WFE-Tab

Overcoming limitations of TabPFN in IIoT-MEC environments with a weighted fusion ensemble-TabPFN model for improved IDS performance

Ruiz-Villafranca, Sergio; Roldán-Gómez, José; Carrillo-Mondéjar, Javier; Martinez, José Luis; Gañán, Carlos H.

DOI
10.1016/j.future.2025.107707

Publication date
2025

Document Version
Final published version

Published in
Future Generation Computer Systems

Citation (APA)
Ruiz-Villafranca, S., Roldán-Gómez, J., Carrillo-Mondéjar, J., Martinez, J. L., & Gañán, C. H. (2025). WFE-Tab: Overcoming limitations of TabPFN in IIoT-MEC environments with a weighted fusion ensemble-TabPFN model for improved IDS performance. *Future Generation Computer Systems*, *166*, Article 107707. https://doi.org/10.1016/j.future.2025.107707

Future Generation Computer Systems 166 (2025) 107707

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

WFE-Tab: Overcoming limitations of TabPFN in IIoT-MEC environments with a weighted fusion ensemble-TabPFN model for improved IDS performance

Sergio Ruiz-Villafranca [a,*], José Roldán-Gómez [b,1], Javier Carrillo-Mondéjar [b,1], José Luis Martinez [a,1], Carlos H. Gañán [c,1]

[a] University of Castilla-La Mancha, Albacete Research Institute of Informatics, Investigación 2, Albacete 02071, Spain
[b] Universidad de Zaragoza, Zaragoza, Spain
[c] Delft University of Technology, The Netherlands

ARTICLE INFO

Dataset link: https://github.com/C4denaX/WFE-Tab

Keywords:
CyberSecurity
Multi-access edge computing
Machine learning
Intrusion Detection System
Industrial Internet of Things

ABSTRACT

In recent years we have seen the emergence of new industrial paradigms such as Industry 4.0/5.0 or the Industrial Internet of Things (IIoT). As the use of these new paradigms continues to grow, so do the number of threats and exploits that they face, which makes the IIoT a desirable target for cybercriminals. Furthermore, IIoT devices possess inherent limitations, primarily due to their limited resources. As a result, it is often impossible to detect attacks using solutions designed for other environments. Recently, Intrusion Detection Systems (IDS) based on Machine Learning (ML) have emerged as a solution that takes advantage of the large amount of data generated by IIoT devices to implement their functionality and achieve good performance, and the inclusion of the Multi-Access Edge Computing (MEC) paradigm in these environments provides the necessary computational resources to deploy IDS effectively. Furthermore, TabPFN has been considered as an attractive option for solving classification problems without the need to reprocess the data. However, TabPFN has certain drawbacks when it comes to the number of training samples and the maximum number of different classes that the model is capable of classifying. This makes TabPFN unsuitable for use when the dataset exceeds one of these limitations. In order to overcome such limitations, this paper presents a Weighted Fusion-Ensemble-based TabPFN (WFE-Tab) model to improve IDS performance in IIoT-MEC scenarios. The presented study employs a novel weighted fusion method to preprocess data into multiple subsets, generating different ensemble family TabPFN models. The resulting WFE-Tab model comprises four stages: data collection, data preprocessing, model training, and model evaluation. The performance of the WFE-Tab method is evaluated using key metrics such as Accuracy, Precision, Recall, and F1-Score, and validated using the Edge-IIoTset public dataset. The performance of the method is then compared with baseline and modern methods to evaluate its effectiveness, achieving an F1-Score performance of 99.81%.

1. Introduction

The industrial sector has undergone a huge transformation since the First Industrial Revolution. The most recent phase in this is referred to as Industry 4.0 or the Industrial Internet of Things (IIoT), and is characterized by the integration of emerging technologies such as the Internet of Things (IoT), Big Data, 5G networks, and innovative applications like artificial vision, Deep Learning (DL), and Machine Learning (ML) models for industrial process optimization [1]. The integration of these modern technologies with traditional Operational Technology (OT) services has facilitated this transformation, giving rise to a coexistence that has led the industrial sector into a new era of innovation and growth [2]. In this context, the integration of IIoT technologies into traditional industries presents a unique set of challenges. These challenges arise from the need to maintain the integrity and efficiency of existing industrial processes while simultaneously incorporating new technologies that can optimize them. One promising solution is the use of Multi-Access Edge Computing (MEC) [3]. MEC is a cloud-based service environment that provides computing capabilities at the edge of the network, and thus closer to the source of data providing multiple benefits in terms of low-latency and scalability.

* Corresponding author.
E-mail addresses: sergio.rvillafranca@uclm.es (S. Ruiz-Villafranca), roldangjose@uniovi.es (J. Roldán-Gómez), jcarrillo@unizar.es (J. Carrillo-Mondéjar), joseluis.martinez@uclm.es (J.L. Martinez), c.hernandezganan@tudelft.nl (C.H. Gañán).

[1] Contributing author.

https://doi.org/10.1016/j.future.2025.107707
Received 31 May 2024; Received in revised form 3 January 2025; Accepted 5 January 2025
Available online 18 January 2025

However, the integration of the IIoT and MEC has introduced new cybersecurity vulnerabilities that must be addressed. These vulnerabilities are especially noticeable in traditional industrial devices, which often lack regular updates and have limited capabilities [4,5], making them prime targets for cyberattackers. The Intrusion Detection System (IDS), which is an application that monitors network traffic to identify known threats and suspicious or malicious activity, is thus a crucial element in securing IIoT networks. For that reason, the use of ML algorithms for developing IDS has become a powerful strategy for securing IIoT-MEC environments [6,7]. These ML-based IDSs can analyse data left by attackers on their networks or applications, making them an effective method for monitoring, preventing (or at least mitigating) the impacts of attacks [8]. Moreover, ML algorithms can be trained to adapt to new types of attacks, making them particularly effective against zero-day ones. This adaptability, coupled with the ability to provide monitoring and response features, significantly reduces the window of opportunity for attackers [9,10]. Nevertheless, the efficacy of this IDS-based ML is largely tied to the expertise of the developer tasked with processing the collected data to enhance its performance and extract valuable insights, thereby selecting the optimal model for attack detection and classification.

In recent years, new approaches have emerged in the field of ML, such as Automated Machine Learning (AutoML), which aims to automate these repetitive tasks in the ML pipeline. This significantly speeds up the model-building process and allows data scientists to focus on higher value-added duties [11]. One of the most promising approaches is the TabPFN model [12], which is a variant of Prior-Data Fitted Networks (PFN). It has the unique ability to achieve high performance on some tabular classification problems [13]. This is even the case when the data set is small and computational resources are limited. However, its utility is somewhat constrained by inherent limitations related to the number of training samples, which is up to 1000 samples, and the number of classes, which is up to 10. These constraints restrict its applicability across a broader range of environments and applications, which includes the deployment of an IDS based on ML. Due to huge amount of data present in IIoT-MEC environments and the numerous types of attacks present at the network layer, it is quite challenging to utilize this technique in such conditions [14]. As a result, the data employed may not offer sufficient diversity for each class to provide the model with the necessary information to differentiate between the various attacks. Furthermore, due to the limitations imposed by the number of classes, the developers were unable to provide a focus detection for the specific techniques employed by the attacks [15,16].

The objective of this paper is to improve the performance and overcome the limitations of training samples and classes for classification in the TabPFN model [12], and enable its use in an IDS in an IIoT-MEC environment. To ensure the model's applicability beyond network traffic classification, we conduct a comprehensive study and develop and test various ensemble-based methodologies to address the limitations of TabPFN that may affect its performance or application. Finally, we present the Weighted Fusion-Ensemble-based TabPFN (WFE-Tab) model, which utilizes a novel fusion pre-processing technique to divide the collected data into multiple subsets. This approach allows the development of multiple specialized sets of TabPFN models, resulting in more accurate decisions during the aggregation stage of the ensemble method. Furthermore, this approach considers that every specialized set must validate its knowledge to inform its decision about the different types of traffic during classification. Additionally, the classification of unknown attacks is considered in order to improve the real-world deployment of our implementation. In summary, the major contributions presented in this paper are the following:

- An in-depth performance analysis of the TabPFN model for network traffic classification was conducted using the Edge-IIoTset public dataset [17], which was specifically designed for IIoT-MEC scenarios. The analysis provided useful information about the strengths and limitations of the models in these particular contexts. To enhance the performance of the TabPFN model in these contexts, ensemble techniques were developed.
- In this study, we introduce a new TabPFN-based model which incorporates a weighted fusion-based approach. The model is capable of handling an extensive range of classification classes, encompassing over 10, and a substantial number of training instances, exceeding 1000. Furthermore, its incorporation into the IDS architecture enables the detection of anomalous events.
- A comprehensive performance analysis of the proposed approach is provided, utilizing a range of evaluation metrics to illustrate the efficacy of the model in addressing imbalanced classification problems. A comparison is provided with other baseline models and algorithms, including Random Forest (RF) and boosting algorithms, which have demonstrated efficacy in addressing tabular data problems [18]. This comparison allows us to evaluate our approach in relation to established methods.

The rest of the paper is organized as follows. Section 2 presents an overview of IDS-based ML approaches, while Section 3 explains the evolution of the different approaches to enhancing TabPFN until the WFE-Tab model implementation is reached is explained. Section 4 explains the assumptions made during the execution of the IDS architecture in our study. The workflow of the experimentation in our study of WFE-Tab and the rest of the ML techniques and TabPFN approaches considered is detailed in Section 5. Section 6 present an assessment of the performance of all the models considered in our study. Finally, Section 7 contains the conclusions and lines for future work.

2. Related work

In this section, we present a synopsis of the most relevant pieces of research on IDS in IIoT environments, where the proposed research makes use of FL, ML, DL or Deep Q-Network (DQN) learning techniques. The focus is on studies that bear a resemblance to our work and show a high degree of relevance in this research domain.

2.1. IDS based on FL approaches

The authors in [19] present a cyber threat intelligence framework for securing IIoT environments. It is based on federated learning and information fusion (FL-CTIF) and uses a cloud-based security auditor to design and test updated models for detecting various cyberattacks, such as man-in-the-middle, SSL-based, and DNS flood attacks. Information fusion techniques are employed to merge features from different datasets, which are ToN-IoT and CIC-DDoS2019. A proposed federated learning-based Artificial Neural Network (ANN) model aims to reduce model training rounds and CPU consumption based on satisfaction level and average accuracy score. The model is evaluated in a digital twin-based IIoT environment and demonstrates improved performance in terms of F1 score, accuracy, recall, and true negative rate compared with existing methods, achieving an overall F1-score performance of 98.73% for each case.

