even higher evaluation metrics.

5.2.2. Models Based on MLP

This section presents the results of experiments on constructing MLP models; subsequent sections will present and discuss the results of hyperparameter tuning and the model-building process itself, covering both the global approach—based on local models—and the centralized approach.

Tuning the Hyperparameters of MLP Models

Table 6 presents the results of hyperparameter tuning for the MLP method for a centralized model trained on the CIC-DDoS2019 dataset (The best configuration is highlighted in bold), whilst Table 7 contains the corresponding results for a global model based on local models. In both cases, the RandomizedSearchCV method with 3-fold cross-validation was used to search for configurations, and the objective function was the weighted F1 score. The tables present the tested configurations, sorted in descending order by the mean F1 score.

Table 6. Results of hyperparameter tuning for the MLP method on the DDoS dataset—centralised model.

Hidden Layer	Alpha	Learning_Rate_Init	Batch Size	Learning Rate	Mean F1 $\pm$ Sd
(64; 32)	0.0001	0.001	64	adaptive	0.909 $\pm$ 0.002
(128; 64)	0.001	0.001	128	adaptive	0.909 $\pm$ 0.001
(64;)	0.001	0.001	64	adaptive	0.908 $\pm$ 0.001
(128;)	0.0001	0.001	64	adaptive	0.906 $\pm$ 0.004
(128;)	1.00×10^{-5}	0.001	128	adaptive	0.906 $\pm$ 0.005
(64;)	1.00×10^{-5}	0.0005	64	adaptive	0.906 $\pm$ 0.002
(64; 32)	1.00×10^{-5}	0.0005	128	constant	0.905 $\pm$ 0.002
(128;)	0.001	0.0005	128	adaptive	0.905 $\pm$ 0.002
(64;)	0.001	0.0005	128	adaptive	0.904 $\pm$ 0.002
(64;)	1.00×10^{-5}	0.0005	128	adaptive	0.904 $\pm$ 0.001
(128; 64)	0.001	0.0001	64	adaptive	0.903 $\pm$ 0.002
(128; 64)	1.00×10^{-5}	0.0001	128	adaptive	0.902 $\pm$ 0.001
(64; 32)	0.0001	0.0001	64	adaptive	0.899 $\pm$ 0.001
(128;)	0.01	0.0001	128	adaptive	0.892 $\pm$ 0.002
(64;)	1.00×10^{-5}	0.0001	128	constant	0.889 $\pm$ 0.002

In Table 6, the best result was achieved by the network configuration with two hidden layers, a batch size of 64, and a learning rate set to ‘adaptive’. For this configuration, a weighted F1 score of 0.909 was obtained. It can also be noted that for other configurations—both with one and two hidden layers—similar results were obtained (F1 at the level of ≈ 0.904 –0.909). As with the RF model, this means that various configurations of MLP hyperparameters on the DDoS dataset yield high-quality results, and minor changes to the parameters (e.g., the number of neurons, the value of α) do not cause a dramatic drop in quality. Another conclusion is that increasing the network’s complexity does not yield a

significant quality improvement, suggesting that the problem of representing raw features does not require very deep or wide networks.

Table 7. Results of hyperparameter tuning for the MLP method on the DDoS dataset—a global model based on local models.

Hidden Layer Sizes	Alpha	Learning Rate Init	Batch Size	Learning Rate	Mean F1 $\pm$ Sd
(64;)	1.00×10^{-5}	0.0001	128	constant	0.937 ± 0.002
(128;)	0.001	0.0001	128	adaptive	0.937 ± 0.001
(64; 32)	0.0001	0.0001	64	adaptive	0.936 ± 0.001
(128; 64)	1.00×10^{-5}	0.0001	128	adaptive	0.936 ± 0.001
(128; 64)	0.001	0.0001	64	adaptive	0.935 ± 0.001
(64; 32)	1.00×10^{-5}	0.0005	128	constant	0.934 ± 0.001
(128;)	0.001	0.0005	128	adaptive	0.934 ± 0.001
(64;)	1.00×10^{-5}	0.0005	128	adaptive	0.934 ± 0.002
(64;)	1.00×10^{-5}	0.0005	64	adaptive	0.933 ± 0.001
(64;)	0.001	0.0005	128	adaptive	0.933 ± 0.002
(64;)	0.001	0.001	64	adaptive	0.933 ± 0.002
(64; 32)	0.0001	0.001	64	adaptive	0.933 ± 0.001
(128;)	0.0001	0.001	64	adaptive	0.933 ± 0.001
(128;)	1.00×10^{-5}	0.001	128	adaptive	0.933 ± 0.001
(128; 64)	0.001	0.001	128	adaptive	0.933 ± 0.001

Table 7 presents the results of hyperparameter tuning for the global MLP model, which takes as input the prediction vectors generated by the local models. The best result achieved was an F1 score of 0.937. This is a better result than that of the best centralized model. Furthermore, as with RF, the best configurations include a whole range of settings with very similar F1 scores, differing both in architecture (networks with one and two hidden layers) and in regularisation and learning parameters. Based on the results, the global MLP trained on the prediction vectors of local models achieves a higher weighted F1 score than the best centralized MLP trained directly on raw features. This means that merely aggregating information from local models yields a better representation than raw feature vectors. As with RF, many hyperparameter configurations yield very similar results. From a practical point of view, highly complex hyperparameter tuning is not necessary to achieve a high-quality global model. Another advantage of the global model is that the best configuration is not complex, yet it performs very well on prediction vectors. This suggests that the problem of aggregating the ‘opinions’ of local models is structurally simpler than direct classification at the level of raw features.

Comparing Tables 6 and 7, it can be seen that the difference between the best global MLP result and the best centralized MLP result is significant. This indicates that aggregating local model predictions can capture information that a single centralized MLP cannot fully utilize, despite having the full feature set at its disposal.

Later in this chapter, these results are compared with full validation experiments (5-fold CV) and contrasted with a line of RF-based experiments, allowing us to draw more general conclusions regarding the suitability of local model-based architectures for detecting cyberattacks.

