electronics

MDPI

Article

Federated Auto-Meta-Ensemble Learning Framework for AI-Enabled Military Operations

Konstantinos Demertzis [1,*], Panayotis Kikiras [2], Charalabos Skianis [3], Konstantinos Rantos [4], Lazaros Iliadis [5] and George Stamoulis [2]

[1] School of Science & Technology, Informatics Studies, Hellenic Open University, 26335 Patra, Greece
[2] Department of Electrical and Computer Engineering, University of Thessaly, 38446 Volos, Greece
[3] School of Engineering, Department of Information and Communication Systems Engineering, University of Aegean, 83200 Samos, Greece
[4] School of Science, Department of Computer Science, International Hellenic University, 65404 Kavala, Greece
[5] Department of Civil Engineering, School of Engineering, Democritus University of Thrace, 67100 Xanthi, Greece
* Correspondence: demertzis.konstantinos@ac.eap.gr

Abstract: One of the promises of AI in the military domain that seems to guarantee its adoption is its broad applicability. In a military context, the potential for AI is present in all operational domains (i.e., land, sea, air, space, and cyber-space) and all levels of warfare (i.e., political, strategic, operational, and tactical). However, despite the potential, the convergence between needs and AI technological advances is still not optimal, especially in supervised machine learning for military applications. Training supervised machine learning models requires a large amount of up-to-date data, often unavailable or difficult to produce by one organization. An excellent way to tackle this challenge is federated learning by designing a data pipeline collaboratively. This mechanism is based on implementing a single universal model for all users, trained using decentralized data. Furthermore, this federated model ensures the privacy and protection of sensitive information managed by each entity. However, this process raises severe objections to the effectiveness and generalizability of the universal federated model. Usually, each machine learning algorithm shows sensitivity in managing the available data and revealing the complex relationships that characterize them, so the forecast has some severe biases. This paper proposes a holistic federated learning approach to address the above problem. It is a Federated Auto-Meta-Ensemble Learning (FAMEL) framework. FAMEL, for each user of the federation, automatically creates the most appropriate algorithm with the optimal hyperparameters that apply to the available data in its possession. The optimal model of each federal user is used to create an ensemble learning model. Hence, each user has an up-to-date, highly accurate model without exposing personal data in the federation. As it turns out experimentally, this ensemble model offers better predictability and stability. Its overall behavior smoothens noise while reducing the risk of a wrong choice resulting from under-sampling.

Keywords: federated learning; model-agnostic; meta-learning; ensemble learning; military operations; cyber defense

Citation: Demertzis, K.; Kikiras, P.; Skianis, C.; Rantos, K.; Iliadis, L.; Stamoulis, G. Federated Auto-Meta-Ensemble Learning Framework for AI-Enabled Military Operations. *Electronics* **2023**, *12*, 430. https://doi.org/10.3390/electronics12020430

Academic Editors: Konstantinos Limniotis, Nicholas Kolokotronis and Stavros Shiaeles

Received: 30 November 2022
Revised: 11 January 2023
Accepted: 12 January 2023
Published: 13 January 2023

1. Introduction

With an increasing pace, artificial intelligence (AI) is becoming a significant and integral part of modern warfare because it offers new opportunities for the complete automation of large-scale infrastructure and the optimization of numerous defence or cyber-defence systems [1]. One of the promises of AI in the military domain [2] that seems to guarantee its adoption is its broad applicability. In a military context, the potential for AI is present in all operational domains (i.e., land, sea, air, space, and cyber-space) and all levels of warfare (i.e., political, strategic, operational, and tactical) [3]. Still, at the same time, the complexity is growing exponentially as the number of interconnected systems involved in continuous interconnection and uninterrupted information exchange services expands in real-time [4]. From a generalized point of view, it can be said that AI will have a significant impact on the following missions:

1. Too fast missions with reaction times of seconds or less to be executed in high complexity (data, context, type of mission).
2. Missions with operation duration beyond human endurance or implying high operational (personnel) costs over a long period.
3. Missions involving an overwhelming complexity which requires agility and adaptation to evolutions in context and objectives.
4. Missions challenging operational contexts implying severe risks to war fighters.

Applications supporting the missions above that monitor events in real-time are receiving a constant, unlimited stream of observations of interlinked approaches. These data exhibit high variability because their features vary substantially and unexpectedly over time, altering their typical, expected behaviour. The latest data are the most important in the typical case, as ageing is based on their timing.

Military AI-enabled intelligent systems that utilize data can transform military commanders' and operators' knowledge and experience into optimal valid and timely decisions [3,4]. However, the lack of detailed knowledge and expertise associated with using complex machine learning architectures can affect the performance of the intelligent model, prevent the periodic adjustment of some critical hyperparameters and ultimately reduce the algorithm's reliability and the generalization that should characterize these systems. These disadvantages are preventing stakeholders of defence, at all echelons of the command chain, from trusting and making effective and systematic use of machine learning systems. In this context and given the inability of traditional decision-making systems to adapt to the changing environment, the adoption of intelligent solutions is imperative.

Furthermore, a general difficulty that reinforces distrust of machine learning systems in defence is the prospect of adopting a single data warehouse for the overall training of intelligent models [1], which could create severe technical challenges and severe issues of privacy [5], logic, and physical security due to the need of establishing a potential single point of failure and a potential strategic/primary target for the adversaries [6]. Accordingly, the exchange of data that could make more complete intelligent categorizers that would generalize also poses risks to the security and privacy of sensitive data, which military commanders and operators do not want to risk [7].

To overcome the above double challenge, this work proposes FAMEL. It is a holistic system that automates selecting and using the most appropriate algorithmic hyperparameters that optimally solve a problem under consideration, approaching it as a model for finding algorithmic solutions where it is solved by mapping between input and output data. The proposed framework uses meta-learning to identify similar knowledge accumulated in the past to speed up the process [8]. This knowledge is combined using heuristic techniques, implementing a single, constantly updated intelligent framework. Data remains in the local environment of the operators, and only the parameters of the models are exchanged through secure processes, thus making it harder for potential adversaries to intervene with the system [9,10].

2. Proposed Framework

In the proposed FAMEL framework, each user uses an automatic meta-learning system in a horizontal federated learning approach (horizontal federated learning uses datasets with the same feature space across all devices. Vertical federated learning uses different datasets of different feature space to jointly train a global model). The most appropriate algorithm with the optimal hyperparameters is selected in a fully automated way, which can optimally solve the given problem. The implementation is based on the entity's available data and is not required to be disposed of in a remote repository or shared with a third party [11].

The whole process is described in Figure 1.