The authors in [20] propose Fed-Inforce-Fusion, which is a federated reinforcement-based fusion model designed to enhance the security and privacy protection of IoMT networks against cyber-attacks. The model utilizes Q-learning to learn the latent relationships of medical data and detect complex attack vectors. Additionally, the model employs a fusion/aggregation strategy to improve detection performance and reduce communication overhead by allowing participating clients to dynamically join the federation process. The model was evaluated on a real-world IoMT dataset, and the results demonstrate higher accuracy and detection rates than existing benchmark methods. Additionally, it exhibits better communication efficiency and privacy preservation.

In [21], the authors present Fed-ANIDS, a distributed network intrusion detection method that utilizes federated learning and autoencoders. Fed-ANIDS facilitates the secure and collaborative learning of a global model for network intrusion detection by enabling each entity in the system to learn locally with its own data. The authors utilize three types of autoencoders: simple autoencoders, variational autoencoders, and adversarial autoencoders. They also compare two federated learning algorithms, namely FedAvg and FedProx. The proposed method is evaluated on three well-known datasets: USTC-TFC2016, CIC-IDS2017, and CSE-CIC-IDS2018. The results show high performance in terms of accuracy, F1-score, and false discovery rate, while preserving data privacy. The authors demonstrate the capacity for generalization of Fed-ANIDS with unseen datasets and compare it with centralized learning and Generative Adversarial Network (GAN)-based models.

The work in [22] introduces DAFL, an innovative FL system for network intrusion detection. DAFL expands the data resources available for training while preserving data privacy, resulting in an improvement in the detection accuracy of the global model. The authors compare the performance of DAFL with other federated and centralized learning schemes. Comparisons are made in convergence speed, recognition performance and communication overhead. The experimental results show that DAFL outperforms other schemes in several respects when using the CSE-CIC-IDS2018 dataset. It achieves a higher convergence speed, which means it can learn and adapt more quickly. Additionally, it delivers a higher detection performance, making it more effective at identifying network intrusions, with a F1-Score performance of 93.3%. Finally, DAFL has a lower communication overhead, making it more efficient and scalable.

In [23], a novel federated deep learning intrusion detection system (FEDGAN-IDS) for IoT systems is introduced. The paper aims to provide a clear and concise explanation of the proposed system and its architecture. The system is designed with a distributed GAN architecture, where each IoT device has a local generator and discriminator network, and the edge node has a central generator and discriminator network. The local generators generate synthetic data to augment the local datasets. These augmented datasets train the local discriminators, which act as classifiers. The local model parameters are periodically aggregated by the central networks using FL, ensuring a global model that benefits from the learning of all local models. The authors evaluate FEDGAN-IDS on three standard datasets: KDD99, NSL-KDD, and UNSW-NB15. The results indicate that FEDGAN-IDS achieves a high F1-Score of 98.53% and a high convergence rate.

Finally, all the FL IDS proposals analysed in this section exhibit common limitations. Firstly, none of the proposals consider detecting anomalous behaviour beyond the attacks present in the dataset. This lack of functionality may result in the IDS failing to detect zero-day attacks in the network. Secondly, most of the proposals employ a DL algorithm as the client model. Due to the computational limitations of typical IIoT devices and gateways, it is necessary to analyse algorithms and adapt them in order to use them on these devices. Our proposal overcomes these limitations by introducing the ability to inherently detect anomalous classes in our model and incorporating the MEC paradigm into our IDS architecture design. Furthermore, our proposal achieves 99.54% and 99.70% for the F1-Score and Accuracy metric, respectively, which is an improvement with 1%–6% compared to other proposals.

2.2. IDS based on ML approaches

The authors of [24] conducted a thorough study of various ensemble ML methodologies, together with a feature selection classifier, designed to detect and prevent intrusions and attacks in IIoT networks. The study analysed Xgboost [25], Bagging, Random Forest (RF) [26], Extremely Randomized Trees (ERT) [27], and Adaptive Boosting (Adaboost) [28] models. The dataset's feature selection process uses the Chi-Square Statistical method, a robust statistical test that measures the dependence between stochastic variables. This method identifies the most relevant features that significantly contribute to the model's predictive power. To assess the efficacy of each model, the authors use seven distinct datasets derived from the Telemetry of Networks for IoT (ToN-IoT) dataset [29]. The analysis reveals that the Xgboost model outperforms the other models in terms of overall performance over all the datasets used for evaluation. The Xgboost model achieves an impressive F1 score of 98.30%, indicating a high degree of precision and recall in its predictions.

The development of an IDS based on an RF model is the subject of the research described in [30]. The approach uses Pearson's Correlation Coefficient (PCC) and Isolation Forest (IF) to perform feature selection and reduce the dimensionality of the datasets used for model training and evaluation. The performance of each model individually and in combination is evaluated in the study, which used two distinct datasets, namely Bot-IoT [31] and WUSTL-IIOT-2021 [32], to assess the models. The datasets offer a comprehensive and diverse range of data for model assessment. The study's results show that the proposed approach, which combines the three models under consideration, outperforms the other baseline models and combinations. This is supported by an overall F1 score of 93.57%.

In [33], the authors propose a new architecture for an IDS using MEC in IIoT environments. The main goal of this architecture is to take advantage of MEC's capabilities, especially in network management and computational resources. The article presents an innovative architecture and conducts a comprehensive study to assess the performance and computational cost differences between various boosting tree algorithms. The study evaluates five models, namely Xgboost, LightGBM [34], AdaBoost, CatBoost [35], and GradientBoosting [36], which are evaluated on a custom dataset that considers traditional industrial protocols such as Modbus/TCP, OPC Unified Architecture (OPC UA), and S7 communications (S7COMM). The evaluation assesses the models' ability to detect various attacks, including scanning techniques, DDoS attacks, packet manipulation, and attacks on web applications. The study found that the Xgboost classifier performs better than other classifiers when dealing with multiple attacks, achieving an impressive F1 score of 99%. Additionally, the study shows that the LightGBM classifier has a better classification ratio and computational cost.

The authors of [37] examine the use of GPT and interpolation-based data augmentation for multiclass intrusion detection in IIoT networks. They compare different data augmentation techniques and their impact on five intrusion detection algorithms, such as Decision Tree (DT) [38] or Xgboost. The study concludes that the advantages of data augmentation are specific to the algorithm and data used. Xgboost outperforms all the other classifiers but does not benefit from data augmentation. This highlights the challenges of using GPT-based methods for tabular data generation, as they were found to generate invalid data, which negatively affects classification performance. The paper presents a systematic evaluation of data augmentation methods for intrusion detection models.

By taking application and transport layer features as a basis, supervised ML was applied in [39]. The researchers developed feature clusters for flow, MQTT, and TCP, using the UNSW-NB15 data-set, and they applied supervised ML algorithms, such as RF and Support Vector Machine (SVM), to these clusters. Their proposal achieved high accuracy and a low training time in binary and multi-class classification of normal and malicious packets. When compared with other contemporary approaches, the proposed feature clusters demonstrated significant advantages, achieving an accuracy of 97.37% when using the RF algorithm for multiclass problems.

The proposals reviewed in this section have one main limitation: they do not detect unknown attacks. It is crucial to consider this functionality in IDSs deployed in IIoT scenarios due to the constant evolution and discovery of new vulnerabilities in this environment. Additionally, each proposal focuses on optimizing the preprocessing stage of the data using different techniques. Our proposal overcomes both limitations effectively. Using TabPFN as a base model allows us to avoid the need for deep data preprocessing and reduces the complexity of the IDS proposal. Additionally, WFE-Tab is designed to directly detect anomalies in the specific application it will be deployed in. Finally, our proposal outperforms previous ones in terms of F1-Score and Accuracy metrics.

Table 1
Related work corresponding to proposals from other researchers.

Ref (Year)	Category	Technique(s)	Dataset	Anomalous detection?	Evaluation metric/Performance
[19] (2023)	FL	ANN	ToN-IoT and CIC-DDoS2019	No	F1-Score overall 98.73%
[20] (2024)		Q-Learning	ToN_IoT	No	Accuracy 94.40%
[21] (2023)		Autoencoders	USTC-TFC2016, CIC-IDS2017, and CSE-CIC-IDS2018	No	F1-Score overall 94.44%
[22] (2023)		DAFL	CSE-CIC-IDS2018	No	F1-Score 93.3%
[23] (2023)		GAN	KDD99, NSL-KDD, and UNSW-NB15	No	F1-Score overall 98.53%
[24] (2023)	ML	Xgboost, Bagging, RF, ET, Adaboost	ToN-IoT	No	Xgboost F1-Score 98.30%
[30] (2022)		RF with IF and PCC	Bot-IoT and WUSTL-IIoT-2021	No	F1-Score overall 93.57%
[33] (2023)		Xgboost, Adaboost, Catboost, GradientBoosting and LightGBM	Custom Dataset	No	Xgboost F1-Score 99%
[37] (2024)		Decision Tree (DT), RF, TabNet, Xgboost	Not specified	No	Xgboost F1-Score 91%
[39] (2021)		RF, SVM	UNSW-NB15	No	RF Accuracy 97.37%
[40] (2022)	DL	DenseNet, Inception Time	Edge-IIoT and UNSW-NB15	No	Both F1-Score overall 96.9%
[41] (2024)		NASP, NASP-MTC	USTC-TFC2016 and Edge-IIoTset	No	NASP-MTC F1-Score overall 99.42%
[42] (2023)		GraphAN	DAPT2020 and Edge-IIoTset	No	F1-Score overall 97.99%
[43] (2021)		ANN, DNN, RNN and LSTM	UNSW-NB15 modified	No	ANN/DNN Accuracy 99.59%
[44] (2023)		DNN with Proposed Feature Selection	NSL-KDD, UNSWNB-15, and CIC-IDS-2017	No	F1-Score overall 98.7%
[45] (2024)	DQN	DQN-HIDS	UNSW-NB15	Yes	F1-Score overall 70.33%
[46] (2024)		Double DQN-LP	Simulation	No	Detection Rate 90%
[47] (2024)		DQN-CVAE	TON-IoT	Yes	F1-Score 99.5%
[48] (2024)		MFGD3QN	Simulation	No	100% accuracy in detection of DoS attacks
Our proposal	FL	TabPFN with Fusion data clustering phase	Edge-IIoTset	Yes	F1-Score 99.81% Accuracy 99.57%

2.3. IDS based on DL approaches

The authors of [40] designed two deep learning models, DenseNet and Inception Time, to detect and classify cyber-attacks. The models were trained and tested on network traffic data from two datasets: Edge-IIoT, and UNSW-NB15. Their performance was evaluated using a comprehensive set of metrics, including accuracy, precision, recall, and F1-score. The DenseNet and Inception Time models outperformed not only traditional ML methods but also other deep learning architectures, achieving an F1-Score of 95.3% when using the Edge-IIoTset and 98.5% with the UNSW-NB15 dataset. The study also explored the application of the sliding window technique and class weights to tackle the challenges posed by imbalanced and heterogeneous data, which are common in network traffic data. The sliding window technique enhanced the models' feature extraction capabilities, while the use of class weights improved their generalization capabilities.