Main MLP Experiments on CIC-DDoS2019 Dataset

This section presents the results of experiments evaluating the quality of MLP models obtained using the proposed hierarchical approach based on local models and a comparative centralized model.

Figure 4 shows the distribution of F1 scores for individual folds in a 5-fold cross-validation for the hierarchical method and the centralized method.

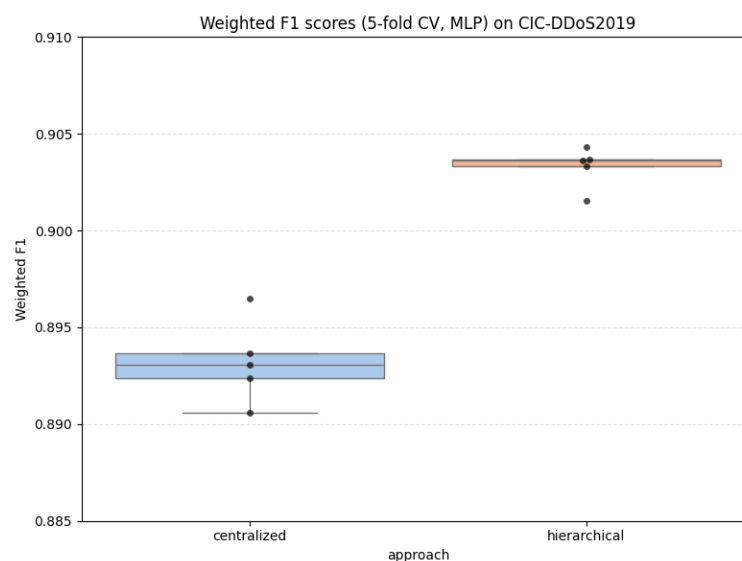


Figure 4. The average weighted F1 score for the centralised and hierarchical models based on local models using MLP model.

From Figure 4, it can be seen that, in each fold, the hierarchical model maintains results above those achieved by the centralized model, and the spread of results (standard deviations) is small in both cases. Table 8 presents the values of the quality assessment metrics for the hierarchical model and the centralized model.

Table 8. Quality assessment metrics for models (a centralised model and hierarchical model based on local models) for the DDoS dataset and the MLP model.

Approach	Accuracy Mean $\pm$ Sd	Precision Weighted Mean $\pm$ Sd	Recall Weighted Mean $\pm$ Sd	F1 Weighted Mean $\pm$ Sd
centralized	0.908 $\pm$ 0.003	0.899 $\pm$ 0.005	0.908 $\pm$ 0.003	0.893 $\pm$ 0.002
hierarchical	0.917 $\pm$ 0.001	0.909 $\pm$ 0.002	0.917 $\pm$ 0.001	0.903 $\pm$ 0.001

From Table 8, it can be seen that the proposed hierarchical model-based approach achieves higher scores on all metrics used to evaluate model quality. The differences for the CIC-DDoS2019 dataset are not particularly large, but for the MLP model they are greater than in previous experiments where RF was used to build the models. The proposed hierarchical approach yields greater benefits for the MLP than the RF approach, indicating that the global model better leverages information from local models than when trained on the entire dataset.

Table 9 contains the values of Cohen's kappa coefficient and Matthews' correlation coefficient (MCC) for both approaches.

Table 9. Cohen's kappa and Mathews coefficients (centralised and hierarchical, based on local models) for the DDoS dataset and the MLP model.

Approach	Cohen Kappa mean $\pm$ Sd	Matthews Corrcoef Mean $\pm$ Sd
centralized	0.884 $\pm$ 0.004	0.885 $\pm$ 0.004
hierarchical	0.896 $\pm$ 0.002	0.897 $\pm$ 0.002

The values of Cohen’s kappa and MCC are higher for the proposed hierarchical approach. In the case of these measures, they are more ‘conservative’ than simple accuracy—they take into account not only the number of correct predictions, but also the distribution of errors and class proportions. The increase in their values for the hierarchical model confirms that the quality improvement is not the result of a random fit to the dominant class, but stems from better alignment of predictions with the actual classes.

Another important observation is that the standard deviations for both metrics are smaller in the hierarchical model than in the centralised one, suggesting greater stability of results across folds. This can be interpreted as the effect of averaging information from multiple local models. Even if one of the local nodes performs worse on a given fold, the global network still benefits from the other nodes’ predictions, thereby reducing sensitivity to errors in individual scenarios.

5.3. CTU13 Dataset

Another set of experiments was conducted on the second dataset, CTU-13. As with the CIC-DDoS2019 dataset, experiments were conducted to verify whether the proposed hierarchical solution, based on local models, would perform better at classifying cyber threats than the traditional approach based on a centralized model. As before, RF and MLP methods were used to build the models.

5.3.1. Model Based on Random Forest

This section presents the results of experiments conducted on the CTU-13 dataset using a random forest model.

Selection of Hyperparameters for the Random Forest Model

Tables 10 and 11 present the results of hyperparameter tuning for the Random Forest method, for the centralized model and the global model (trained on the predictions of local models), carried out on the CTU-13 dataset (the best configuration is shown in bold). In both cases, as with the CIC-DDoS dataset, the F1-weighted score, calculated using 3-fold cross-validation, was adopted as the objective function in RandomizedSearchCV. The tables show the best configurations found, sorted in descending order by the mean F1 score (with standard deviation).

Table 10. Results of hyperparameter tuning for the random forest method on the CTU13 dataset—centralised model.