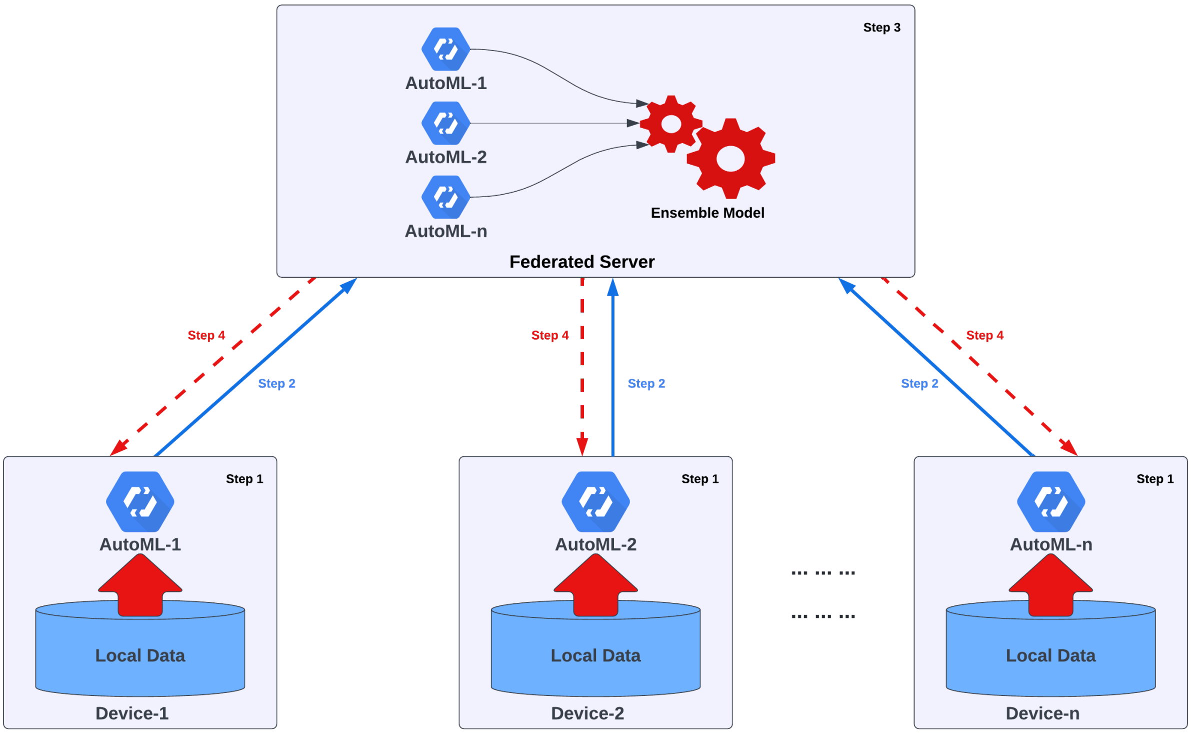

Figure 1. The proposed block diagram of the FAMEL framework.

Specifically:

5. Step 1—Fine-tune the best local model. The fine-tuning process will help to improve the accuracy of each machine learning model by integrating data from an existing dataset and using it as an initialization point to make the training process time- and resource-efficient.
6. Step 2—Upload the local model to the federated server.
7. Step 3—Ensemble the model by the federated server. This ensemble method uses multiple learning algorithms to obtain a better predictive performance than could be obtained from any of the constituent learning algorithms alone.
8. Step 4—Dispatch the ensemble model to local devices.

The best models (winner algorithm) that result from the process are channelled to a federated server, where an ensemble learning model through a heuristic mechanism is created. This ensemble model essentially incorporates the knowledge represented by the local best models, which, as mentioned, came from the local data held by the users [12]. Hence, collectively, the ensemble model offers high generalization, better predictability, and stability. Its general behaviour smoothens noise while lowering the overall danger of making a false choice due to modelling or prejudice in handling scenarios of local data [13,14].

2.1. Federated Learning

Assuming that $F_i = 1,2,\ldots,N$ data owners want to train a machine learning model using their data $\mathrm{D} = \{D_i, i = 1,2,\ldots,N\}$. A traditional way would be to collect all data into a single set $D_{sum} = D_1 \cup D_1 \cup \cdots \cup D_N$ to train a model M_{sum}. The proposed federated learning system creates a single universal model [15]:

$$M_{fed} = \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^{k}$$

where K is the total number of nodes used in the process, n is the data points, and t is the number of federated learning rounds [16]. The model comes from local models $\Delta w^1, \Delta w^2, \ldots, \Delta w^K$, which are trained from the D_i of each federal user separately. Data D1 of the user F1 is not exposed to other federal users. In addition, the accuracy V_{sum} and $M_{sum}$of the models V_{fed} and M_{fed}, must be close or equal. Specifically, if δ is a negative number, then the federal learning method suffers from a loss of accuracy, as indicated in the following formula [11,13]:

$$\left|V_{fed} - V_{sum}\right| < \delta$$

The Auto-Machine Learning technique is used to develop an accurate and robust federal system that will remain stable in new information without the ability to generalize or suffer a considerable loss of δ-accuracy [17,18].

2.2. Auto-Machine Learning

Initially, each federation member has a set of D data containing attribute vectors and class tags on a supervised problem linked with a job. Data set D is specifically divided into two parts: a set of training S and a set of forecasts B for testing and testing so that $D = \langle S, B \rangle$. Furthermore, the data set D contains vector-label pairings such that $D = \{(x_i, y_i)\}$. Each label represents a known class and belongs to a known set of L labels [19].

Considering $P(D)$ the distribution of aggregate data held by federal agencies, we can sample the issuance of an individual data set such that $P_d = P_d(\mathbf{x}, y)$. Our problem lies in creating a trained classifier $\mathcal{M}_\lambda : \mathbf{x} \mapsto y$ which is fully and optimally configured with $\lambda \in \Lambda$ so that it can automatically generate predictions for samples from the P_d distribution minimizing the expected generalization error so that [20]:

$$GE(\mathcal{M}_\lambda) = \mathbb{E}_{(\mathbf{x},y)\sim P_d}[\mathcal{L}(\mathcal{M}_\lambda(\mathbf{x}), y)]$$

The first phase is the best model selection procedure, which appears to be a standard learning procedure in which a data set is regarded as a sample of data. Furthermore, given that each data set of each independent body can only be observed through a set of n independent observations, i.e.,:

$$D_d = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\} \sim P_d$$

Implies that we can only empirically approach the generalization error in data samples, i.e., [20,21]:

$$\widehat{GE}(\mathcal{M}_\lambda, \mathcal{D}_d) = \tfrac{1}{n}\textstyle\sum_{i=1}^{n} \mathcal{L}(\mathcal{M}_\lambda(\mathbf{x}_i), y_i)$$

From the above, we conclude that we have access to unconnected, finite samples in practice where D_{train} and D_{test} ($D_{d,train}, D_{d,test} \in P_d$). Therefore, to search for the best machine learning algorithm, we only have access to D_{train}. However, in the end, the performance is calculated once in D_{test}.

Assume a classifierf_λ, the parameter λ obtains the likelihood that a data point belongs to the class y specified by the attribute vector $x, P_\lambda\ (y|x)$. The best model should increase the likelihood of correctly detecting tags over several training batches $B \subset D$ so that [18,22]:

$$\lambda^* = argmax_\lambda \mathbb{E}_{B \subset D}\left[\sum_{(x,y)\in B} P_\lambda(y|x)\right]$$

Given that there is only a limited collection of quick learning support that can act as fine-tuning, the objective, as with any other work using machine learning, is to minimize the prediction error made on data samples with unknown labels. It is possible that obtaining the best model is challenging to undertake. A fake data set is created with only a tiny fraction of labels to prevent releasing all labels in the model. The optimization technique is modified to make it easier to acquire knowledge quickly. According to this interpretation, each sample pair can be regarded as a data point. As a direct consequence, the model has been educated to the point where it can generalize to fresh, untested data sets. To summarize, the process of computing the best model through the application of the meta-learning approach is represented by the following function [20]:

$$\lambda^{*} = argmax_{\lambda}\mathbb{E}_{L_s \subset L}\left[\mathbb{E}_{S^L \subset D, B^L \subset D}\left[\sum\nolimits_{(x,y)\in B^L} P_{\lambda}(x, y, S^L)\right]\right]$$

Therefore, the proposed framework performs an automatic search in the solutions area to identify the optimal $\mathcal{M}_{\lambda^*}$:

$$\mathcal{M}_{\lambda^*} \in \underset{\lambda\in\Lambda}{\operatorname{argmin}} \widehat{GE}(\mathcal{M}_{\lambda}, \mathcal{D}_{\text{train}})$$