In [41] the authors made a significant contribution by developing an automatic and efficient method for classifying malware traffic. Their method, called Neural Architecture Search via Proximal Iterations (NASP), is designed to quickly search for the optimal neural network model based on network traffic in real-world environments. The search process is redefined as an optimization problem with discrete constraints of the model complexity. This allows NASP to adapt to different network traffic scenarios. The effectiveness of the NASP-MTC method was evaluated on two realistic datasets, namely USTC-TFC2016 and Edge-IIoTset. The results showed superior classification performance and lower model complexity compared with existing methods, with an overall F1-Score of 99.42%. This research represents a significant advancement in the field of malware traffic classification, offering a promising approach for enhancing IoT security.

In the realm of IIoT-enabled Cyber–Physical Systems (CPS), detecting and classifying Advanced Persistent Threat (APT) attacks is a critical challenge. The authors of [42] propose a novel method that leverages the power of Graph Attention Networks (GraphANs) to capture the complex and dynamic features of APT attacks. Their system consists of five layers, each contributing to the overall efficacy of the detection process. The method was validated using two publicly available datasets, namely DAPT2020 and Edge-IIoTset, and it achieved high detection accuracy and a low prediction time, outperforming conventional ML techniques. The F1 Score metric was 96.58% with the DAPT2020 dataset and 99.4% with the Edge-IIoTset dataset, indicating that their approach could provide effective protection against APT attacks in the IIoT-enabled CPS environment.

The researchers in [43] propose a deep intrusion detection system (Deep-IDS) that is based on three types of deep learning models: Artificial Neural Network (ANN), Deep Neural Network (DNN), and Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM). They utilized the UNSW-NB15 dataset, which contains realistic network traffic data with various types of attacks, and enhanced it by merging, cleaning, normalizing, and relabelling the data. The study compared the performance of their proposed Deep-IDS models with 12 other ML and DL models in terms of accuracy, loss, and training time. The ANN and DNN models proposed by the authors achieved the highest accuracy, with 99.59% accuracy for multi-class classification. This finding highlights the effectiveness of deep learning models for intrusion detection, particularly when using the enhanced UNSW-NB15 dataset.

In [44] the author propose a new feature selection technique for deep neural network-based intrusion detection systems. The technique is based on the fusion of the statistical importance of features using standard deviation and difference of mean and median. The authors evaluate their approach on three intrusion detection datasets, namely NSL-KDD, UNSWNB-15, and CIC-IDS-2017, and compare it with existing feature selection techniques. The results indicate that the proposed technique achieves better performance in terms of accuracy, precision, recall, F1-Score, false positive rate, and execution time. They also conducted a statistical test to validate the significance of their results. Their approach achieved an overall F1-Score performance of 98.7%.

DL algorithm-based approaches have an inherent limitation due to the computational cost of running these models on IIoT devices.

Depending on the deployment, some of the approaches may not be suitable for certain IIoT architectures. Another limitation that could prevent the proposals from reaching their optimal performance is the generalization, preprocessing, and feature selection of the data. Finally, as has been mentioned, none of the proposals using DL algorithms consider detecting anomalous or unknown attacks. Our proposal introduces an MEC layer to overcome the inherent limitations of the IIoT paradigm, and the design of WFE-Tab includes the detection of anomalous samples. Furthermore, our proposal's performance surpasses that of other approaches based on DL algorithms.

2.4. IDS based on DQN approaches

A heuristic IDS based on DQN for Social Internet of Things (SIoT) environments, named as DQN-HIDS, is proposed in [45]. It is designed to address the challenge of zero-day attacks in SIoT networks, which often lack sufficient training data. The DQN-HIDS employs a traffic processing module and a heuristic learning network, thereby enhancing the system's capacity to classify SIoT network traffic. A reward and penalty mechanism is introduced to impose a penalty on incorrect labelling actions and to adapt rewards to different actions. The results of the study demonstrate that DQN-HIDS is an effective approach for improving the accuracy of SIoT traffic labelling while reducing the workload of cybersecurity examiners. A comparison with state-of-the-art DL models, including CNN-BiLSTM, and ML methods, such as RF, shows that DQN-HIDS outperforms other methods in terms of F1-Score, with an average of 70.33% in testing scenarios with fewer training samples.

In [46], authors present a novel intrusion detection approach for the IIoT, which combines stochastic games and Double DQN with a "lazy penalty" mechanism to enhance performance. The strategy employs a dynamic, adversarial, stochastic-game model, which incorporates incomplete information to simulate interactions between IIoT attackers and defenders. The proposed system dynamically adjusts its detection strategies by analysing Nash equilibria and leveraging reinforcement learning. Simulation results demonstrate that this approach achieves higher detection rates and lower resource consumption compared to existing methods, such as Actor–Critic and Reinforce. In particular, the Double DQN-LP algorithm outperforms the baselines with a detection rate of 90%, while significantly reducing overhead through its "lazy penalty" mechanism.

The authors in [47] present a novel intrusion detection solution for the IIoT, one that is based on a DQN framework and has been specifically designed for open-set recognition problems. The solution incorporates a conditional variational autoencoder (CVAE) to augment the value network in DQN, thereby enabling it to distinguish between known and unknown malicious traffic. By treating the intrusion detection task as a discrete-time Markov decision process, the model efficiently classifies known traffic and uses reconstruction errors to detect unknown attacks. The authors conducted experiments utilizing the TON-IoT dataset, and their results demonstrate that the DC-IDS model exhibited superior performance compared to existing methodologies such as EVM, or an implementation of CVAE with extreme value theory (EVT), achieving an F1-score of 99.5%. Furthermore, the model demonstrated higher recognition rates for unknown attacks whilst maintaining consistent classification accuracy for known traffic.

In [48], a novel defence mechanism against Distributed Denial-of-Service (DDoS) attacks in edge intelligence environments is presented. It employs a Mean-Field Game (MFG) framework in conjunction with a Dueling Double Deep Q-Network (D3QN), designated as MFGD3QN. The objective of this method is to optimize the defence strategy for large-scale edge intelligence devices (EIDs) in scenarios where they are subjected to intense DDoS attacks. By modelling the interactions between attackers and defenders as a multi-agent problem, the system employs a combination of mean-field games and multi-agent deep reinforcement learning to compute optimal defence policies. The results of the simulation demonstrate that MFGD3QN outperforms traditional algorithms, including A2C, DDQN, D3QN, MFGDDQN, and MFGA2C. It achieves faster convergence and a higher defence success rate against DDoS attacks. In particular, MFGD3QN achieved a 0.2 reward by the 500th iteration and effectively mitigated all DDoS attacks in the simulation after 400 iterations.

DQN approaches for intrusion detection in IIoT environments present a number of significant challenges, primarily related to the high computational demand and the necessity for extensive training data, which may prove to be a barrier to their implementation in resource-constrained IIoT devices. Furthermore, these methods frequently encounter difficulties in adapting to evolving attack strategies, as they require constant retraining in order to maintain optimal performance.

To conclude, the literature review shows that there are several ways to implement IDS using FL, DL, ML, and DQN. These solutions achieve performance ratios measured by the F1-Score, a metric commonly used in imbalanced classification problems. However, most of these studies do not focus on the unique environment of IIoT-MEC and its inherent advantages. Furthermore, the distributed solutions proposed using FL techniques require a significant number of clients to achieve optimal performance. In this study, we introduce WFE-Tab, a novel model that is based on the TabPFN model and designed for the implementation of a smart IDS specifically tailored for IIoT-MEC scenarios. Our proposal can detect unknown attacks that the model has not been trained to recognize, a functionality which is absent from most other studies. Lastly, our proposal outperforms the majority of the studies considered in this section, even with a reduced number of clients, demonstrating the efficiency and effectiveness of our approach.

Table 1 presents a comparison of the current FL, DL, ML, and DQN approaches for IDS, as well as the original approach proposed in this work.

3. Methodology

This section outlines the procedural steps and reviews the previous approach taken before implementing the WEF-Tab model. The aim of the model is to address the limitations of TabPFN, specifically the limited number of training samples and classification classes. Additionally, there is a detailed description of the IDS architecture that WFE-Tab can use to detect and classify malicious traffic in an IIoT-MEC environment.

3.1. First development iterations

This section explains the incremental iterations made to reach our final proposal. We started with the base model and then incorporated various ensemble and clustering approaches to the TabPFN model.

3.1.1. TabPFN base model limitations

During the initial phase of this research, we attempted to implement an intrusion detection application using the TabPFN model. However, we found that this model, which has demonstrated robust performance in various applications [13,49], was unsuitable for the specific requirements of another real-world application that contains multiple classes, or that needs a large amount of data to perform well.

The TabPFN model lacked the specialized properties required for effective intrusion detection in an IIoT-MEC context, as it was originally designed for a wide range of applications. The features required for such environments include the ability to handle high volumes and varieties of data, as well as to accurately classify and respond to potential security threats.

When considering the complexity and diversity of IIoT-MEC environments, the limitations of the TabPFN model became particularly apparent. This context requires the model to have knowledge about the data from various IoT devices, each designed for specific tasks and producing unique types of data. Additionally, this fact makes the model training stage challenging, as it is necessary to check every 1000 sample combinations obtained from the environment, and ensure that there are sufficient samples from each class to provide the model with minimal knowledge.

On the basis of these considerations, it was concluded that the TabPFN model, despite its strengths, was not the optimal choice for this particular application or other applications with similar requirements, which could benefit from the use of TabPFN to reduce the complexity of their implementation. As a result, a new model was developed to address the limitations of TabPFN.

3.1.2. Ensemble approaches with TabPFN

After initial attempts with the TabPFN model, the research direction shifted towards implementing ensemble approaches. This strategic shift aimed to overcome the inherent limitations of the TabPFN model, particularly its inability to handle a large number of training samples.