No. Estimators	Max Depth	Min Samples Split	Min Samples Leaf	Max Features	Bootstrap	Mean F1 $\pm$ Sd
300	–	2	2	0.5	False	0.866 $\pm$ 0.002
200	20	2	1	0.5	False	0.866 $\pm$ 0.001
100	30	2	1	sqrt	True	0.865 $\pm$ 0.002
200	40	5	1	sqrt	False	0.865 $\pm$ 0.001
200	40	10	1	sqrt	True	0.864 $\pm$ 0.002
400	40	2	1	0.5	False	0.863 $\pm$ 0.001
100	20	10	4	0.5	False	0.863 $\pm$ 0.001
500	20	2	2	sqrt	True	0.862 $\pm$ 0.002
100	30	10	4	0.5	True	0.862 $\pm$ 0.001
200	30	10	4	0.5	True	0.861 $\pm$ 0.002
500	–	2	4	log2	False	0.861 $\pm$ 0.002
200	–	2	4	log2	True	0.858 $\pm$ 0.002
400	20	2	4	sqrt	True	0.857 $\pm$ 0.001
200	20	2	4	log2	True	0.857 $\pm$ 0.001
100	10	2	2	0.5	False	0.851 $\pm$ 0.003

In the case of the centralized RF model on the CTU-13 dataset (Table 2), the best configuration achieved a mean F1 score of 0.866. It can also be observed that, as in the CIC-DDoS dataset, other hyperparameter configurations yielded very similar F1 scores. These results suggest that, for the CTU-13 dataset, the centralized RF model is fairly robust to changes in hyperparameters: across various configurations, the model achieves comparable performance. From a practical point of view, this means that it is not necessary to fine-tune all parameters with great precision.

Table 11. Results of hyperparameter tuning for the random forest method on the CTU13 dataset—a global model based on local models.

No. Estimators	Max Depth	Min Samples Split	Min Samples Leaf	Max Features	Bootstrap	Mean F1 $\pm$ Sd
300	–	2	2	0.5	False	0.910 $\pm$ 0.002
400	40	2	1	0.5	False	0.909 $\pm$ 0.002
200	20	2	1	0.5	False	0.906 $\pm$ 0.002
100	20	10	4	0.5	False	0.906 $\pm$ 0.002
200	30	10	4	0.5	True	0.905 $\pm$ 0.002
100	30	10	4	0.5	True	0.905 $\pm$ 0.001
100	10	2	2	0.5	False	0.893 $\pm$ 0.002
200	40	5	1	sqrt	False	0.889 $\pm$ 0.001
200	40	10	1	sqrt	True	0.887 $\pm$ 0.001
100	30	2	1	sqrt	True	0.886 $\pm$ 0.001
500	20	2	2	sqrt	True	0.881 $\pm$ 0.001
400	20	2	4	sqrt	True	0.878 $\pm$ 0.002
500	–	2	4	log2	False	0.849 $\pm$ 0.003
200	–	2	4	log2	True	0.845 $\pm$ 0.003
200	20	2	4	log2	True	0.839 $\pm$ 0.003

Table 11 shows the results of tuning a global RF model based on local models. The best global RF configuration matches the best centralized model on key parameters but achieves a higher F1 score of 0.910. As with the centralized ensemble, other hyperparameter configurations yield similar F1 scores, indicating that the results are not very sensitive to minor changes in hyperparameters. The main conclusion that can be drawn from the results is that the global RF, which operates on features derived from local model predictions, achieves significantly better F1 scores than the centralized RF on raw features (an increase from approx. 0.866 to approx. 0.910), which shows that the global model performs better on the CTU-13 dataset.

Main RF Experiments on CTU-13 Dataset

This section presents the results of the quality assessment of the hierarchical and centralized RF models for the CTU-13 dataset.

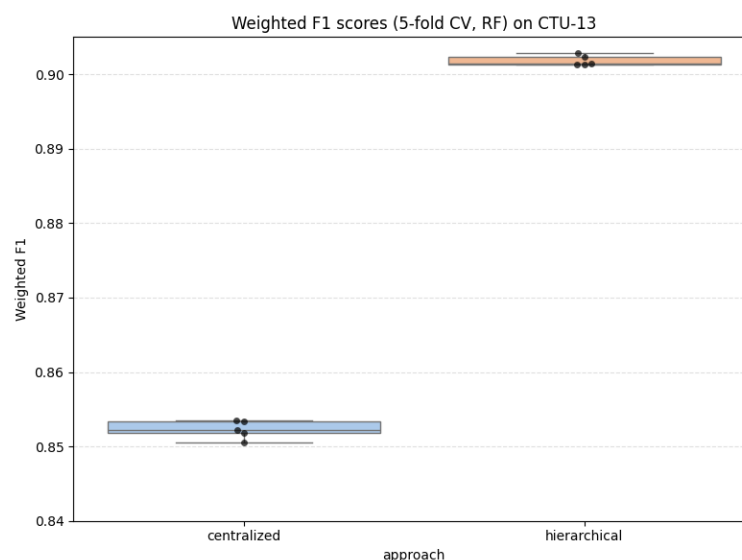
Figure 5 shows the distribution of the weighted F1 score across the cross-validation folds for the hierarchical and centralized models.

Figure 5 confirms that the proposed hierarchical model performs better than the classical model. The spread of the results is very small in both cases.

Table 12 shows that the differences between the proposed hierarchical approach and the centralized approach for the CTU dataset under the RF model are significantly greater than for the DDoS dataset. The global model based on local models achieves an F1-weighted increase of almost 6 percentage points over the best centralized RF model.

Table 12. Quality assessment metrics for models (a centralised model and hierarchical model based on local models) for the CTU-13 dataset and the RF model.

Approach	Accuracy Mean $\pm$ Sd	Precision Weighted Mean $\pm$ Sd	Recall Weighted Mean $\pm$ Sd	F1 Weighted Mean $\pm$ Sd
centralized	0.874 $\pm$ 0.001	0.849 $\pm$ 0.002	0.874 $\pm$ 0.003	0.852 $\pm$ 0.002
hierarchical	0.893 $\pm$ 0.001	0.894 $\pm$ 0.002	0.900 $\pm$ 0.001	0.910 $\pm$ 0.001

**Figure 5.** The average weighted F1 score for the centralised and hierarchical models based on local models using random forest.

These results suggest that, for the CTU-13 dataset, the hierarchical architecture offers a marked improvement over the traditional centralized approach. CTU-13 comprises many heterogeneous scenarios with different botnets and traffic backgrounds, and the centralized RF combines all these scenarios into a single dataset, in which certain patterns may partially overlap or mask one another. In the hierarchical approach, local models learn separately for individual scenarios and can specialize in detecting specific behaviors for a given traffic type. The global model then learns how to combine its predictions—as a result, it utilizes the structure of the scenarios that the centralized model cannot see.