For the calculation of GE, with cross-validation k-fold, the following relation is used [17,20,23]:

$$\widehat{GE}_{\text{CV}}(\mathcal{M}_{\lambda}, \mathcal{D}_{\text{train}}) = \frac{1}{K}\sum\nolimits_{k=1}^{K} \widehat{GE}\left(\mathcal{M}_{\lambda}^{\mathcal{D}_{\text{train}}^{(\text{train},k)}}, \mathcal{D}_{\text{train}}^{(\text{val},k)}\right)$$

where $\mathcal{M}_{\lambda}^{\mathcal{D}_{\text{train}}^{(\text{train},k)}}$ denote that $\mathcal{M}_{\lambda}$ was trained based on the k-fold dataset $\mathcal{D}_{\text{train}}^{(\text{train},k)} \subset \mathcal{D}_{\text{train}}$ and then evaluated by:

$$\mathcal{D}_{\text{train}}^{(\text{val},k)} = \frac{\mathcal{D}_{\text{train}}}{\mathcal{D}_{\text{train}}^{(\text{train},k)}}$$

Accordingly, the problem of optimizing the hyperparameters $\lambda \in \mathit{\Lambda}$ of the best learning algorithm A is essentially similar to selecting the best model. Some significant characteristics are that hyperparameters are frequently continuous, hyperparameter spaces are often vast, and we can benefit from the correlation between different hyperparameter settings $\lambda_1, \lambda_2, \ldots, \lambda_n \in \mathit{\Lambda}$.

Specifically, when n hyperparameters $\lambda_1, \lambda_2, \ldots, \lambda_n \in \mathit{\Lambda}$, the hyperparameter space L includes the subsets$\mathit{\Lambda}_1, \mathit{\Lambda}_2, \ldots, \mathit{\Lambda}_n$. This logic strictly defines each subset, so some hyperparameter settings make other hyperparameters inactive.

Specifically, the hyperparameter λ_i is subject to the sub-constraints of another hyperparameter λ_j, if λ is active only if the hyperparameter λ_j takes values from a given set $V_i(j) \subsetneq \Lambda_j$. Based on this logic, the hyperparameters in the proposed framework create a structured solution space which is determined on the basis of a pair of variables with $B = \langle G, \Theta \rangle$ (where G a graph). Graph G conveys the assumption that each variable λ_i is independent of the inheritance undertaken by G. It determines the parameters of the network and, in particular, the whole $\theta_{\lambda_i|\pi_i} = P_B(\lambda_i|\pi_i)$ for each $\lambda_i \subset \Lambda_i$ based on the constraint condition π_i, for the set of constraints in G. Therefore, B defines a unique probability distribution such that [24]:

$$P_B = (\mathit{\Lambda}_1, \mathit{\Lambda}_2, \ldots, \mathit{\Lambda}_n) = \prod_{i=1}^{n} P_B(\pi_i) = \prod_{i=1}^{n} \theta_{\lambda_i}|\pi_i)$$

Finding the optimal graph path based on the Markov inequality is calculated as [25,26]:

$$\sum_{k=\omega}^{n} \binom{n}{k}(k-1)!\,p^k = \sum_{k=\omega}^{n} \frac{\prod_{i=0}^{k-1}(n-i)\,\lambda^k}{n^k}\frac{}{k} \leq \sum_{k=\omega}^{n} \lambda^k = O(\lambda^{\omega})$$

Hence, with the following equation, the calculation of its expectation is performed by $\mathit{\Lambda}_n$:

$$\mathbb{E}[\Lambda_n] = \sum_{k=3}^{n} \binom{n}{k} (k-1)!\, p^k$$

It follows from the above equation:

$$\lim_{n\to\infty} \mathbb{E}(\Lambda_n) = \lim_{n\to\infty} \sum_{k=3}^{n} \frac{\prod_{i=0}^{k-1} (n-i)}{n^k} \frac{\lambda^k}{k} \sim \sum_{k=3}^{\infty} \frac{\lambda^k}{k} = -\log(1-\lambda) - \lambda - \frac{\lambda^2}{2} = a(\lambda)$$

Hence, its r-to factor momentum Λ_n is:

$$\mathbb{E}[(\Lambda_n)_r] = \hat{\theta} \sum_{k_1=3}^{n} \sum_{k_2=3}^{n-k_1} \cdots \sum_{k_r=3}^{n-\sum_{i=1}^{r-1} k_i} \binom{n}{k_1, k_2, \ldots, k_r, n-k_1-\cdots-k_r} \prod_{i=1}^{r} (k_i-1)!\, p^{k_i}$$

Finally, given the above-structured solution space, the hyperparameter optimization issue is as follows [27,28]:

$$\lambda^* \in \underset{\lambda\in\Lambda}{\operatorname{argmin}} \frac{1}{k} \sum_{i=1}^{k} \mathcal{L}\left(A_\lambda, \mathcal{D}_{\text{train}}^{(i)}, \mathcal{D}_{\text{valid}}^{(i)}\right)$$

The Meta-Ensemble Learning technique is used for the proposed framework to lead to stable prediction models while offering generalization, minimizing bias, reducing variance, and eliminating overfitting.

2.3. Meta-Ensemble Learning

Once the above procedure has identified the most appropriate algorithm with the optimal hyperparameters to create a single model that improves generalization, the proposed framework creates a boosting ensemble model of all the optimal models that emerged by auto-machine learning.

The proposed technique is based on the logic of the boosting process, where through the creation of successive tree structures, information transfer is applied to solve a distributed problem [29]. Specifically, it is set$f(x) = 0$ and $\varepsilon_i = y_i$ for each observation in the set of training data of each body.

The winning algorithm from the process of auto-machine learning $\widehat{f^k}$ is trained in each round k with d nodes having as response variable the categorization errors resulting from the previous classification round (auto-machine learning process), which are denoted by ε_i. For the most efficient, effective, and computable feasible implementation of the proposed framework, we consider a tree and even pruned version of a new tree so that [12,30]:

$$\hat{f}(x) \leftarrow \hat{f}(x) + \lambda\widehat{f^k}(x) \text{ or } \varepsilon_i \leftarrow \varepsilon_i - \lambda\widehat{f^k}(x)$$

Repeating the procedure K times (the user-specified K), the final form of the model is obtained:

$$\hat{f}(x) = \lambda \sum_{k=1}^{K} \widehat{f^k}(x)$$

For the proposed technique to be effective, the user must specify the number and depth of trees to be created. The incredible depth of the trees can easily create over-adaptation processes and cannot be generalized. Accordingly, the number of trees controls the complexity of the process [31]. The λ parameter defines the learning rate of the model.

Its derivative is first calculated to find the total minimum of the function using the proposed technique, and then the inverse procedure of finding the derivative is used. The derivative measures whether the value of a process will change $J(\theta)$ if the variable θ

(slope of the function) changes slightly. High values of the function indicate a significant slope and, therefore, a substantial change in its value $J(\theta)$ for small changes of θ.

This algorithm is iterative, initializes a random value in θ, calculates the derivative of the function at the given point, and changes θ so that [28,32]:

$$\theta = \theta - \rho \frac{dj}{d\theta}$$

Taking as a function of loss the sum of the squares of the incorrect classifications ε_i is divided by two so that: the parameter ρ determines how fast it will move in the negative direction of the derivative. The process is repeated until the algorithm converges, which proposes training trees in the negative derivative of the loss function:

$$L(y_i, \hat{y}_i) = \frac{1}{2}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

Calculating the above derivative, we have:

$$\frac{dL(y_i,\hat{y}_i)}{d\hat{y}_i} = \hat{y}_i - y_i$$

The negative derivative of the loss function is equal to the classification errors ε_i. Hence, essentially, the procedure provides for the training of a tree based on the classification errors ε_i, to which a pruned version of the new tree is added. In this manner, the approach adds successive trees to the negative derivative of the loss function at each given time t, such that [33,34]:

$$\hat{y}_i^{(t)} = \sum_{t=1}^{K} f_t(x_i),\ f_t \in F$$

where $F = \{f(x) = w_{q(x)}\}$ and $q: R^m \to T, w \in R^T$. The q represents the structure of each tree, the T the number of leaves, and each f_t corresponds to an independent tree structure q with the leaf weights plotted as w. The loss function that is minimized at any time t has a formula:

$$L^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{\,t}\right) + \sum_{k=1}^{T} \Omega_{f(t)}$$

Two terms are important: the model's capacity for learning from training data (low values imply good learning) and the complexity of each tree (adding a new term to the number of leaves (T), which shrinks the weights of leaves so that:

$$\Omega_{f(t)} = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$$

The parameter γ indicates the penalty value for the tree's growth so that large values of γ will lead to small trees. Respectively small values of γ will lead to large trees. The parameter λ regulates whether the tree weights will shrink so that as its value increases, the tree weights will shrink.