The original design of the TabPFN model constrained it to work with a maximum of 1000 training samples. This highlighted a critical limitation in the TabPFN model's applicability to real-world applications.

To tackle this issue, we investigated the implementation of ensemble methods. This involved using multiple instances of the TabPFN model, known as client models, in conjunction. By doing so, a greater proportion of the data collected could be incorporated during the training phase, resulting in the model being exposed to a wider range of data samples.

In our research, we implemented two ensemble methods:

- **Sequential Majority Voting (SMV)** is a simple yet effective ensemble method. In this approach, each model is trained with a sequential subset of 1000 samples from the original dataset, and in the ensemble step, each model makes a prediction for each sample. The class that obtains the majority of votes across all models is chosen as the final prediction. This method is based on the principle that the collective wisdom of multiple models is likely to be more accurate than the prediction of any single model [50]. However, one limitation of Majority Voting is that it assumes all models in the ensemble are equally reliable, which may not always be the case.
- **Enhanced Majority Voting (EMV)** is another ensemble method that we implemented. Derived from SMV, EMV involves creating multiple random subsets of the original dataset, keeping a similar proportion of the classes in the subset to ensure correct performance. A separate model is trained on each subset, and for each sample, the final prediction is made by majority voting. EMV can reduce overfitting and reduce the determinism contributed by the order of the samples in the original dataset, and improve the stability and accuracy of ML algorithms [51]. However, the ensemble may not perform well if the models are too similar, as they may all make the same mistakes. This disadvantage is also present in the SMV proposal.

Nevertheless, this approach presented its own set of difficulties. Despite the increased number of training samples, the classification was limited to only 10 classes. Additionally, although the ensemble model was expected to perform better with the increased number of training samples, this was not consistently observed. During the combination decision process, each model in the ensemble was given equal weight. However, this approach did not consider the varying levels of information contained in different samples or the potential for misclassifications by individual models.

In some situations, certain models in the ensemble lacked sufficient information to classify accurately. In some instances, misclassifications made by models were propagated through the ensemble due to equal weighting. These issues highlight the necessity for a more nuanced approach to model weighting and combination in the ensemble.

3.1.3. Clustering-based Ensemble (CE) approach

In the following research phase, we utilized clustering algorithms to improve our model's performance, specifically the K-Means++ algorithm. This method involves an initial step of clustering the data, which generates subsets based on the produced clusters.

Each subset contains classes that are highly similar to each other, allowing us to create multiple families of TabPFN models. Each model family specializes in different types of classes contained in the dataset, as they are trained on a specific subset of data that best represents a particular class.

This approach has the significant advantage of identifying anomalous classes by designating one of the clusters as the 'anomalous' cluster, which contains packets that are significantly different from those in other clusters. These unusual packets are then distributed among the other subsets. This approach is especially effective in identifying unknown attacks, as it enables the model to flag any packet that does not fit well into any of the known attack types or benign packet types for the IDS.

This implementation allowed us to overcome several limitations of the basic TabPFN model. For example, the model could be trained with over 1000 samples, classify more than 10 families, and use more than 100 features depending on the dataset and use case. This represents a significant improvement with respect to the base model, which was limited by these constraints.

However, this approach also presents its own set of challenges. For example, classes with a low number of samples may not have the correct distribution across the clusters, and underrepresented classes in the training data could negatively impact the model's performance.

Furthermore, the fact that the ensemble's models have equal weight may lead to misclassification of similar packets generated by different attacks. This is due to the model's inability to distinguish subtle differences in the packets that indicate different types of attacks.

In order to classify anomalous attacks, it is necessary to establish a threshold that determines which attacks are considered anomalous or misclassified. This is because only one subfamily of models will correctly classify the class, and the rest will determine that the packet is anomalous. Determining the optimal threshold is a challenging task that requires a careful balance between sensitivity and specificity.

These issues highlight the need for a more nuanced approach to model weighting and combination in the ensemble. In spite of the challenges, the clustering-based ensemble approach is a promising direction for improving the performance of the TabPFN base model, and it forms the basis of our final WFE-Tab approach.

3.2. The WFE-Tab approach: A clustering fusion-based ensemble method

This methodology is an advanced version of the TabPFN model and previous iterations of this model detailed in Section 3.1, incorporating several enhancements to address its limitations and improve performance.

WFE-Tab introduces significant improvements in model weighting, data distribution and anomalous sample handling to the TabPFN model, as it is shown in Fig. 1. These improvements increase the accuracy and reliability of intrusion detection, making WFE-Tab a significant upgrade of TabPFN. The methodology of our purpose is divided into the following steps:

1. Data division by clustering. This step provides the distribution of the classes of different subsets in accordance with the clusters identified by the K-Means++ algorithm. In order to determine the number of clusters that the algorithm will consider, K-Means++ attempts to allocate each class to a cluster during 300 iterations, thereby ensuring optimal algorithm performance [52]. Following the completion of the algorithmic iterations, the resulting clusters will identify the subsets of classes that have converged. At the conclusion of this phase, the user may designate one of the subsets as anomalous. This will enable the

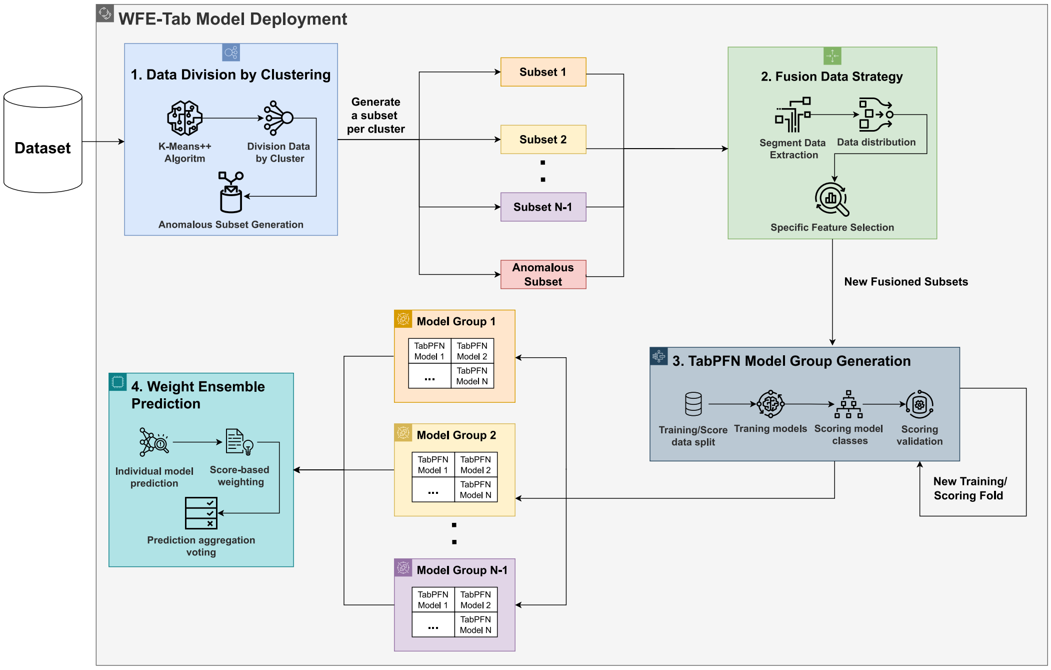

Fig. 1. Schematic WFE-Tab deployment.

model to determine that certain samples that have not been previously classified may be assigned to a specific class defined during this step.

The K-Means++ algorithm employs a mathematical representation of the clustering process, whereby the objective is to minimize the distance between data points and their assigned cluster centroids.

$$\arg\min_{C} \sum_{i=1}^{k} \sum_{x\in C_i} \|x-\mu_i\|^2 \tag{1}$$

The objective function for K-Means++ is described by Eq. (1), where x represents the samples found in the dataset used for the training of WFE-Tab, C_i is the ith cluster. μ_i is the centroid of the ith cluster, and $\|\cdot\|$ denotes the Euclidean distance.

2. Fusion data strategy. In this step, a small proportion of each subset selected by the user (approximately 1%–20% of the total subset) is redistributed to one or more other subsets to ensure cross-sample representation. In addition, a designated anomalous subset is partially included in the fusion process, thereby allowing the model's ability to detect anomalies. Let the data from each cluster C_i be represented as a subset D_i. A fusion process is applied where a portion $\alpha \cdot D_i$ is mixed with another subset D_j, where $\alpha \in [0.01, 0.20]$ and a portion $\beta \cdot D_a$, where D_a is the anomalous subset and $\beta \in [0, 0.10]$. This fusion process is represented by Eq. (2)

$$D_i^{\text{fused}} = D_i \cup (\alpha \cdot D_j) \cup (\beta \cdot D_a) \tag{2}$$

The fusion process ensures that each subfamily of models has some familiarity with classes from different subsets, which is the primary objective of this fusion strategy, even when an anomalous subset is included to prevent misclassification. The inclusion of an anomalous subset will not contain every heterogeneous class present in this subset, allowing for the classification of certain unknown samples without prior knowledge. This is accomplished by incorporating a diverse array of samples into the training data for each model. Although the number of samples from different classes may be limited, exposure to these samples can enhance the model's capacity to accurately recognize and classify various classes. Subsequently, a distinct feature selection process is conducted for each subset. This process involves the identification of the most pertinent features for each subset, employing the ERT algorithm.

3. Model group generation. The subsets are divided into two distinct sets: $D = D_{train} \cup D_{score}$, where D_{train} is the training set and D_{score} is the scoring set. D_{score} plays a significant role in determining the weight $w_j = \{w_{jc_0}, w_{jc_1}, \dots, w_{jc_n}\}$ assigned to each model for a given class. The proportion of accurate predictions made by the model M_j for a given class c_n is directly proportional to the weight assigned to that model for that predicted class w_{jc_n}. The calculation of weight is represented by Eq. (3)

$$w_{jc_n} = \frac{\text{Correct Predictions by } M_j \text{ classifying } c_n}{\text{Total Samples of } c_n \text{ in } D_{score}} \tag{3}$$

In order to achieve optimal splitting, a score evaluation stage is established that is similar in nature to a typical FL system training stage. Then, a number of iterations k is evaluated to identify the most suitable partitioning of training and scoring data. For each iteration k the optimal partitioning is selected based on performance.

$$\text{Optimal split} = \arg\max_{k}(f_{score}(D^k_{train}, D^k_{score})) \tag{4}$$

The selection is described by Eq. (4), where f_{score} is the evaluation function that assesses the quality of the split based on classification performance.