Table 13 contains the values of Cohen’s kappa coefficient and Matthews’ correlation coefficient for both approaches. Although the numerical differences in the coefficients are smaller than in the case of weighted F1, they are consistent and show that the quality improvement applies to the entire confusion matrix, rather than just a single class type or decision threshold.

Table 13. Cohen’s kappa and Matthews coefficients (centralised and hierarchical, based on local models) for the CTU-13 dataset and the RF model.

Approach	Cohen Kappa Mean $\pm$ Sd	Matthews Corcoef Mean $\pm$ Sd
centralized	0.839 $\pm$ 0.001	0.840 $\pm$ 0.001
hierarchical	0.848 $\pm$ 0.002	0.851 $\pm$ 0.001

Figure 6 shows the distribution of F1 scores for the local models used in the hierarchical model. It clearly shows that the quality of the local models varies: some local datasets achieve very high F1 scores, whilst others perform less well. This reflects the differences between the CTU-13 scenarios—in some, attacks are easier to distinguish, whilst in others,

there is greater chaos or a significant class imbalance. It is important to note that the global model handles this and achieves good classification performance.

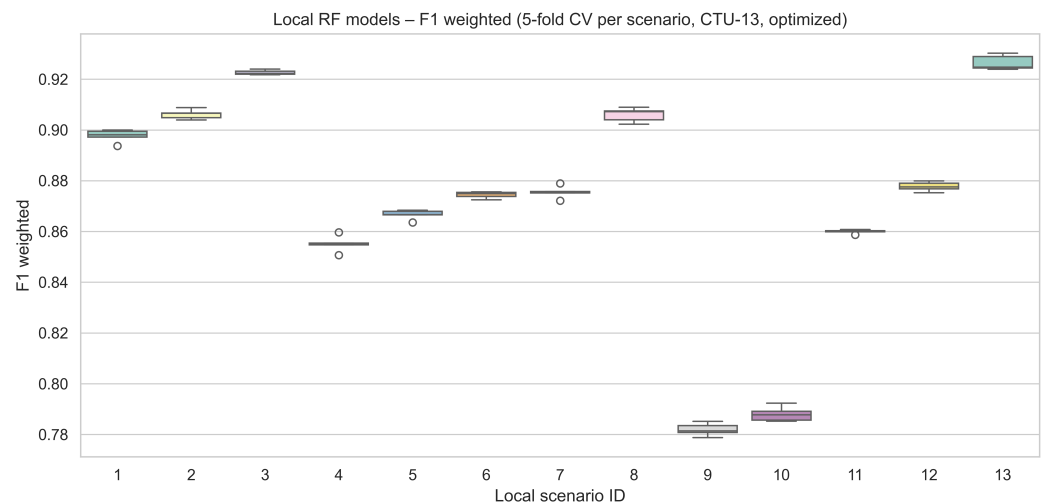


Figure 6. The F1 score for local models in a hierarchical RF model.

RF experiments on the CTU-13 dataset show significantly greater differences between approaches than on the DDoS dataset; the hierarchical model based on local models clearly outperforms the centralized model. This suggests that in distributed environments, such as CTU-13, the hierarchical approach allows for better utilization of the information contained in the data, whilst a single centralized model struggles to adapt to all the patterns present.

5.3.2. Models Based on MLP

This section presents the results of experiments on constructing MLP models; subsequent sections will present and discuss the results of hyperparameter tuning and the model-building process itself, covering both the global approach—based on local models—and the centralized approach.

Tuning the Hyperparameters of MLP Models

For MLP-type models, the situation is qualitatively similar, though absolute F1 scores are lower due to the greater complexity of the CTU-13 dataset. Table 14 contains the results of hyperparameter tuning for the centralized model, and Table 15 for the global model based on local models.

According to Table 14, the best configuration achieved an average F1 score of 0.653. As in previous experiments, there are several other network configurations whose F1 scores are close to that of the best configuration. Based on the results, it can also be observed that the centralized MP model for the CTU-13 dataset is more sensitive to hyperparameter selection than a random forest, and that achieving high quality requires extensive tuning.

Table 15 presents the results of hyperparameter tuning for the global MLP, i.e., the network trained on the prediction vectors of local MLP models. The best configuration (in bold) achieved an average F1 score of 0.836. As in previous cases, the other configurations tested also achieved F1 scores close to the best result, suggesting that the global MLP is more stable with respect to hyperparameters, in contrast to the centralized MLP, where the fluctuations are significantly greater.

Comparing Tables 14 and 15, it can be seen that the difference in quality between the best centralized MLP (0.653) and the best global MLP (0.836) is very significant. This shows that for a dataset such as CTU-13, where the data is highly heterogeneous and varies across scenarios, a representation based on local model predictions is a much better solution for a

neural network than direct training on the entire dataset. Secondly, as with RF, several good configurations for the global model can be observed—different hyperparameter values yield similar results. Thirdly, the fact that the global MLP achieves such high F1 scores despite operating on probability vectors confirms that the proposed approach based on local models can yield better results than the classic centralized approach.

Table 14. Results of hyperparameter tuning for the MLP method on the CTU13 dataset—centralised model.