Thus, it follows that [33,35]:

$$\hat{y}_i^{(t)} = \sum_{t=1}^{K} f_t(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i)$$

Therefore, the problem now is deciding which $f_t(x_i)$ minimizes the time loss function t:

$$L^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t)}\right) + \sum_{k=1}^{T} \Omega_{f(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right) + \sum_{k=1}^{T} \Omega_{f(t)}$$

Taylor’s Development shows:

$$f(x + \Delta x) \cong f(x) + f'(x)\Delta x + \frac{1}{2} f''(x)(\Delta x)^2$$

Hence, the resulting relationship is [33,35]:

$$L^{(t)} \cong \sum_{i=1}^{n}\left[l\left(y_i, \hat{y}_i^{(t-1)}\right)+g_i f_t(x_i)+\frac{1}{2}h_i f_t^2(x_i)\right]+\Omega_{f(t)}$$

where:

$$g_i = d_{\hat{y}_i^{(t-1)}}l\left(y_i, \hat{y}_i^{(t-1)}\right)\ and\ h_i = d^2_{\hat{y}_i^{(t-1)}}l\left(y_i, \hat{y}_i^{(t-1)}\right)$$

Subtracting the constants, the loss function becomes:

$$L'^{(t)} \cong \sum_{i=1}^{n}\left[g_i f_t(x_i)+\frac{1}{2}h_i f_t^2(x_i)\right]+\Omega_{f(t)}$$

where:

$$I_j = \{i|g(x_i) = j\}$$

The set of observations on a leaf j in the above relation is recorded as follows:

$$L'^{(t)} \cong \sum_{i=1}^{n}\left[g_i w_q(x_i)+\frac{1}{2}h_i w_q^2(x_i)\right]+\Omega_{f(t)} = \sum_{i=1}^{T}\left[\left(\sum_{i\in I_j} g_i\right)w_j+\frac{1}{2}\left(\sum_{i\in I_j} h_i+\lambda\right)w_j^2\right]+\gamma T$$

where:

$$G_j = \sum_{i\in I_j} g_i\ and\ H_j = \sum_{i\in I_j} h_i$$

The following relation emerges:

$$L'^{(t)} = \sum_{i=1}^{T}\left[G_j w_j+\frac{1}{2}\left(H_j+\lambda\right)w_j^2\right]+\gamma T$$

If the structure of the tree($q(x)$) is given, the optimal weight on each sheet is obtained by minimizing the concerning w_j in the above relationship so that [22,36,37]:

$$w_j = -\frac{G_j}{H_j+\lambda}$$

Finally, with its replacement w_j, the following equation is obtained, which calculates the quality of the new tree:

$$L'^{(t)} = -\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j+\lambda}+\gamma T$$

Finally, the algorithm creates divisions using the formula:

$$Gain = \frac{1}{2}\left[\frac{G_L^2}{H_L+\lambda}+\frac{G_R^2}{H_R+\lambda}-\frac{(G_L+G_R)^2}{H_L+H_R+\lambda}\right]-\gamma$$

where the first fraction is the score of the left part of the partition, the second is the score of the right amount of the division, the third is the score if the division is not made, and γ measures the cost of the complexity of the partition.

3. Experiments and Results

For the experimental implementation of the proposed FAMEL and the performance of the scenario, a collaborative network of three federated partners (domain_alpha, domain_bravo and domain_charlie) was simulated (Figure 2).

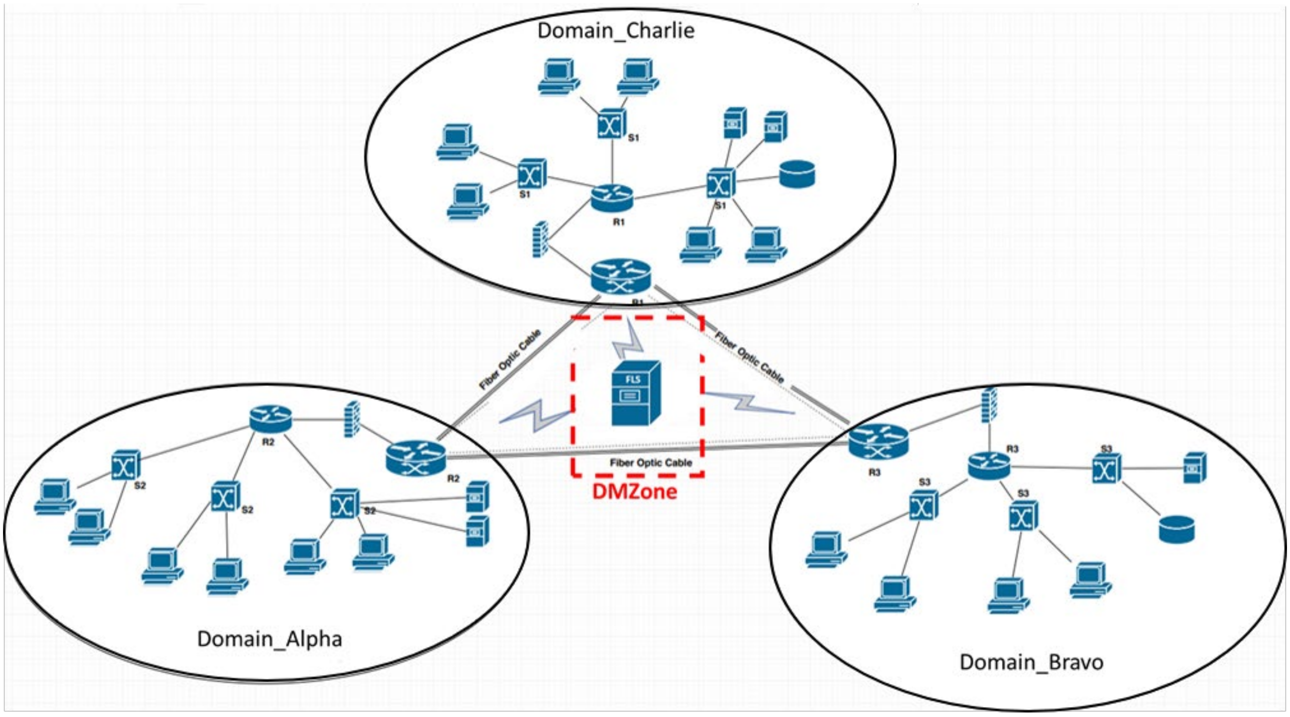

Figure 2. FAMEL architectural modelling.