The optimum split is then implemented for the final deployment of the WFE-Model model groups. Each model within a group is trained on a specific subset of data, thereby becoming an expert in detecting and classifying the classes represented in its subset. The number of TabPFN models within each model group will be consistent in order to ensure a fair classification between the various subset classes and to optimize the benefits of the weighting voting. Each TabPFN model is trained on a different subset of 1000 samples from the training set of the model group to ensure diversity and reduce overlap between models. Furthermore, the possibility of detecting anomalous and new classes that may not have been previously identified is also considered, with the inclusion of an anomalous class in each model group with the anomalous subset that was designate during the Step 1.

4. Weight ensemble prediction. In the final step, the predictions produced by the various models within the ensemble are aggregated in accordance with the objective of determining the final classification.

To guarantee that the models with greater reliability have a more significant impact on the final decision, each model's prediction is weighted according to its performance on the scoring set. The WFE-Tab approach employs a weighted voting mechanism to leverage the specialized knowledge of each model in the ensemble, thereby facilitating the generation of a more accurate and robust classification.

$$y(x_i) = \arg\max_c \sum_{j=1}^{N} w_j \cdot \mathbb{I}(M_j(x_i) = c) \tag{5}$$

This weighted voting mechanism is represented by Eq. (5), where $M_j(x_i)$ is the prediction of the jth model for instance x_i, w_j is the weight assigned to model M_j, $\mathbb{I}$ is an indicator function that returns 1 if $M_j(x_i) = c$, and 0 otherwise, and N is the total number of models.

Algorithm 1 WFE-Tab Model algorithm

```
Require: Dataset $D$
Apply K-Means++ to divide data into $k$ clusters.
Result: Subsets $D_1, D_2, ..., D_k$
for each subset $D_i$ do
    Introduce a proportion $\alpha \cdot D_j$ from other subsets and a proportion of $\beta \cdot D_a$ from anomalous subset:
    $D_i^{\text{fused}} = D_i \cup (\alpha \cdot D_j) \cup (\beta \cdot D_a)$, where $j \neq i$.
end for
for each subset $D_i^{\text{fused}}$ do
    Split into training and scoring sets:
    $D_i^{\text{fused}} = D_i^{\text{train}} \cup D_i^{\text{score}}$
    Train indicated number of models $M$ on $D_i^{\text{train}}$
end for
for each model $M_j$ do
    Evaluate $M_j$ on the scoring set $D_i^{\text{score}}$
end for
Classification
for each sample instance $x$ do
    for each model $M_j$ do
        Predict class $M_j(x)$
        Apply the model's weight $w_j$
    end for
    Compute final classification $y(x)$ and return it
end for
```

The WFE-Tab classification process is outlined in Algorithm 1. This illustrates how each step, from data division and clustering to model training, weighting, and final ensemble voting, contributes to the overall classification.

3.3. Time complexity of WFE-Tab model

The WFE-Tab model employs a multi-step process comprising clustering, training and ensemble voting, which collectively ensure robust classification results. In this context, N represents the total number of instances in the dataset, while K denotes the number of clusters generated during the data division by the clustering step. The time complexity of this step is $O(N \times K \times d \times i)$, where d is the number of features and i is the number of iterations for convergence, and it is achieved using the K-means++ algorithm.

Once the clustering and fusion processes have been completed, WFE-Tab trains multiple TabPFN models for each subset. We let M_k, C_k, $f(x)$ and $I_{tra-sco}$ denote the TabPFN models of a group k, the size of the fusion subset, the time complexity of the TabPFN training models, and the iterations associated to obtain the optimal split with the scoring split. The complexity of the training time is $O\left(K \times M \times |C_k| \times f(n) \times I_{\text{tra-sco}}\right)$. After training the models, a scoring set is used to evaluate the accuracy of each one, and a weight is assigned to each based on its performance for specific classes. The complexity of the scoring depends on the size of the scoring set S and the number of models M_k per cluster. The total scoring complexity for all clusters is $O\left(I_{tra-sco} \times \sum_{k=1}^{K} M_k \times S\right)$. Finally, after the weights are determined, the models' predictions are combined using a weighted voting mechanism. The complexity of the vote depends on the number of models M_k per cluster and the size of the data P to predict. The total complexity of the vote is $O\left(\sum_{k=1}^{K} M_k \times P\right)$.

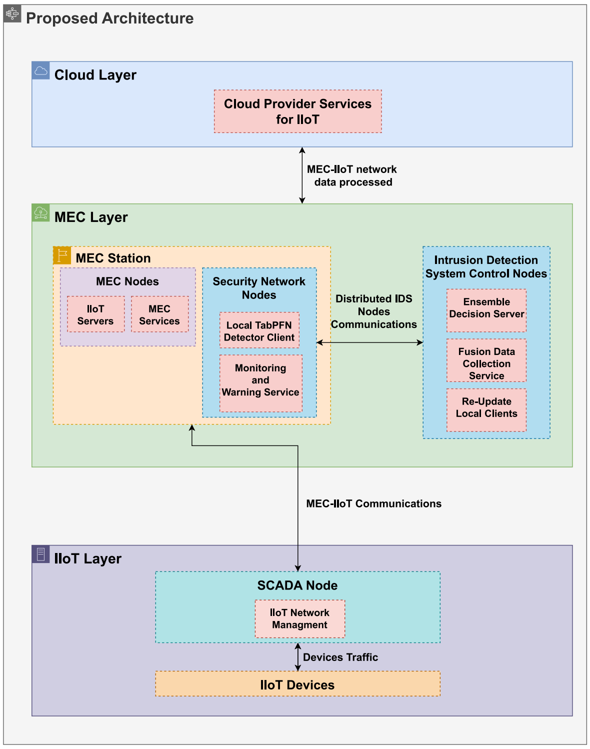

Fig. 2. Proposed IIoT-MEC IDS architecture.

By summing up the time complexities of the clustering, training, scoring, and voting stages, we obtain the time complexity of WFE-Tab represent in Eqs. (6) and (7):

$$O_{training}(N \times K \times d \times i + K \times M \times |C_k| \times I_{\text{tra-sco}} \times f(n) + I_{tra-sco} \times \sum_{k=1}^{K} M_k \times S) \tag{6}$$

$$O_{prediction}(\sum_{k=1}^{K} M_k \times P) \tag{7}$$

3.4. Proposed IDS architecture

Taking into account the benefits offered by MEC, particularly the computational resources that facilitate the implementation of an IDS based on ML models distributed over multiple MEC stations, and considering the characteristics of our WFE-model, we chose to use it as the basis for the proposed IIoT architecture referenced in [53].

This architecture is a three-layer structure that describes the different components of an IIoT-MEC scenario, with the addition of a cloud layer to incorporate potential IoT, Big Data, or storage services. The proposed architecture, depicted in Fig. 2, integrates the MEC stations into the MEC layer that will be present within every IIoT network associated with the manufacturing process.

Two types of nodes are considered within these MEC stations, each with its own functionality. The MEC nodes run various IIoT servers to provide and compute IIoT applications, along with MEC services

Table 2
Distribution of class values in the Edge-IIoTset.

Traffic	Classes	Records	Total
Normal	Normal	11 223 940	11 223 940
Attack	Backdoor	24 862	9 728 708
	DDoS_HTTP	229 022	
	DDoS_ICMP	2 914 354	
	DDoS_TCP	2 020 120	
	DDoS_UDP	3 291 626	
	Fingerprinting	1001	
	Man In The Middle	1229	
	Password	1 053 385	
	Port_Scanning	22 564	
	Ransomware	10 925	
	SQL_injection	51 203	
	Uploading	37 634	
	Vulnerability_scanner	145 869	
	XSS	15 915	

to ensure their correct functionality. Additionally, a Security Network Node is introduced, which implements the Local Model Detector and the Monitoring service. This service sends alarms to alert administrators or activate defensive measures, thereby mitigating potential risks within the topology.

This node establishes communication with the IDS Control nodes, which contain the main functionality of our architecture. The Ensemble Decision propagates and receives the classification of every packet received by the model clients present in the MEC layer, returning the final decision of the classification to the requesting client. The Fusion Data Collection Service is responsible for receiving each packet received by the IDS Control Nodes. It also performs scheduled implementation of preprocessing operations on the data that has been collected and stored. This service is used by the Re-Update Local Clients implementation to prevent data drift in classifications by Local Clients and to add more information and knowledge to these models.

In the IIoT layer, each IIoT device is connected to the Supervisory Control And Data Acquisition (SCADA) node, enabling communication with other IIoT devices or IIoT servers located within the MEC layer. The SCADA node's primary function within our architecture is to identify and determine measures to prevent anomalous behaviour detected in the IIoT network.

4. Threat model

With reference to the IDS architecture outlined in Section 3.4, our local model detector operates as an application within the MEC stations connected to the IIoT networks. The IDS Control nodes are executed on designated machines, which are strategically isolated from the remaining nodes of the MEC stations.

The local model detector receives mirrored traffic from the SCADA of the IIoT topology, undertakes a comprehensive analysis, and initiates a classification process. The Ensemble Decision Server receives the information regarding the classification, which is the result of an ensemble decision among multiple local model detectors located within the MEC layer network. The classification is then returned to the originating local model detector. If a packet is classified as malicious, the monitoring application receives an alert. This application, which is also operational within the MEC station, can initiate defensive countermeasures to prevent potential damage to the IIoT network.

The assumption is that the nodes where the IDS functions are located are free of malware infection. Additionally, it is assumed that during the collection of normal traffic for model training, the devices within the IIoT topology exhibit typical behaviour and are functioning optimally. The model has been trained using the types of attacks that a malicious attacker would typically employ within the IIoT. During the data preprocessing phase, each packet is tagged according to the attack it originates from. Furthermore, the results of the model will not be subject to manipulation by attackers by means of poisoning or tampering attacks.

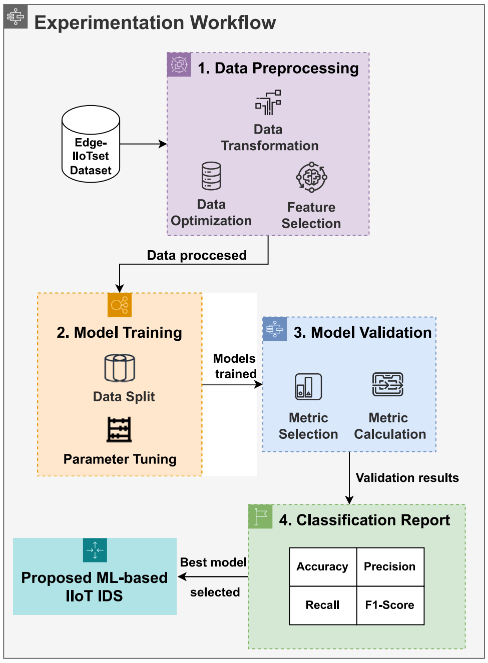

Fig. 3. Workflow of the experimentation.