Hidden Layer Sizes	Alpha	Learning Rate Init	Batch Size	Learning Rate	Mean F1 ± Sd
(128; 64)	1.00×10^{-5}	0.001	64	adaptive	0.653 ± 0.003
(128; 64)	0.001	0.001	256	constant	0.647 ± 0.014
(64; 32)	0.0001	0.001	64	adaptive	0.646 ± 0.010
(128; 64)	1.00×10^{-6}	0.0005	64	constant	0.646 ± 0.008
(128; 64)	1.00×10^{-6}	0.001	128	constant	0.640 ± 0.004
(128; 64)	0.001	0.0005	128	constant	0.640 ± 0.006
(64; 32)	1.00×10^{-5}	0.001	128	constant	0.636 ± 0.009
(64; 32)	1.00×10^{-5}	0.0005	128	constant	0.635 ± 0.007
(64; 32)	1.00×10^{-6}	0.0005	256	adaptive	0.631 ± 0.009
(128;)	0.0001	0.001	128	constant	0.607 ± 0.011
(128; 64)	0.001	0.0001	64	adaptive	0.607 ± 0.027
(32;)	0.001	0.001	128	constant	0.602 ± 0.013
(128; 64)	1.00×10^{-5}	0.0001	128	adaptive	0.601 ± 0.014
(128; 64)	1.00×10^{-6}	0.0001	128	adaptive	0.601 ± 0.012
(64;)	0.0001	0.001	256	constant	0.600 ± 0.010
(128; 64)	0.001	0.0001	128	constant	0.596 ± 0.015
(32;)	0.001	0.001	64	constant	0.594 ± 0.010
(64;)	0.0001	0.0005	128	adaptive	0.590 ± 0.007
(128;)	1.00×10^{-6}	0.0005	256	constant	0.590 ± 0.009
(128; 64)	0.001	0.0001	256	constant	0.584 ± 0.010
(32;)	1.00×10^{-5}	0.0005	128	adaptive	0.577 ± 0.002
(128;)	1.00×10^{-6}	0.0001	64	adaptive	0.568 ± 0.002
(128;)	0.001	0.0001	64	adaptive	0.562 ± 0.003
(128;)	1.00×10^{-6}	0.0001	128	constant	0.547 ± 0.004
(128;)	0.0001	0.0001	128	constant	0.546 ± 0.004
(64;)	1.00×10^{-5}	0.0001	128	constant	0.535 ± 0.004
(64;)	0.001	0.0001	256	adaptive	0.497 ± 0.010
(32;)	1.00×10^{-6}	0.0001	128	adaptive	0.495 ± 0.009
(32;)	0.0001	0.0001	128	constant	0.494 ± 0.009
(32;)	0.001	0.0001	128	constant	0.491 ± 0.008

Main MLP Experiments on CTU-13 Dataset

The final experiment assessed the quality of predictions from the hierarchical and centralized MLP models on the CTU dataset.

Figure 7 shows a plot of the distribution of F1 scores across the individual cross-validation folds.

Based on Figure 7, the superiority of the hierarchical model over the centralised model can be confirmed. The proposed hierarchical model not only achieved higher F1 scores in individual folds, but also had a lower standard deviation.

Table 16 shows the quality assessment metrics for the hierarchical and centralised models.

From Table 16, it can be seen that the differences between the two approaches tested are particularly pronounced for the MLP on the CTU-13 dataset. The results indicate moderate classification performance, significantly lower than that of the RF models on the

same dataset. Against this background, the results of the hierarchical model are clearly superior to those of the centralized model. The hierarchical model based on local models significantly improves detection quality on the CTU-13 dataset.

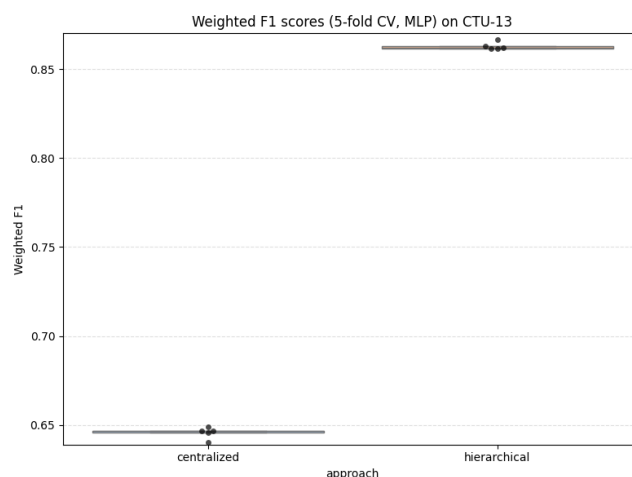


Figure 7. The average weighted F1 score for the centralised and hierarchical models based on local models using MLP model.

Table 15. Results of hyperparameter tuning for the MLP method on the CTU13 dataset—a global model based on local models.

Hidden Layer Sizes	Alpha	Learning Rate Init	Batch Size	Learning Rate	Mean F1 $\pm$ Sd
(128; 64)	0.001	0.0005	128	constant	0.836 $\pm$ 0.001
(128; 64)	1.00×10^{-6}	0.0005	64	constant	0.835 $\pm$ 0.003
(128; 64)	0.001	0.001	256	constant	0.835 $\pm$ 0.001
(128; 64)	1.00×10^{-6}	0.001	128	constant	0.834 $\pm$ 0.002
(64; 32)	1.00×10^{-5}	0.001	128	constant	0.834 $\pm$ 0.001
(128; 64)	0.001	0.0001	64	adaptive	0.833 $\pm$ 0.003
(128; 64)	1.00×10^{-5}	0.001	64	adaptive	0.833 $\pm$ 0.001
(64; 32)	0.0001	0.001	64	adaptive	0.831 $\pm$ 0.004
(64; 32)	1.00×10^{-5}	0.0005	128	constant	0.829 $\pm$ 0.002
(128; 64)	1.00×10^{-5}	0.0001	128	adaptive	0.828 $\pm$ 0.002
(64; 32)	1.00×10^{-6}	0.0005	256	adaptive	0.827 $\pm$ 0.002
(128; 64)	0.001	0.0001	256	constant	0.827 $\pm$ 0.003
(128; 64)	0.001	0.0001	128	constant	0.826 $\pm$ 0.001
(128; 64)	1.00×10^{-6}	0.0001	128	adaptive	0.825 $\pm$ 0.003
(64;)	0.0001	0.001	256	constant	0.824 $\pm$ 0.001
(64;)	0.0001	0.0005	128	adaptive	0.824 $\pm$ 0.003
(32;)	0.001	0.001	128	constant	0.822 $\pm$ 0.002
(128;)	1.00×10^{-6}	0.0005	256	constant	0.822 $\pm$ 0.003
(128;)	0.0001	0.001	128	constant	0.822 $\pm$ 0.001
(32;)	0.001	0.001	64	constant	0.821 $\pm$ 0.002
(32;)	1.00×10^{-5}	0.0005	128	adaptive	0.820 $\pm$ 0.002
(128;)	1.00×10^{-6}	0.0001	64	adaptive	0.774 $\pm$ 0.067
(128;)	0.0001	0.0001	128	constant	0.734 $\pm$ 0.063
(128;)	1.00×10^{-6}	0.0001	128	constant	0.733 $\pm$ 0.064
(128;)	0.001	0.0001	64	adaptive	0.728 $\pm$ 0.064
(64;)	1.00×10^{-5}	0.0001	128	constant	0.693 $\pm$ 0.004
(64;)	0.001	0.0001	256	adaptive	0.691 $\pm$ 0.003
(32;)	1.00×10^{-6}	0.0001	128	adaptive	0.685 $\pm$ 0.003
(32;)	0.001	0.0001	128	constant	0.684 $\pm$ 0.004
(32;)	0.0001	0.0001	128	constant	0.684 $\pm$ 0.004

Table 16. Quality assessment metrics for models (a centralised model and hierarchical model based on local models) for the CTU-13 dataset and the MLP model.