We consider that the optimal model created by the Auto-Machine Learning process is an internal affair of each domain, which is implemented on a local server based on the respective architecture of each domain. In the Demilitarized Zone (DMZone) is the Federated Learning Server (LFS), which creates the ensemble model by applying the algorithmic process of assembling the optimal models with the technique discussed above. The proposed intelligent system was evaluated using one of the most extensive datasets for web traffic analysis called CICDoS2019. This dataset was developed under the supervision of the Canadian Institute for Cybersecurity. The evaluation's primary objective was to identify well-organized attacks in which the intruder’s identity remained a legal component of a third party [31]. Each domain includes 70 independent variables: characteristics or statistics of network analysis and six classes (Benign, Infiltration, SSH-Bruteforce, FTP-BruteForce, DoS Attack-Hulk, and DDOS attack-HOIC). The individual sets include Alpha_dataset 70553, Bravo_dataset 69551, and Charlie_dataset 70128 instances [38].

The initial results of the Auto-Machine Learning process based on the data available in each domain are presented in Tables 1–9 below, as well as the parameters of each optimal model that emerged for each collaborative domain. We used the Area under the ROC Curve (AUC) metric, which represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes. Specifically, AUC (also known as AUROC) is the Area beneath the entire ROC curve. AUC provides a convenient, single performance metric for our classifiers independent of the specific classification threshold. This enables us to compare models without even looking at their ROC curves.

Table 1. The best model for the Domain Alpha.

Domain Alpha					
Model	**Accuracy**	**AUC**	**Recall**	**Precision**	**F1-Score**
Light Gradient Boosting Machine	0.879	0.926	0.876	0.879	0.879
Gradient Boosting Classifier	0.878	0.926	0.875	0.878	0.878
K Neighbours Classifier	0.876	0.927	0.873	0.876	0.876
Logistic Regression	0.873	0.924	0.869	0.873	0.873
SVM—Linear Kernel	0.870	0.925	0.867	0.870	0.870
Ada Boost Classifier	0.868	0.000	0.865	0.868	0.868
Random Forest Classifier	0.865	0.926	0.862	0.865	0.865
Linear Discriminant Analysis	0.864	0.924	0.861	0.864	0.864
Ridge Classifier	0.860	0.000	0.857	0.860	0.860
Extra Trees Classifier	0.853	0.920	0.852	0.853	0.853
Decision Tree Classifier	0.824	0.883	0.824	0.824	0.824
Naive Bayes	0.747	0.904	0.733	0.770	0.734
Quadratic Discriminant Analysis	0.367	0.900	0.405	0.575	0.321

Table 2. Best parameters of the winner model of the Domain Alpha.

Domain_Alpha	
Best Model	**Best Parameters of the Winner Model**
LGBMClassifier	boosting_type = 'gbdt', class_weight = None, colsample_bytree = 1.0, importance_type = 'split', learning_rate = 0.1, max_depth = -1, min_child_samples = 20, min_child_weight = 0.001, min_split_gain = 0.0, n_estimators = 100, n_jobs = -1, num_leaves = 31, objective = None, random_state = 1599, reg_alpha = 0.0, reg_lambda = 0.0, silent='warn',subsample = 1.0, subsample_for_bin = 200,000, subsample_freq = 0

Table 3. The best model for the Domain Bravo.

Domain_Bravo					
Model	**Accuracy**	**AUC**	**Recall**	**Precision**	**F1-Score**
Gradient Boosting Classifier	0.877	0.926	0.875	0.877	0.877
Light Gradient Boosting Machine	0.876	0.926	0.874	0.876	0.876
K Neighbours Classifier	0.876	0.926	0.873	0.874	0.875
Ada Boost Classifier	0.870	0.925	0.868	0.870	0.870
Random Forest Classifier	0.870	0.923	0.868	0.870	0.870
Linear Discriminant Analysis	0.865	0.923	0.863	0.865	0.865
SVM—Linear Kernel	0.865	0.000	0.863	0.865	0.865
Logistic Regression	0.863	0.925	0.861	0.863	0.862
Ridge Classifier	0.861	0.000	0.859	0.862	0.861
Extra Trees Classifier	0.849	0.920	0.849	0.849	0.849
Decision Tree Classifier	0.816	0.878	0.816	0.816	0.815
Naive Bayes	0.739	0.905	0.727	0.765	0.724
Quadratic Discriminant Analysis	0.594	0.917	0.570	0.572	0.545

Table 4. Best parameters of the winner model of the Domain Bravo.

Domain_Bravo	
Best Model	**Best Parameters of the Winner Model**
GradientBoostingClassifier	ccp_alpha = 0.0, criterion = 'friedman_mse', init = None, learning_rate = 0.1, loss = 'deviance', max_depth = 3, max_features = None, max_leaf_nodes = None, min_impurity_decrease = 0.0, min_impurity_split = None, min_samples_leaf = 1, min_samples_split = 2, min_weight_fraction_leaf = 0.0, n_estimators = 100, n_iter_no_change = None, presort = 'deprecated', random_state = 8515, subsample = 1.0, tol = 0.0001, validation_fraction = 0.1, verbose = 0, warm_start = False

Table 5. The best model for the Domain Charlie.

Domain_Charlie					
Model	**Accuracy**	**AUC**	**Recall**	**Precision**	**F1-Score**
k-Neighbours Classifier	0.866	0.927	0.864	0.867	0.866
Light Gradient Boosting Machine	0.865	0.926	0.864	0.866	0.866
Gradient Boosting Classifier	0.865	0.926	0.865	0.865	0.866
Ada Boost Classifier	0.861	0.921	0.861	0.861	0.861
Logistic Regression	0.860	0.922	0.860	0.861	0.860
SVM—Linear Kernel	0.855	0.923	0.852	0.855	0.855
Random Forest Classifier	0.853	0.925	0.851	0.853	0.853
Linear Discriminant Analysis	0.851	0.923	0.849	0.852	0.851
Extra Trees Classifier	0.847	0.921	0.847	0.848	0.849
Ridge Classifier	0.847	0.920	0.848	0.849	0.848
Decision Tree Classifier	0.819	0.880	0.821	0.820	0.819
Naive Bayes	0.687	0.900	0.668	0.680	0.644
Quadratic Discriminant Analysis	0.542	0.914	0.536	0.662	0.528

Table 6. Best parameters of the winner model of the Domain Charlie.

Domain_Charlie	
Best Model	**Best Parameters of the Winner Model**
KNeighborsClassifier	algorithm = 'auto', leaf_size = 30, metric = 'minkowski', metric_params = None, n_jobs = -1, n_neighbors = 5, *p* = 2, weights = 'uniform'

Table 7. Ensemble model for the Domain Alpha.

Domain_Alpha					
Model	**Accuracy**	**AUC**	**Recall**	**Precision**	**F1-Score**
Ensemble model	0.898	0.933	0.899	0.897	0.898
Light Gradient Boosting Machine	0.879	0.926	0.876	0.879	0.879
Gradient Boosting Classifier	0.878	0.926	0.875	0.878	0.878
k-Neighbors Classifier	0.876	0.927	0.873	0.876	0.876
Logistic Regression	0.873	0.924	0.869	0.873	0.873
SVM—Linear Kernel	0.870	0.925	0.867	0.870	0.870
Ada Boost Classifier	0.868	0.000	0.865	0.868	0.868
Random Forest Classifier	0.865	0.926	0.862	0.865	0.865
Linear Discriminant Analysis	0.864	0.924	0.861	0.864	0.864
Ridge Classifier	0.860	0.000	0.857	0.860	0.860
Extra Trees Classifier	0.853	0.920	0.852	0.853	0.853
Decision Tree Classifier	0.824	0.883	0.824	0.824	0.824
Naive Bayes	0.747	0.904	0.733	0.770	0.734
Quadratic Discriminant Analysis	0.367	0.900	0.405	0.575	0.321

Table 8. Ensemble model for the Domain Bravo.