The behaviour of the attackers is primarily aimed at scanning the IIoT to gather information about the connected devices. The objective of these attacks is to disrupt the normal operation of industrial devices, manipulate the information exchanged between them, halt their service through DoS attacks, gain unauthorized access to the devices' various services, generate blocks in the devices using different encryption methods, install backdoors to control already compromised IIoT devices, and employ various techniques to crack passwords. These attacks primarily target the network layer and the web application layer. It is assumed that the attacker is connected to the IIoT network using a previously compromised device, from which the attacks are launched.

The IDS is not limited to correctly classifying only those attacks it has been trained to detect. It can also provide general anomalous classifications if the attacker executes an unknown attack, and alert administrators with a special alarm. The IDS nodes are centralized in the MEC layer across different network segments and machines. If a disconnection occurs between the two networks, the detector will stop functioning. This is to prevent any potential manipulation of the classification by attackers. If an attacker gains access to the machine running the detector and infects it, they could potentially conceal attacks on the industrial network.

5. Experimentation

The performance of our proposed architecture was tested using the Edge-IIoTset dataset with the experimentation aimed at analysing its performance in comparison with other baseline algorithms which shows a good performance solving tabular data problems [18,54]. The ML algorithms selected for our experimentation are: DT, RF, Xgboost, LightGBM, AdaBoost, GradientBoosting, CatBoost.

This section describes the dataset, the hardware, and the data science process used to measure the performance of each model and obtain the results for the situations considered. The workflow used during this process is illustrated in Fig. 3.

Table 3
Experimentation setup.

Resource	Details
Central Processing Unit (CPU)	Intel i7-13700KF
Random Access Memory (RAM)	32 GB
Graphics Processing Unit (GPU)	NVIDIA RTX 3060 Ti
Operating System	Ubuntu 22.03 LTS
Language	Python 3.10

5.1. Dataset

The Edge-IIoTset dataset [17] collection was carried out by orchestrating a bespoke IIoT-MEC testbed design. This design includes a range of devices, sensors, protocols, and configurations, providing a comprehensive and representative dataset.

This dataset comprises data from over ten distinct types of IoT devices, each designed for specific tasks. These devices include digital sensors for monitoring temperature and humidity, ultrasonic sensors, water level sensors, pH meters, soil moisture sensors, heart rate sensors, and flame sensors, among others.

Various features from different sources such as alarms, system resources, logs and network traffic are used to increase the specificity of the dataset. It is worth noting that the dataset has been enriched by the authors of the dataset with 61 new features, carefully selected from a pool of 1176 features considered during the deployment and running stages of the scenario. This careful analysis has increased the complexity of the dataset, making it a valuable resource for our analysis and modelling process.

The Edge-IIoT set addresses the connectivity challenges of IIoT and MEC protocols, as analysed by an extensive examination of 14 interrelated attacks. These attacks can be broadly categorized into five groups:

- **DoS/DDoS attacks.** This types of attacks aim to disrupt services available to authorized users, either individually or through a distributed approach. Four primary techniques that are commonly used in such situations are considered in this dataset, these being TCP SYN Flood, UDP flood, HTTP flood, and ICMP flood.
- **Information Gathering attacks.** The first step in any effective attack is usually the gathering of information about the target of the attack. This dataset examines port scanning, operating system fingerprinting and vulnerability scanning, three key actions that malicious actors often take during the information gathering phase.
- **Man in the Middle (MitM) attacks.** The aim of this attack is to intercept and control communication between two entities who believe they are directly interacting. The dataset focuses on the use of this attack strategy, targeting protocols that are widely used in almost all modern systems, namely the Domain Name System (DNS) and Address Resolution Protocol (ARP) protocols.
- **Malware attacks.** The impact of such attacks can be considerable, with the potential for significant damage caused by a range of malware types. The dataset focuses on the examination of backdoor, password cracking, and ransomware attacks.
- **Injection attacks.** These attacks aim to compromise the security and confidentiality of the system being targeted. Three methods have been considered for the dataset: Cross-site Scripting (XSS), SQL injection, and upload attacks.

In Table 2, we illustrate the distribution of different traffic types across the attacks and techniques in the dataset.

5.2. Experimentation setup

During the experimental phase of our research, we employed a high-performance workstation that was equipped with the hardware and software specifications detailed in Table 3.

Table 4
Features selected from the EdgeIIoTset dataset by ERT algorithm.

Name	
http.request.version-0.0	icmp.seq_le
http.request.method-0.0	tcp.ack
http.request.version-0	tcp.connection.rst
http.request.method-0	tcp.seq
http.referer-0.0	tcp.len
http.referer-0	http.request.version-HTTP/1.1
tcp.flags	http.request.method-GET
tcp.flags.ack	http.response
tcp.checksum	http.content_length
tcp.ack_raw	tcp.connection.fin
icmp.checksum	dns.qry.name.len
udp.stream	http.request.version-HTTP/1.0
tcp.connection.syn	

5.3. Data preprocessing

The dataset requires adaptation to ensure compatibility with the designated algorithms. It is important to identify techniques suitable for data manipulation and to determine the most relevant features. This process includes preparing data for analysis, which requires the strategic selection of techniques to improve data processing quality. In addition, selecting features carefully can greatly improve the performance of the algorithms, allowing them to better identify patterns and produce accurate results. This phase consists of the following tasks:

Data transformation. To make alphanumeric data compatible with ML algorithms, it needs to be transformed into a binary format. Binarisation techniques, which assign a unique numerical value to each category, are used to perform this transformation. This process ensures that the data is in a format that is suitable for being processed by the ML algorithms. This phase for the CE approach and WFE-Tab is explained in Sections 3.1.3 and 3.2. For this experiment, three subsets of classes were identified and an additional anomalous subset was designated to include samples representing diverse and heterogeneous classes not included in the previous subsets.

Data optimization. After completing data preprocessing, it is important to review data storage and correct any instances of incorrect data types. Unnecessary memory consumption can be avoided and the overall performance of the ML model can be improved by ensuring the correctness of the data types.

Feature selection. Some features may even have a negative impact on the model's performance. Therefore, it is crucial to identify the features that have the most influence for predictive purposes. During our experimentation phase, we used the ERT [27] methodology to select the attributes that provide the most valuable information to the models. This approach enhances the model's predictive accuracy by focusing on the most informative features. The features selected by the algorithm during the experimentation are shown in Table 4.

5.4. Model training

The final set of data is used for the training of the ML models in our study. The training phase is divided into the following tasks:

Data split. Data splitting, which is a crucial step in our research workflow, involves the dataset or subset being split according to the specific ML algorithm used in the experiment. The data has been divided into three distinct phases: training, scoring, and validation. The training phase accounts for 70% of the data, the scoring phase accounts for 10% of the data, which is utilized exclusively for the WFE-Tab to enable weight voting, and the validation phase accounts for the remaining 20% of the data. We used cross-validation to divide the training set into five subsets during the training phase. The evaluation and selection of the optimal model is facilitated by this approach.

Table 5
Hyperparameters selected for the ML algorithms of the study.

Algorithm	Best hyperparameters
DT	criterion:gini, max_depth: 19, min_samples_leaf: 20, min_samples_split: 8
RF	max_depth: 12, n_estimators: 174, criterion: gini, max_features: auto,
Xgboost	learning_rate: 0.1, eta: 0.1, max_depth: 6, subsample: 0.8, seed: 0
LightGBM	learning_rate: 0.0952, max_bin: 20, max_depth: 15, num_leaves: 80, subsample: 0.75
AdaBoost	n_estimators: 526, learning_rate: 0.1
GradientBoosting	max_depth: 13, max_features: auto, min_samples_leaf: 60, min_samples_split: 871, n_estimators: 140, subsample: 0.8
CatBoost	depth: 16, iterations: 30, learning_rate: 0.01

Parameter tuning. The efficacy of the training process can be enhanced by fine-tuning the parameters of the algorithms. In our study, we used Grid Search [55], a widely recognized technique, to systematically traverse and determine the optimal parameters for our ML models in a heuristic fashion, ensuring superior performance and robustness.The hyperparameters found for the ML algorithms considered in our study are detailed in Table 5. It should be noted that for our methodology involving individual TabPFN models, this step is considered unnecessary. However, for the distributed approaches the experimentation is repeated until reaching the optimal ratio of number of clients and performance. Finally, for WFE-Tab score validation stage we established 15 rounds as a way to obtain the best training/score splitting [56].

5.5. Model validation

This validation phase is crucial for evaluating the model's predictive behaviour when classifying IIoT traffic. Several metrics should be considered for a comprehensive evaluation of the model during this stage. All of these metrics use the information contained in the confusion matrix, which includes:

- **True Positives (TP):** In this scenario, TP represents the instances in which the model correctly identifies packets as belonging to a particular type of attack or as belonging to a benign classification.
- **True Negatives (TN):** TN represents the number of instances correctly classified as non-members of a particular attack or category.
- **False Positives (FP):** When dealing with specific types of attacks or benign categories, FP refers to cases where the model incorrectly classifies benign traffic as a specific type of attack, or vice versa.
- **False Negatives (FN):** In this scenario, FN refers to situations in which packets of a particular type of attack are mistakenly labelled as non-attack or benign traffic.

These values align with the classification performed by the model for each packet, and the metrics considered are:

Accuracy. This metric that measures the ratio of correct predictions to the total number of predictions that are generated by the model, and is calculated using the formula outlined in Eq. (8).

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{8}$$

Precision. The precision metric represents the ratio of true positive predictions to the total number of positive predictions made by the model, as shown in Eq. (9). This result highlights the effectiveness of the detector in reducing the number of legitimate packets that are mistakenly identified as malicious.

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

Recall. This measure represents the ratio of true positive predictions to the total number of positive predictions produced by the model, and is calculated via Eq. (10). A higher recall score indicates the effectiveness of the intelligent intrusion detector in accurately identifying a significant number of attacks that match the category it aims to detect.