Approach	Accuracy $\pm$ Sd	Precision Weighted $\pm$ Sd	Recall Weighted $\pm$ Sd	F1 Weighted $\pm$ Sd
centralized	0.633 $\pm$ 0.006	0.610 $\pm$ 0.003	0.639 $\pm$ 0.003	0.648 $\pm$ 0.002
hierarchical	0.807 $\pm$ 0.004	0.717 $\pm$ 0.002	0.745 $\pm$ 0.001	0.861 $\pm$ 0.001

Table 17 shows the values of two measures of agreement: Cohen’s kappa coefficient and Matthews’ correlation coefficient (MCC).

Table 17. Cohen’s kappa and Mathews coefficients (centralised and hierarchical, based on local models) for the CTU-13 dataset and the MLP model.

Approach	Cohen Kappa $\pm$ Sd	Matthews Corrcoeff $\pm$ Sd
centralized	0.633 $\pm$ 0.011	0.673 $\pm$ 0.010
hierarchical	0.736 $\pm$ 0.035	0.742 $\pm$ 0.027

For the centralised MLP, the coefficient values indicate that the model reproduces the class distribution to a moderate extent, whilst the number of errors between scenarios is significant. For the hierarchical model, the coefficient values are higher, indicating a significant improvement in the alignment of predictions with true labels, accounting for both class distribution and random alignment. In contrast to the results for the DDoS dataset, here the increases in kappa and MCC are comparable in scale to the increase in weighted F1, suggesting that the improvement applies across the entire error matrix rather than just one class type.

The large difference between the centralized and hierarchical models in CTU-13 is well justified by the nature of this dataset. The individual CTU-13 scenarios represent different botnets, different types of background traffic and different class proportions. A single, common MLP model trained on the combined dataset must ‘reconcile’ conflicting signals from multiple scenarios simultaneously, which proves difficult. A centralised MLP network tends to adapt to dominant patterns and loses the differences specific to individual locations. In a hierarchical approach, each local MLP learns separately for a given scenario, allowing it to capture its specific characteristics more effectively.

5.4. Summary of the Experiments

Although both CIC-DDoS2019 and CTU-13 have been divided into parts to approximate logically distributed nodes, they are nevertheless experimental surrogate models, rather than measurements derived from independently managed, physically separate networks. Consequently, our conclusions regarding distributed network environments should be interpreted as evidence that the proposed architecture performs well in heterogeneous, distributed environments.

To summarize the experimental evaluation conducted across the CIC-DDoS2019 and CTU-13 datasets, several critical conclusions emerge regarding the efficacy of the proposed hierarchical architecture. Across all tested scenarios—encompassing both Random Forest (RF) and Multilayer Perceptron (MLP) base algorithms—the two-level global model consistently outperformed or strictly matched its classical centralized counterpart.

The performance advantage of the hierarchical approach was particularly pronounced in the most complex and heterogeneous scenarios. For instance, on the highly diverse CTU-13 dataset, the global MLP meta-classifier achieved a weighted F1 score of 0.836, compared to merely 0.653 for the centralized MLP. Similarly, the hierarchical RF model on CTU-13 improved the F1 score from 0.866 to 0.910. This substantial leap in classification quality

demonstrates that direct training on raw, highly variable network features from disparate sources is often suboptimal. By transforming raw data into a meta-feature space composed of probability distributions (prediction vectors) generated by specialized local experts, the global model successfully captures complex inter-relational patterns that remain obscured in a centralized raw dataset.

Furthermore, the hyperparameter optimization phase revealed that the global meta-classifiers exhibit remarkable stability. Unlike the centralized models (particularly the centralized MLP on CTU-13), which showed high sensitivity to parameter changes, the hierarchical models achieved optimal or near-optimal results across a broad spectrum of configurations. This indicates that the task of aggregating local “opinions” is structurally more robust and less reliant on exhaustive hyperparameter fine-tuning.

In conclusion, the experimental outcomes unequivocally validate the core hypothesis of this study. The proposed hierarchical architecture not only circumvents the need for raw data aggregation, satisfying strict privacy and operational constraints inherent to dispersed network nodes, but it also actively exploits the localized specialization of these nodes. Ultimately, fusing localized expertise via a global meta-learner provides a more accurate, stable, and robust cyberattack detection mechanism than traditional centralized learning methodologies.

In our research, we have limited ourselves to highly centralized RF and MLP models as upper reference bounds. We do not consider other distributed methods, such as FedAvg-style federated learning, simple ensemble averaging, soft voting, classical stacking, or purely local, personalized schemes. At this stage of our research, we have confirmed the hypothesis that the proposed hierarchical meta-learning architecture can match or outperform well-tuned centralized models without aggregating raw data; a broader, direct comparison with standard distributed and federated baseline models is a direction for future research.

Please note that the article does not include a full ablation study of the proposed architecture. In particular, we do not carry out a systematic comparison of the global meta-classifier with simpler aggregation rules (e.g., soft voting or averaging of local prediction vectors), nor do we investigate the impact of removing the ‘out-of-fold’ protocol or changing the number and quality of local experts. At this stage, the experiments primarily demonstrate that the complete two-level architecture can match or outperform centralised models.