Domain_Bravo					
Model	**Accuracy**	**AUC**	**Recall**	**Precision**	**F1-Score**
Ensemble model	0.891	0.930	0.890	0.890	0.892
Gradient Boosting Classifier	0.877	0.926	0.875	0.877	0.877
Light Gradient Boosting Machine	0.876	0.926	0.874	0.876	0.876
k-Neighbors Classifier	0.876	0.926	0.873	0.874	0.875
Ada Boost Classifier	0.870	0.925	0.868	0.870	0.870
Random Forest Classifier	0.870	0.923	0.868	0.870	0.870
Linear Discriminant Analysis	0.865	0.923	0.863	0.865	0.865
SVM—Linear Kernel	0.865	0.000	0.863	0.865	0.865
Logistic Regression	0.863	0.925	0.861	0.863	0.862
Ridge Classifier	0.861	0.000	0.859	0.862	0.861
Extra Trees Classifier	0.849	0.920	0.849	0.849	0.849
Decision Tree Classifier	0.816	0.878	0.816	0.816	0.815
Naive Bayes	0.739	0.905	0.727	0.765	0.724
Quadratic Discriminant Analysis	0.594	0.917	0.570	0.572	0.545

Table 9. Ensemble model for the Domain Charlie.

Domain_Charlie					
Model	**Accuracy**	**AUC**	**Recall**	**Precision**	**F1-Score**
Ensemble model	0.871	0.929	0.871	0.871	0.872
k-Neighbors Classifier	0.866	0.927	0.864	0.867	0.866
Light Gradient Boosting Machine	0.865	0.926	0.864	0.866	0.866
Gradient Boosting Classifier	0.865	0.926	0.865	0.865	0.866
Ada Boost Classifier	0.861	0.921	0.861	0.861	0.861
Logistic Regression	0.860	0.922	0.860	0.861	0.860
SVM—Linear Kernel	0.855	0.923	0.852	0.855	0.855
Random Forest Classifier	0.853	0.925	0.851	0.853	0.853
Linear Discriminant Analysis	0.851	0.923	0.849	0.852	0.851
Extra Trees Classifier	0.847	0.921	0.847	0.848	0.849
Ridge Classifier	0.847	0.920	0.848	0.849	0.848
Decision Tree Classifier	0.819	0.880	0.821	0.820	0.819
Naive Bayes	0.687	0.900	0.668	0.680	0.644
Quadratic Discriminant Analysis	0.542	0.914	0.536	0.662	0.528

AUC is measured on a scale of 0 to 1, with higher numbers indicating better performance. Scores in the [0.5, 1] range indicate good performance, while anything less than 0.5 indicates very poor performance. An AUC of 1 indicates a perfect classifier, while an AUC of 0.5 indicates a perfectly random classifier. A model that always predicts a negative sample is more likely than a positive sample to have a positive label. It will have an AUC of 0, indicating a severe modelling failure.

It should be noted that all the tests were performed with 10-fold cross-validation. Each of the ten subsets was used for the algorithm's training and certainly once for its evaluation, so there was no case of misleading the algorithmic result.

Meta-Ensemble Learning is created with an ensemble model that includes the best classifiers. The ensemble model returns through the Federated Learning process in each domain and retests in each local dataset (Alpha_dataset, Bravo_dataset, and Charlie_dataset). Then, the three best models from each domain (LGBMClassifier, Gradient BoostingClassifier, and k-NeighborsClassifier) are sent with the Federated Learning process to FLS. Again, it should be emphasized that all the tests were performed with the method of 10-fold cross-validation so that there was no case of misleading the algorithmic result. The results of the process are presented in the following tables.

The ensemble model ensures an improved categorization accuracy and smoothening of the system. This dramatically simplifies trend detection and visualization by eliminating or reducing statistical noise in the data. The experimental results suggest that using the ensemble model ensures improved categorization accuracy. The categorization becomes more accurate with each instance, providing critical pointers to the failure problems that an individual algorithm's bias could generate [39]. This allows for a precise diagnosis before embarking on a new condition or occurrence associated with adversarial attacks or zero-day exploits. This is one of the most effective strategies for predicting a trend's strength and the likelihood of shifting direction [40].

The convergence achieved by employing multiple models provides more outstanding reliability than any of them could provide separately. This revelation, directly related to the experimental outcomes, significantly accelerates arriving at the optimum decision in ambiguous situations [41]. It is also important to remember that this process is dynamic, which must be emphasized. This dynamic process ensures the system's adaptability by providing impartiality and generalization, resulting in a system that can respond to highly complicated events [42,43].

4. Conclusions

Applying machine learning to real-world problems is still particularly challenging [44]. This is because highly trained engineers and military specialists who have a wealth of experience and information will be required to coordinate the numerous parameters of the respective algorithms, correlate them with the specific problems, and use the data sets that are currently available. This is a lengthy, laborious, and expensive undertaking. However, the hyperparametric features of algorithms and the design choices for ideal parameters can be viewed as optimization problems because machine learning can be thought of as a search problem that attempts to approach an unknown underlying mapping function between input and output data.

Utilizing the above view, in the present work, FAMEL was presented, extending the idea of formulating a general framework of automatic machine learning with effective universal optimization, which operates at the federal level. It uses automated machine learning to find the optimal local model in the data held by each federal user and then, making extensive meta-learning, creates an ensemble model, which, as shown experimentally, can generalize, providing highly reliable results. In this way, the federal bodies have a dedicated, highly generalized model, the training of which does not require exposure to the federation of the data in their possession. In this regard, FAMEL can be applied to several military applications where continuous learning and environmental adaptation are critical for the supported operations and where the exchange of information might be difficult or not possible due to security reasons. For example, which is the case in the real-time optimization of information sharing concerning tasks and situations. The application of FAMEL would be of special interest in congested environments where IoT sensor grids are deployed, and many security constraints need to be met. Similarly, it can be applied in cyberspace operations to find and identify potential hostile activities in cluttered information environments and complex physical scenarios in real-time, including countering negative digital influence [45,46]. It must be noted that the proposed technique can be extended to cover a wider scientific area without reducing the main points that are currently described. It is a universal technic that develops and produces an open-frame holistic federated learning approach.

Although, in general, the methodology of the federated learning technique, the ensemble models, and recently the meta-learning methods have occupied the research community intensely, and relevant work has been proposed that has upgraded the relevant research area, this is the first time that such a comprehensive framework is presented in the international literature. The methodology offered herein is an advanced form of learning. The computational process is not limited to solving a problem but through a productive method of searching the solution space and selecting the optimal one in a meta-heuristic way [47,48].

On the other hand, the federated learning model should apply average aggregation methods to the set of cooperative training data. This raises serious concerns for the effectiveness of this universal approach and, therefore, for the validity of federated architectures in general. Generally, it flattens the unique needs of individual users without considering the local events to be managed. How one can create personalized intelligent models that solve the above limitations is currently a prominent research problem. For example, the study [49] is based on the needs and events that each user must address in a federated format. Explanations are the assortment of characteristics of the interpretable system, which, in the case of a specified illustration, helped to bring about a conclusion and provided the function of the model on both local and global levels. Retraining is suggested only for those features for which the degree of change is considered quite important for the evolution of its functionality.

Essential topics that could expand the research area of the proposed framework concern the Meta-Ensemble Learning process and, specifically, how to solve the problem of creating trees and their depth so that the process is automatically fully simplified. An automated process should also be identified for pruning each tree with optimal separations to avoid negative gain. Finally, explore procedures to add an optimally trimmed tree version to the model to maximize frame efficiency, accuracy, and speed.

Author Contributions: Conceptualization, K.D. and P.K.; methodology, K.D., P.K., and K.R.; software, X.X.; validation, K.D., P.K., C.S., K.R., L.I., and G.S.; formal analysis, C.S., K.R., and L.I.; investigation, K.D. and P.K.; resources, C.S., K.R., L.I. and G.S.; data curation, K.D., P.K., C.S., K.R., L.I., and G.S.; writing—original draft preparation, K.D. and P.K.; writing—review and editing, K.D., P.K., C.S., K.R., L.I., and G.S.; visualization, K.D. and K.R.; supervision, P.K. and C.S.; project administration, L.I. and G.S.; funding acquisition, C.S. and G.S. 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: Data Availability https://www.unb.ca/cic/datasets/ddos-2019.html.