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

F1-score. The F1 score is a metric commonly used to evaluate models trained on imbalanced datasets. It is calculated as the harmonic mean of precision and recall as shown in Eq. (11), and is particularly useful for models dealing with skewed class distributions.

$$F1\_Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{11}$$

Training-test time. The time in seconds taken by each model to complete the training, validation, and testing stages is used as a metric for measuring the complexity of each method. This metric enables an evaluation of the impact of complexity on performance with respect to the given dataset.

6. Results

The results of our extensive analysis of ML models are presented in this section. Starting with an examination of our model approach and its improved models, the results are structured to reflect the progression of our research. We then highlight the strengths and weaknesses of our approach in comparison with others, and provide a comparative analysis of WEF-Tab with baseline algorithms.

6.1. Comparison between TabPFN approaches

Our analysis begins with an examination of the original TabPFN model, which, despite its strengths, is limited by its capacity to train on a maximum of 1000 samples. Furthermore, both the original model and ensemble approaches encounter difficulties when tasked with classifying more than 10 classes, as has been mentioned in Section 3.1. In the subsequent analyses, we execute each derived TabPFN model to its full potential and compare their performance. This comparison will not only highlight the advances our approach offers over the original model and ensemble methods, but also underscore the potential of our derived models in handling complex classification tasks. As SMV and EMV are ensemble models, their performance has been evaluated using a range of client or model combinations, which consider the number of models used to generate the predictions, up to the maximum number that could be deployed with the available dataset. In the analysis of these approaches, only the number of clients demonstrating the best performance in terms of the metrics used in this study will be considered. Finally, the clustering-based approaches, which are Clustering-based Ensemble TabPFN and WFE-Tab, will be tested with different numbers of clients to see the importance and the impact of increasing this parameter. The results are shown in Table 6.

Table 6
Results of experiments for the different approaches from TabPFN.

Approach	Number of clients	Accuracy	Precision	Recall	F1-Score	Training test time (s)
TabPFN	1	75.52	69.44	70.15	69.79	**4.01**
SMV	89	81.07	81.32	81.40	81.36	104.5
EMV	75	80.81	84.00	81.33	82.64	100.8
CE approach	9	**99.61**	90.58	90.90	90.73	12.1
	12	94.10	95.88	95.63	95.55	15.2
	15	96.20	96.73	90.87	92.18	18.2
WFE-Tab	9	99.20	99.66	99.42	99.53	12.3
	12	99.12	99.62	99.63	99.61	15.5
	15	99.57	**99.81**	**99.82**	**99.81**	18.6

The TabPFN model has an accuracy of 75.52%, a precision of 69.44%, a recall of 70.15%, and an F1 score of 69.79%. These metrics indicate that the model's performance is poor for an IDS. In order to correctly identify and classify network activity, IDSs require high accuracy and precision. TabPFN's relatively low metrics are an indication that it may have a high rate of false positives and negatives, which is undesirable in an IDS.

Although the SMV and EMV models have an accuracy above 80%, they are still not suitable for IDS use. Despite outperforming TabPFN, they still do not provide the required level of precision and recall for effective intrusion detection. The EMV model shows some promise with a higher accuracy of 84.00%, but further improvements will be required before it can be considered a viable solution for use in an IDS.

The CE approach exhibits a notable enhancement in comparison to the previous models, emphasizing a reduction in the number of clients with respect to SMV and EMV, which ultimately leads to an improvement in performance. It achieves the highest accuracy of all the models of 99.61% with 9 clients. However, its precision, recall, and F1-Score, while higher than some other models, are still not the highest. With 12 clients, the CE approach shows consistent performance across all metrics. It achieves an accuracy of 94.10%, a precision of 95.88%, a recall of 95.63% and an F1 score of 95.55%. With 15 clients, the accuracy increases to 96.20%, the precision rises to 96.73%, but the recall drops slightly to 90.87%. This fact makes it possible to see that the CE approach has difficulties in correctly identifying the classes that are not common.

The WFE-Tab model performs better than all the other models and configurations, in particular with 15 clients. It achieves an accuracy of 99.57%, the highest precision (99.81%) and the highest recall (99,82%) and F1 score (99.81%). This suggests that in this comparative analysis, the WFE-Tab model is the most effective model for IDS use, as it provides the most reliable and consistent results.

In the context of integrating a model into the IDS architecture, it is of the utmost importance to consider the balance between the classification of performance and the computational costs. The training test time, as it is often referred to, is a critical factor in determining the efficiency and effectiveness of a model within the system during the phases of training and testing of the different approaches. The Training Test Time metric allows us to determine the number of resources that may require a proposal to run multiple predictions and the time needed for this. This is a crucial element to take into account, as it is not only essential to ensure the model accurately classifies the diverse samples, but also to minimize the time required to provide predictions and to reduce the resources and time needed in the event of future re-training. This factor is of increasing importance in the context of an IDS application.

In view of this, it can be concluded that the WFE-Tab model exhibits an optimum performance training test time ratio. This indicates that the WFE-Tab model is capable of delivering high-quality performance while maintaining a low time cost, rendering it an optimal choice for deployment within the IDS architecture. This finding emphasizes the necessity of considering both performance and cost when selecting a model for deployment. Furthermore, this finding demonstrates the value of the WFE-Tab model in maintaining balance between the two critical factors, thereby enhancing the overall efficiency and effectiveness of the IDS architecture. Fig. 4 shows the confusion matrix of the WFE-Tab and the distribution of the attack classifications.

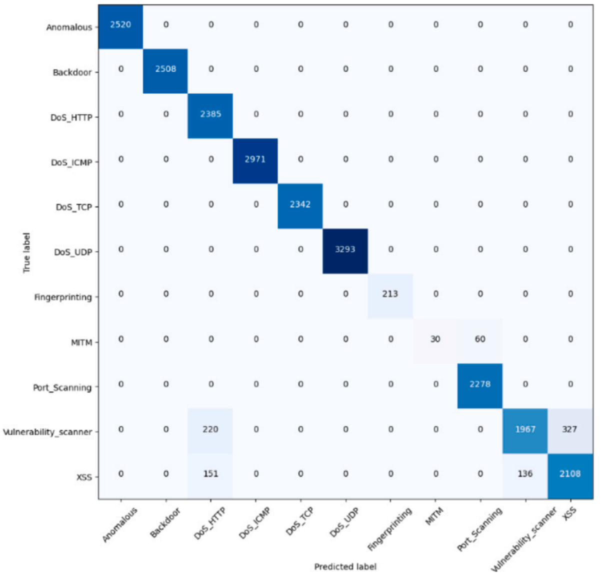

Fig. 4. Confusion matrix of WFE-Tab attack classification.

Table 7
Results of experiments with the baseline algorithms and our approach.

Model	Accuracy	Precision	Recall	F1-Score	Training-test time (s)
DT	96.20	96.74	90.87	92.18	**10.42**
RF	96.48	94.84	93.87	94.28	19.68
Xgboost	96.48	96,31	94.69	95.39	25.16
LightGBM	97.41	96.85	94.96	95.71	20.31
AdaBoost	97.31	95.32	93.37	94.15	21.72
GradientBoosting	96.45	93.76	93.97	93.86	35.14
CatBoost	97.62	97.41	94.99	95.95	20.52
WFE-Tab	**99.57**	**99.81**	**99.82**	**99.81**	18.6

To conclude, the WFE-Tab model with 15 clients shows superior performance across all metrics, making it the most effective model to use when deploying IDS.

6.2. Comparison of WEF-Tab with baseline algorithms

The performance in terms of accuracy, precision, recall and F1 score of the baseline algorithms, which were selected in Section 5, was compared with our approach, the WFE-Tab model using 15 clients. Table 7 summarizes the results of the comparison, and it can be seen that the WFE-Tab outperformed all the other ML models significantly in all the metrics.

Among the other models, CatBoost achieved the second-best performance with an accuracy of 97.62%, a precision of 97.41%, a recall of 94.99% and an F1 score of 95.95%. However, the second-best model is still lacking in accuracy and precision by approximately 2%, and in recall and F1-Score by about 5%, when compared with WFE-Tab. The high degree of precision demonstrates the efficacy of CatBoost in minimizing false positives. This can be attributed to the innovative handling of categorical features and the method employed to combat overfitting through gradient boosting. However, the slightly lower recall rate indicates that, while the model is highly accurate in correctly identifying positive cases, it may potentially overlook a few true positives.

The DT model falls short with a recall of 90.87% and an F1 score of 92.18%, despite its relatively high accuracy of 96.20% and precision of 96.74%. Nevertheless, DT has the potential to overfit in this context, which may account for the reduced recall. The elevated precision indicates that although the model predicts fewer false positives, it may also fail to identify some true positives, resulting in a trade-off in recall.

The RF model shows a balanced performance with an accuracy of 96.48%, a precision of 94.84%, a recall of 93.87% and an F1 score of 94.28%. The Random Forest algorithm produces slightly superior results to those obtained by the Decision Tree. This enhancement can be attributed to the ensemble nature of the Random Forest, whereby multiple decision trees are trained and aggregated to avoid overfitting and enhance generalization. However, it falls short compared to WFE-Tab, suggesting that there is room for improvement in reducing the misclassification of attack packets as normal traffic, which remains a challenge in achieving higher detection accuracy.

The Xgboost model also achieved a high performance with an accuracy of 96.48%, a precision of 96.31%, a recall of 94.69%, and a F1-Score of 95.39%. The model demonstrates a balanced performance across all metrics, but falls short when compared to WFE-Tab. This suggests that XGBoost's strong performance is due to its gradient boosting mechanism that optimizes model performance iteratively. It tends to achieve higher precision by efficiently handling complex data patterns. However, it may not be the optimal choice for this particular task when its compared with CatBoost or WFE-Tab, whose approaches present a better performance classifying and detecting different anomalies in the network.

The LightGBM model performs better than the above models with an accuracy of 97.41%, a precision of 96.85%, a recall of 94.96% and an F1 score of 95.71%, but is still outperformed by the WFE-Tab. The findings suggest that LightGBM's gradient-based one-sided sampling and efficient leaf-wise tree growth enable it to outperform XGBoost in terms of F1-Score. However, it is observed that the model falls short in certain aspects during the classification process when compared to the proposed approach.

The AdaBoost model achieved an accuracy of 97.31%, a precision of 95.32%, a recall of 93.37%, and a F1-Score of 94.15%. Despite its high accuracy, its precision, recall and F1-Score are lower than those of WFE-Tab. This suggests that AdaBoost's mechanism of focusing on difficult samples in each iteration helps it maintain a strong accuracy, though the slightly lower recall indicates it struggles with correctly identifying all true positives. This is likely due to its sensitivity to noisy data.