6. Conclusions

In this paper, we introduced a novel two-level hierarchical classification architecture dedicated explicitly to cybersecurity applications, specifically targeting the detection of cyberattacks and anomalies in dispersed network environments. A core motivation for this research is the critical importance of training models locally, as individual network nodes inherently exhibit diverse data characteristics and varying levels of data quality. Particularly in the cybersecurity domain, the quality of collected data—and consequently the efficacy of the local model—is heavily dependent on the specific monitoring device, the types of attacks encountered, and the available computational resources. When attempting to centralize such heterogeneous data, all raw instances are treated equally. This conventional aggregation often leads to reduced model stability and robustness, as low-quality or noisy data from constrained edge devices can dilute and degrade the valuable patterns extracted from high-quality sources.

As demonstrated by our experimental results, the proposed hierarchical model successfully mitigates these issues. By training a global meta-classifier exclusively on the abstract prediction vectors generated by the first-level models, the architecture effectively

captures and weighs the differences in local data quality. The global model proved to be highly robust against poorer quality local data and localized variations, yielding highly accurate and stable final predictions. Furthermore, this approach inherently protects the privacy of sensitive network logs by eliminating the need for raw data transmission.

6.1. Limitation

At the current stage of the research, one limitation is that we are using a synthetic node partition within individual datasets; furthermore, each node has a homogeneous set of features. Consequently, further research will require empirical validation on physically independent sensors with truly heterogeneous feature spaces. However, the proposed two-level approach, by design, separates the global decision-making stage from the representation of raw features: the global meta-learner utilizes only calibrated probability vectors derived from local models. In this approach, differences in local feature sets influence the global model solely through the quality and calibration of these output vectors, rather than through direct incompatibility of input dimensions. It is possible to mitigate this heterogeneity by employing simple mechanisms, such as node-level calibration or masking missing experts. In future research, we will discuss these mechanisms and outline a specific validation plan using independently collected traffic data from various nodes.

The CTU-13 dataset used in our experiments captures various botnet behaviors, but it does not fully reflect the diversity of real-world traffic across different locations. Consequently, the CTU-13 dataset has been found useful for research on attack patterns and background noise, but its scope is narrower than that of independent datasets sourced from various nodes. Research using datasets covering different locations is the next step in this line of investigation.

6.2. Future Work

The next step in this research is to evaluate the proposed meta-learning architecture on genuinely distributed data collected from multiple independently managed network locations to confirm that the approach will also deliver benefits in real-world environments.

In future research, we plan to extend this framework by implementing coalitions of compatible local models specifically for cybersecurity applications. Additionally, we aim to thoroughly investigate the robustness and scalability of the proposed architecture with respect to an increasing number of local models and varying degrees of data dispersion.

We also plan to compare the proposed two-level architecture with classical sub-network learning algorithms, simple ‘soft-voting’ and averaging of local prediction vectors, conventional model aggregation, and purely local or personalized models, in order to verify the benefits and potential limitations of non-iterative meta-learning compared with methods based on parameter synchronization. We will also investigate how the number and reliability of local models affect overall performance.

Author Contributions: Conceptualization, M.P.-K. and K.C.; methodology, M.P.-K. and K.C.; software, K.C.; validation, M.P.-K. and K.C.; formal analysis, M.P.-K. and K.C.; investigation, M.P.-K. and K.C.; resources, M.P.-K. and K.C.; data curation, M.P.-K. and K.C.; writing—original draft preparation, M.P.-K. and K.C.; writing—review and editing, M.P.-K. and K.C.; visualization, M.P.-K. and K.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. The CIC-DDoS2019 dataset can be found in [32], and the CTU-13 dataset can be found in [33].