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

References

1. Nikoloudakis, Y.; Kefaloukos, I.; Klados, S.; Panagiotakis, S.; Pallis, E.; Skianis, C.; Markakis, E. Towards a Machine Learning Based Situational Awareness Framework for Cybersecurity: An SDN Implementation. *Sensors* **2021**, *21*, 4939. https://doi.org/10.3390/s21144939.
2. Wang, W.; Liu, H.; Lin, W.; Chen, Y.; Yang, J.-A. Investigation on Works and Military Applications of Artificial Intelligence. *IEEE Access* **2020**, *8*, 131614–131625. https://doi.org/10.1109/access.2020.3009840.
3. Lukman, S.; Nazaruddin, Y.Y.; Joelianto, E.; Ai, B. Military 5G Mobile Networking as Driver of UAVs in Detecting RAM for Stealth Operation. In Proceedings of the 2019 IEEE 6th Asian Conference on Defence Technology (ACDT), Bali, Indonesia, 13–15 November 2019; pp. 91–96. https://doi.org/10.1109/acdt47198.2019.9072805.
4. Ghataoura, D.; Ogbonnaya, S. Application of Image Captioning and Retrieval to Support Military Decision Making. In Proceedings of the 2021 International Conference on Military Communication and Information Systems (ICMCIS), The Hague, Netherlands 4–5 May 2021; pp. 1–8. https://doi.org/10.1109/icmcis52405.2021.9486395.
5. Drosatos, G.; Rantos, K.; Karampatzakis, D.; Lagkas, T.; Sarigiannidis, P. Privacy-preserving solutions in the Industrial Internet of Things. In Proceedings of the 2020 16th International Conference on Distributed Computing in Sensor Systems (DCOSS), Marina del Rey, CA, USA, 25–27 May 2020; pp. 219–226. https://doi.org/10.1109/dcoss49796.2020.00044.
6. D'Acquisto, G.; Domingo-Ferrer, J.; Kikiras, P.; Torra, V.; de Montjoye, Y.-A.; Bourka, A. Privacy by design in big data: An overview of privacy enhancing technologies in the era of big data analytics. *arXiv* **2015**, arXiv:151206000 Cs,. https://doi.org/10.2824/641480.
7. Neumann, N.M.P.; van Heesch, M.P.P.; Phillipson, F.; Smallegange, A.A.P. Quantum Computing for Military Applications. In Proceedings of the 2021 International Conference on Military Communication and Information Systems (ICMCIS), The Hague, Netherlands 4–5 May 2021; pp. 1–8. https://doi.org/10.1109/icmcis52405.2021.9486419.
8. Demertzis, K.; Tsiknas, K.; Takezis, D.; Skianis, C.; Iliadis, L. Darknet Traffic Big-Data Analysis and Network Management for Real-Time Automating of the Malicious Intent Detection Process by a Weight Agnostic Neural Networks Framework. *Electronics* **2021**, *10*, 781. https://doi.org/10.3390/electronics10070781.
9. Demertzis, K.; Iliadis, L.; Pimenidis, E.; Tziritas, N.; Koziri, M.; Kikiras, P.; Tonkin, M. Federated Blockchained Supply Chain Management: A CyberSecurity and Privacy Framework. In *Artificial Intelligence Applications and Innovations*; Springer: Cham, Switzerland, 2021; pp. 769–779. https://doi.org/10.1007/978-3-030-79150-6_60.
10. Demertzis, K.; Iliadis, L.; Pimenidis, E.; Tziritas, N.; Koziri, M.; Kikiras, P. Blockchained Adaptive Federated Auto MetaLearning BigData and DevOps CyberSecurity Architecture in Industry 4.0. In Proceedings of the 22nd Engineering Applications of Neural Networks Conference, Crete, Greece, 25–27 June 2021; Springer: Cham, Swizterland, 2021; pp. 345–363. https://doi.org/10.1007/978-3-030-80568-5_29.
11. Chang, Y.; Fang, C.; Sun, W. A Blockchain-Based Federated Learning Method for Smart Healthcare. *Comput. Intell. Neurosci.* **2021**, *2021*, e4376418. https://doi.org/10.1155/2021/4376418.
12. Alam, K.M.R.; Siddique, N.; Adeli, H. A dynamic ensemble learning algorithm for neural networks. *Neural Comput. Appl.* **2020**, *32*, 8675–8690. https://doi.org/10.1007/s00521-019-04359-7.
13. Hou, R.; Tang, F.; Liang, S.; Ling, G. Multi-Party Verifiable Privacy-Preserving Federated k-Means Clustering in Outsourced Environment. *Secur. Commun. Networks* **2021**, *2021*, e3630312. https://doi.org/10.1155/2021/3630312.
14. Xu, G.; Li, H.; Liu, S.; Yang, K.; Lin, X. VerifyNet: Secure and Verifiable Federated Learning. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 911–926. https://doi.org/10.1109/tifs.2019.2929409.
15. Zheng, H.; Hu, H.; Han, Z. Preserving User Privacy for Machine Learning: Local Differential Privacy or Federated Machine Learning?. *IEEE Intell. Syst.* **2020**, *35*, 5–14. https://doi.org/10.1109/mis.2020.3010335.

16. Felbab, V.; Kiss, P.; Horváth, T. Optimization in Federated Learning. In Proceedings of the Conference on Theory and Practice of Information Technologies, 2019.
17. Alsaleh, M.N.; Al-Shaer, E. Automated Cyber Risk Mitigation: Making Informed Cost-Effective Decisions. In *Adaptive Autonomous Secure Cyber Systems*; Jajodia, S., Cybenko, G., Subrahmanian, V.S., Swarup, V., Wang, C., Wellman, M., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 131–157. https://doi.org/10.1007/978-3-030-33432-1_7.
18. Feurer, M.; Klein, A.; Eggensperger, K.; Springenberg, J.T.; Blum, M.; Hutter, F. Efficient and Robust Automated Machine Learning. p. 9.
19. Ferrag, M.A.; Friha, O.; Maglaras, L.; Janicke, H.; Shu, L. Federated Deep Learning for Cyber Security in the Internet of Things: Concepts, Applications, and Experimental Analysis. *IEEE Access* **2021**, *9*, 138509–138542. https://doi.org/10.1109/access.2021.3118642.
20. Feurer, M.; Eggensperger, K.; Falkner, S.; Lindauer, M.; Hutter, F. Auto-Sklearn 2.0: Hands-free AutoML via Meta-Learning. *arXiv* **2021**, arXiv:2007.04074, Sep. https://doi.org/10.48550/arXiv.2007.04074.
21. Kotthoff, L.; Thornton, C.; Hoos, H.H.; Hutter, F.; Leyton-Brown, K. Auto-WEKA: Automatic Model Selection and Hyperparameter Optimization in WEKA. In *Automated Machine Learning: Methods, Systems, Challenges*; Hutter, F., Kotthoff, L., Vanschoren, J., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 81–95. https://doi.org/10.1007/978-3-030-05318-5_4.
22. Bing, X. Critical infrastructure protection based on memory-augmented meta-learning framework. *Neural Comput. Appl.* **2020**, *32*, 17197–17208. https://doi.org/10.1007/s00521-020-04760-7.
23. Romero, V.; Ly, M.; Rasheed, A.H.; Charrondière, R.; Lazarus, A.; Neukirch, S.; Bertails-Descoubes, F. Physical validation of simulators in computer graphics: A new framework dedicated to slender elastic structures and frictional contact. *ACM Trans. Graph.* **2021**, *40*, 66:1–66:19. https://doi.org/10.1145/3450626.3459931.
24. Garrett, A.J.M. Review: Probability Theory: the Logic of Science, by E. T. Jaynes. *Law Probab. Risk* **2004**, *3*, 243–246. https://doi.org/10.1093/lawprj/3.3-4.243.
25. Platis, A.; Limnios, N.; Le Du, M. Dependability analysis of systems modeled by non-homogeneous Markov chains. *Reliab. Eng. Syst. Saf.* **1998**, *61*, 235–249. https://doi.org/10.1016/s0951-8320(97)00073-2.
26. Cavazos-Cadena, R.; Montes-De-Oca, R. The Value Iteration Algorithm in Risk-Sensitive Average Markov Decision Chains with Finite State Space. *Math. Oper. Res.* **2003**, *28*, 752–776. https://doi.org/10.1287/moor.28.4.752.20515.
27. Ghannoum, E.; Kieloch, Z. Use of modern technologies and software to deliver efficient design and optimization of 1380 km long bipole III 500 kV HVDC transmission line, Manitoba, Canada. In Proceedings of the PES T&D 2012, Orlando, FL, USA, 7–10 May 2012; pp. 1–6. https://doi.org/10.1109/tdc.2012.6281455.
28. Gupta, S.; Al-Obaidi, S.; Ferrara, L. Meta-Analysis and Machine Learning Models to Optimize the Efficiency of Self-Healing Capacity of Cementitious Material. *Materials* **2021**, *14*, 4437. https://doi.org/10.3390/ma14164437.
29. Zioviris, G.; Kolomvatsos, K.; Stamoulis, G. On the Use of a Sequential Deep Learning Scheme for Financial Fraud Detection. In *Intelligent Computing*; Springer: Cham, Switzerland, 2021; pp. 507–523. https://doi.org/10.1007/978-3-030-80126-7_37.
30. Demertzis, K.; Iliadis, L.; Anezakis, V.-D. A Dynamic Ensemble Learning Framework for Data Stream Analysis and Real-Time Threat Detection. In *Artificial Neural Networks and Machine Learning–ICANN*; Springer: Cham, Switzerland, 2018; pp. 669–681. https://doi.org/10.1007/978-3-030-01418-6_66.
31. Zioviris, G.; Kolomvatsos, K.; Stamoulis, G. Credit card fraud detection using a deep learning multistage model. *J. Supercomput.* **2022**, *78*, 14571–14596. https://doi.org/10.1007/s11227-022-04465-9.
32. Ahmadlou, M.; Adeli, H. Enhanced probabilistic neural network with local decision circles: A robust classifier. *Integr. Comput. Aided Eng.* **2010**, *17*, 197–210. https://doi.org/10.3233/ICA-2010-0345.
33. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.-Y. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in Neural Information Processing Systems*; 2017, Volume 30. Available online: https://papers.nips.cc/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html (accessed on 30 May 2022)
34. Bellare, M.; Goldreich, O. On Probabilistic versus Deterministic Provers in the Definition of Proofs of Knowledge. In *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation*; Avigad, L., Bellare, M., Brakerski, Z., Goldwasser, S., Halevi, S., Kaufman, T., Levin, L., Nisan, N., Ron, D., Sudan, M., et al.; Eds.; Springer: Berlin/Heidelberg, Germany, 2011; pp. 114–123. https://doi.org/10.1007/978-3-642-22670-0_14.
35. Yu, Z.; Chen, F.; Liu, J.K. Sampling-Tree Model: Efficient Implementation of Distributed Bayesian Inference in Neural Networks. *IEEE Trans. Cogn. Dev. Syst.* **2020**, *12*, 497–510. https://doi.org/10.1109/tcds.2019.2927808.
36. Melicher, V.; Slodička, M. Determination of missing boundary data for a steady-state Maxwell problem. *Inverse Probl.* **2006**, *22*, 297–310. https://doi.org/10.1088/0266-5611/22/1/016.
37. Wang, X.; Tu, S.; Zhao, W.; Shi, C. A novel energy-based online sequential extreme learning machine to detect anomalies over real-time data streams. *Neural Comput. Appl.* **2022**, *34*, 823–831. https://doi.org/10.1007/s00521-021-05731-2.
38. Chauhan, R.; Heydari, S.S. Polymorphic Adversarial DDoS attack on IDS using GAN. In Proceedings of the 2020 International Symposium on Networks, Computers and Communications (ISNCC), Montreal, QC, Canada, 20–22 October 2020; pp. 1–6. https://doi.org/10.1109/isncc49221.2020.9297264.
39. Hellen, N.; Marvin, G. Explainable AI for Safe Water Evaluation for Public Health in Urban Settings. in 2022 International Conference on Innovations in Science, Engineering and Technology (ICISET), Chittagong, Bangladesh, 26–27 February 2022; pp. 1–6. https://doi.org/10.1109/iciset54810.2022.9775912.

40. Demertzis, K.; Iliadis, L. GeoAI: A Model-Agnostic Meta-Ensemble Zero-Shot Learning Method for Hyperspectral Image Analysis and Classification. *Algorithms* **2020**, *13*, 61. https://doi.org/10.3390/a13030061.
41. Demertzis, K.; Iliadis, L. Detecting invasive species with a bio-inspired semi-supervised neurocomputing approach: the case of Lagocephalus sceleratus. *Neural Comput. Appl.* **2017**, *28*, 1225–1234. https://doi.org/10.1007/s00521-016-2591-2.
42. Holloway, L.E.; Krogh, B.H.; Giua, A. A Survey of Petri Net Methods for Controlled Discrete Event Systems. *Discret. Event Dyn. Syst.* **1997**, *7*, 151–190. https://doi.org/10.1023/a:1008271916548.
43. nbsp;Lu, Y.; Huang, X.; Li, D.; Zhang, Y. Collaborative Graph-Based Mechanism for Distributed Big Data Leakage Prevention. In Proceedings of the 2018 IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, United Arab Emirates, 9–13 December 2018; pp. 1–7.
44. Tsimenidis, S.; Lagkas, T.; Rantos, K. Deep Learning in IoT Intrusion Detection. *J. Netw. Syst. Manag.* **2021**, *30*, 8. https://doi.org/10.1007/s10922-021-09621-9.
45. Iezzi, M. Practical Privacy-Preserving Data Science With Homomorphic Encryption: An Overview. In Proceedings of the 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 10–13 December 2020; pp. 3979–3988. https://doi.org/10.1109/bigdata50022.2020.9377989.
46. Apiecionek, L.; Makowski, W.; Biernat, D.; Lukasik, M. Practical implementation of AI for military airplane battlefield support system. in 2015 8th International Conference on Human System Interaction (HSI), Warsaw, Poland, 25–27 June 2015; pp. 249–253. https://doi.org/10.1109/hsi.2015.7170675.
47. Sharma, A.; Rani, R. C-HMOSHSSA: Gene selection for cancer classification using multi-objective meta-heuristic and machine learning methods. *Comput. Methods Programs Biomed.* **2019**, *178*, 219–235. https://doi.org/10.1016/j.cmpb.2019.06.029.
48. Huang, B.; Sun, Y.; Sun, Y.-M.; Zhao, C.-X. A hybrid heuristic search algorithm for scheduling FMS based on Petri net model. *Int. J. Adv. Manuf. Technol.* **2010**, *48*, 925–933. https://doi.org/10.1007/s00170-009-2329-8.
49. Demertzis, K.; Iliadis, L.; Kikiras, P.; Pimenidis, E. An explainable semi-personalized federated learning model. *Integr. Comput. AidedEng.* 2022, *Preprint*, 1–16. https://doi.org/10.3233/ica-220683.

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.