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Adedeji, K.B.; Abu-Mahfouz, A.M.; Kurien, A.M. DDoS attack and detection methods in internet-enabled networks: Concept, research perspectives, and challenges. *J. Sens. Actuator Netw.* **2023**, *12*, 51. [\[CrossRef\]](#)
- Milosevic, M.S.; Ciric, V.M. Extreme minority class detection in imbalanced data for network intrusion. *Comput. Secur.* **2022**, *123*, 102940. [\[CrossRef\]](#)
- Jiang, M.; Cui, M.; Hou, C.; Cai, W.; Li, Z.; Xiong, G.; Gou, G. FA-Net: More accurate encrypted network traffic classification based on burst with self-attention. In *International Joint Conference on Neural Networks (IJCNN)*; IEEE: New York, NY, USA, 2023; pp. 1–10.
- AlHayan, A.; Al-Muhtadi, J. Federated learning-powered real-time behavioral intrusion detection leveraging LSTM, attention, GANs, and large language models. *Sci. Rep.* **2026**, *16*, 10172. [\[CrossRef\]](#) [\[PubMed\]](#)
- Czarnowski, I. Weighted ensemble with one-class classification and over-sampling and instance selection (WECOI): An approach for learning from imbalanced data streams. *J. Comput. Sci.* **2022**, *61*, 101614. [\[CrossRef\]](#)
- Pekala, B.; Szkoła, J.; Grochowalski, P.; Gil, D.; Kosior, D.; Dyczkowski, K. A novel method for human fall detection using federated learning and interval-valued fuzzy inference systems. *J. Artif. Intell. Soft Comput. Res.* **2025**, *15*, 77–90.
- Szkoła, J.; Pekala, B.; Dyczkowski, K. Refining uncertainty management in machine learning: An interval-valued fuzzy set approach to logistic regression. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*; Springer Nature: Cham, Switzerland, 2024; pp. 233–246.
- Yuan, L.; Wang, Z.; Sun, L.; Yu, P.S.; Brinton, C.G. Decentralized federated learning: A survey and perspective. *IEEE Internet Things J.* **2024**, *11*, 34617–34638. [\[CrossRef\]](#)
- Przybyła-Kasperek, M.; Addo, B.A. Novel hierarchical decision tree frameworks introducing tree method bagging-stump integration and height optimization. In *International Conference on Computational Science*; Springer Nature: Cham, Switzerland, 2025; pp. 3–11.
- Przybyła-Kasperek, M.; Addo, B.A.; Kuształ, K.M. Dual-level decision tree-based model for dispersed data classification. In *Harnessing Opportunities: Reshaping ISD in the Post-COVID-19 and Generative AI Era*; Marcinkowski, B., Przybyłek, A., Jarzębowicz, A., Iivari, N., Insfran, E., Lang, M., Linger, H., Schneider, C., Eds.; University of Gdańsk: Gdańsk, Poland, 2024. [\[CrossRef\]](#)
- Dunning, B.B.; Switlik, J. A real-time expert system for computer network monitor and control. *SIGMIS Database* **1988**, *19*, 35–38. [\[CrossRef\]](#)
- Siddhartha, K.H. Flow-Based Botnet Detection Using Ensemble Machine Learning on the CTU-13 Dataset. Available online: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5881823 (accessed on 7 December 2025).
- Alduailij, M.; Khan, Q.W.; Tahir, M.; Sardaraz, M.; Alduailij, M.; Malik, F. Machine-Learning-Based DDoS Attack Detection Using Mutual Information and Random Forest Feature Importance Method. *Symmetry* **2022**, *14*, 1095. [\[CrossRef\]](#)
- Al-Tarawneh, M.A.; Alir, O.; Kanj, H. Performance Evaluation of Machine Learning-Based Cyber Attack Detection in Electric Vehicles Charging Stations. *Int. J. Adv. Comput. Sci. Appl.* **2025**, *16*, 901–914. [\[CrossRef\]](#)
- Ghazal, S.F.; Mjlae, S.A. Cybersecurity in deep learning techniques: Detecting network attacks. *Int. J. Adv. Comput. Sci. Appl.* **2022**, *13*, 221–230. [\[CrossRef\]](#)
- Becerra-Suarez, F.L.; Tuesta-Monteza, V.A.; Mejia-Cabrera, H.I.; Arcila-Diaz, J. Performance evaluation of deep learning models for classifying cybersecurity attacks in IoT networks. *Informatics* **2024**, *11*, 32. [\[CrossRef\]](#)
- Inuwa, M.M.; Das, R. A comparative analysis of various machine learning methods for anomaly detection in cyber attacks on IoT networks. *Internet Things* **2024**, *26*, 101162. [\[CrossRef\]](#)
- Achaal, B.; Adda, M.; Berger, M.; Ibrahim, H.; Awde, A. Study of smart grid cyber-security, examining architectures, communication networks, cyber-attacks, countermeasure techniques, and challenges. *Cybersecurity* **2024**, *7*, 10. [\[CrossRef\]](#) [\[PubMed\]](#)
- Onur, F.; Gönen, S.; Barışkan, M.A.; Kubat, C.; Tunay, M.; Yılmaz, E.N. Machine learning-based identification of cybersecurity threats affecting autonomous vehicle systems. *Comput. Ind. Eng.* **2024**, *190*, 110088. [\[CrossRef\]](#)
- Alsuwaiket, M.A. ZeroDay-LLM: A Large Language Model Framework for Zero-Day Threat Detection in Cybersecurity. *Information* **2025**, *16*, 939. [\[CrossRef\]](#)
- Hussain, M.; O’Nils, M.; Lundgren, J.; Mousavirad, S.J. A comprehensive review on deep learning-based data fusion. *IEEE Access* **2024**, *12*, 180093–180124. [\[CrossRef\]](#)
- Chung, W.C.; Lo C.A.; Lin, Y.H.; Chen, Z.H.; Hung, C.L. Decentralized federated learning with Non-IID data: Challenges, trends, and future opportunities. *ACM Comput. Surv.* **2026**, *58*, 1–41. [\[CrossRef\]](#)
- Sun, T.; Li, D.; Wang, B. Decentralized federated averaging. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 4289–4301. [\[CrossRef\]](#)
- Li, M.H.; Luo, M.L.; Wang, M.H. Federated learning on non-independent and identically distributed data. In *International Conference on Machine Learning and Computer Application (ICMLCA)*; ACM: New York, NY, USA, 2023; Volume 12636, p. 154.

25. Tabrizchi, H.; Aghasi, A. *Federated Cyber Intelligence*; SpringerBriefs in Computer Science: Berlin/Heidelberg, Germany, 2025. [\[CrossRef\]](#)
26. Sheikh, M.S.; Peng, Y. A comprehensive review on traffic control modeling for obtaining sustainable objectives in a freeway traffic environment. *J. Adv. Transp.* **2022**, 2022, 1012206. [\[CrossRef\]](#)
27. Khraisat, A.; Alazab, A.; Singh, S.; Jan, T.; Gomez, A., Jr. Survey on federated learning for intrusion detection system: Concept, architectures, aggregation strategies, challenges, and future directions. *ACM Comput. Surv.* **2024**, 57, 1–38. [\[CrossRef\]](#)
28. Buyuktanir, B.; Altinkaya, Ş.; Baydogmus, G.K.; Yildiz, K. Federated learning in intrusion detection: Advancements, applications, and future directions. *Clust. Comput.* **2025**, 28, 473. [\[CrossRef\]](#)
29. Vyas, A.; Lin, P.-C.; Hwang, R.-H.; Tripathi, M. Privacy-preserving federated learning for intrusion detection in IoT environments: A survey. *IEEE Access* **2024**, 12, 127018–127050. [\[CrossRef\]](#)
30. Mazid, A.; Kirmani, S.; Manaullah; Yadav, M. FL-IDPP: A Federated Learning Based Intrusion Detection Approach with Privacy Preservation. *Trans. Emerg. Telecommun. Technol.* **2025**, 36, e70039.
31. Ceviz, O.; Sadioglu, P.; Sen, S.; Vassilakis, V.G. A novel federated learning-based IDS for enhancing UAVs privacy and security. *Internet Things* **2025**, 31, 101592. [\[CrossRef\]](#)
32. Sharafaldin, I.; Habibi Lashkari, A.; Hakak, S.; Ghorbani, A.A. Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy. In Proceedings of the IEEE 53rd International Carnahan Conference on Security Technology (ICCST), Chennai, India, 1–3 October 2019.
33. Garcia, S.; Grill, M.; Stiborek, J.; Zunino, A. An empirical comparison of botnet detection methods. *Comput. Secur.* **2014**, 45, 100–123. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.