The model with the lowest performance was Gradient Boosting, which achieved an accuracy of 96.45%, a precision of 93.76%, a recall of 93.97% and an F1 score of 93.86%. Although this model produces competitive outcomes, its slightly inferior F1-score in comparison to alternative boosting techniques may be attributed to overfitting or sensitivity to noise. This highlights the effectiveness of the WFE-Tab, as it outperforms the worst performing model by more than 3% in accuracy and by approximately 6% in precision, recall and F1 score.

The computational time and cost associated with training and testing various models were also evaluated. It was found that the DT model was highly efficient, requiring only 10.42 s to complete the entire process. This efficiency is indicative of the streamlined nature of the DT model, which allows for rapid data processing and decision-making.

The WFE-Tab model, configured with 15 clients, is the next most efficient, requiring 18.6 s for completion. This slightly longer time can be attributed to the increased complexity of the WFE-Tab model, which incorporates additional parameters and computations in its process. The RF model trails closely behind, requiring 19.68 s. The RF model is renowned for its resilience and accuracy. Its training phase involves the generation of numerous decision trees, which contributes to its slightly longer time requirement. However, it is essential to note that the remaining models exhibit a significant increase in computational cost, resulting in training and testing times exceeding 20 s.

Despite differences in computational cost and time, analysis of the performance classification metrics indicates that the WFE-Tab represents the optimal balance between performance and computational resources. This balance between efficiency and effectiveness is crucial in the field of machine learning, where both are key factors in model success.

In conclusion, due to its superior performance in all metrics, WFE-Tab is the best model for implementation in an IDS architecture after a comparative analysis of model algorithm performance.

7. Conclusions and future work

The WFE-Tab model addresses the shortcomings of the existing TabPFN model, which encountered difficulties in dealing with larger classes and training samples. The WFE-Tab model employs a weighted fusion technique to preprocess data into subsets, thereby creating an ensemble of specialized TabPFN models. The IDS is capable of detecting intrusions in IIoT-MEC environments. The WFE-Tab model attained a F1-score of 99.81%. It should be acknowledged that our approach is not without inherent limitations. These include the protection of the model against attackers who may attempt to poison the data used during the training stage, as well as the issue of imbalanced classes with fewer samples, which could present a challenge for the training and scoring stages of multiple models. Secondly, the model has the capacity to identify unknown or anomalous samples during prediction. However, it does not assign these samples to any of the predefined known classes. Moreover, this study has only considered the IIoT-MEC environment, and thus the model has not been tested in other contexts.

To address these limitations, we propose the following enhancements to the proposal and its security:

Algorithm optimization. The objective of our approach is to surpass the intrinsic constraints of the TabPFN model. Nevertheless, the methodology presented in this work could be extended to other baseline algorithms with the aim of improving their performance for different use cases and, as a result, consider possible strategies to address the challenges presented by under-represented classes in the dataset.

Inclusion of blockchain to the IDS architecture. The intrinsic decentralization, transparency and immutability of blockchain could enhance the security of our IDS architecture against potential attacks on our model.

Classification of anomalous samples. We plan to focus on the detailed classification of anomalous samples. This will involve exploring techniques such as semi-supervised learning and refined clustering methods to better differentiate and categorize unknown attacks.

Extended cross-domain study. It would be beneficial to assess the suitability of the WFE-Tab model in a variety of domains in order to evaluate its efficacy in different environments. Even with the potential for adaptations to enhance its functionality for each case.

CRediT authorship contribution statement

Sergio Ruiz-Villafranca: Writing – review & editing, Writing – original draft, Software, Investigation, Formal analysis, Data curation, Conceptualization. **José Roldán-Gómez:** Writing – review & editing, Validation, Supervision, Methodology, Conceptualization. **Javier Carrillo-Mondéjar:** Writing – review & editing, Methodology, Conceptualization. **José Luis Martinez:** Writing – review & editing, Resources, Conceptualization, Project administration. **Carlos H. Gañán:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by the Universidad de Castilla-La Mancha, Spain under the predoctoral contract 2022-PRED-20677 and the project 2023-GRIN-34056, both funded by the European Social Fund Plus (ESF+), Spain, and by the JCCM, Spain under the project SBPLY/21/180501/000195. In addition, this work is part of the R&D project PID2021-123627OB-C52, funded by the MCIN, Spain and the European Regional Development Fund: “a way of making Europe”, and is also funded by MCIN/AEI, Spain/10.13039/501100011033 (FEDER, EU) under the grant PID2022-142332OA-I00. Javier Carrillo-Mondéjar is also supported by MCIN/AEI, Spain/10.13039/501100011033 and European Union NextGenerationEU/PRTR under grant TED2021-131115A-I00, by the Spanish National Cybersecurity Institute (INCIBE), Spain under *Proyectos Estratégicos de Ciberseguridad – CIBERSEGURIDAD EINA UNIZAR*, and by the University, Industry and Innovation Department of the Aragonese Government, Spain under *Programa de Proyectos Estratégicos de Grupos de Investigación* (DisCo research group, ref. T21-23R).

Data availability

The source code used in our experimentation and model implementation is accessible via the following link: https://github.com/C4denaX/WFE-Tab.

References

[1] D. Ivanov, C. Tang, A. Dolgui, D. Battini, A. Das, Researchers’ perspectives on Industry 4.0: multi-disciplinary analysis and opportunities for operations management, Int. J. Prod. Res. (2020) 1–24, http://dx.doi.org/10.1080/00207543.2020.1798035.

[2] P.K.R. Maddikunta, Q.-V. Pham, P. B, N. Deepa, K. Dev, T.R. Gadekallu, R. Ruby, M. Liyanage, Industry 5.0: A survey on enabling technologies and potential applications, J. Ind. Inf. Integr. 26 (2022) 100257, http://dx.doi.org/10.1016/j.jii.2021.100257.

[3] Y. Deng, X. Chen, G. Zhu, Y. Fang, Z. Chen, X. Deng, Actions at the edge: Jointly optimizing the resources in multi-access edge computing, IEEE Wirel. Commun. 29 (2) (2022) 192–198, URL https://ieeexplore.ieee.org/abstract/document/9773060/.

[4] S. Ruiz-Villafranca, J. Carrillo-Mondéjar, J.M. Castelo Gómez, J. Roldán-Gómez, Mecinot: a multi-access edge computing and industrial internet of things emulator for the modelling and study of cybersecurity threats, J. Supercomput. 79 (11) (2023) 11895–11933, http://dx.doi.org/10.1007/s11227-023-05098-2.

[5] T. Gueye, Y. Wang, M. Rehman, R.T. Mushtaq, S. Zahoor, A novel method to detect cyber-attacks in IoT/iIoT devices on the modbus protocol using deep learning, Cluster Comput. 26 (5) (2023) 2947–2973, http://dx.doi.org/10.1007/s10586-023-04028-4.

[6] B. Babayigit, M. Abubaker, Industrial Internet of Things: A review of improvements over traditional SCADA systems for industrial automation, IEEE Syst. J. 18 (1) (2024) 120–133, http://dx.doi.org/10.1109/JSYST.2023.3270620.

[7] W. Xiang, K. Yu, F. Han, L. Fang, D. He, Q.-L. Han, Advanced manufacturing in industry 5.0: A survey of key enabling technologies and future trends, IEEE Trans. Ind. Inform. 20 (2) (2024) 1055–1068, http://dx.doi.org/10.1109/TII.2023.3274224.

[8] K. Zidi, K. Ben Abdellafou, A. Aljuhani, O. Taouali, M.F. Harkat, Novel intrusion detection system based on a downsized kernel method for cybersecurity in smart agriculture, Eng. Appl. Artif. Intell. 133 (2024) 108579, http://dx.doi.org/10.1016/j.engappai.2024.108579, URL https://www.sciencedirect.com/science/article/pii/S0952197624007371.

[9] M. Al-Ambusaidi, Z. Yinjun, Y. Muhammad, A. Yahya, ML-IDS: an efficient ML-enabled intrusion detection system for securing IoT networks and applications, Soft Comput. 28 (2023) 1765–1784.

[10] Unknown, Hybrid deep learning enabled intrusion detection in clustered IIoT, 2023, URL https://www.techscience.com/cmc/v72n2/47259.

[11] F. Conrad, M. Mälzer, M. Schwarzenberger, H. Wiemer, S. Ihlenfeldt, Benchmarking AutoML for regression tasks on small tabular data in materials design, Sci. Rep. 12 (1) (2022) 19350.

[12] N. Hollmann, S. Müller, K. Eggensperger, F. Hutter, TabPFN: A transformer that solves small tabular classification problems in a second, in: The Eleventh International Conference on Learning Representations, 2023, URL https://openreview.net/forum?id=cp5PvcI6w8_.

[13] L. Magadán, J. Roldán-Gómez, J.C. Granda, F.J. Suárez, Early fault classification in rotating machinery with limited data using TabPFN, IEEE Sens. J. 23 (24) (2023) 30960–30970, http://dx.doi.org/10.1109/JSEN.2023.3331100.

[14] M. Nuaimi, L.C. Fourati, B.B. Hamed, Intelligent approaches toward intrusion detection systems for industrial Internet of Things: A systematic comprehensive review, J. Netw. Comput. Appl. 215 (2023) 103637, http://dx.doi.org/10.1016/j.jnca.2023.103637, URL https://www.sciencedirect.com/science/article/pii/S1084804523000565.

[15] T. Sutojo, S. Rustad, M. Akrom, A. Syukur, G.F. Shidik, H.K. Dipojono, A machine learning approach for corrosion small datasets, Npj Mater. Degrad. 7 (1) (2023) 18, http://dx.doi.org/10.1038/s41529-023-00336-7.

[16] S. Ruiz-Villafranca, J. Roldán-Gómez, J.M.C. Gómez, J. Carrillo-Mondéjar, J.L. Martinez, A TabPFN-based intrusion detection system for the industrial internet of things, J. Supercomput. 80 (14) (2024) 20080–20117, http://dx.doi.org/10.1007/s11227-024-06166-x.

[17] M.A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, H. Janicke, Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning, IEEE Access 10 (2022) 40281–40306, http://dx.doi.org/10.1109/ACCESS.2022.3165809.

[18] R. Shwartz-Ziv, A. Armon, Tabular data: Deep learning is not all you need, Inf. Fusion 81 (2022) 84–90, http://dx.doi.org/10.1016/j.inffus.2021.11.011, URL https://www.sciencedirect.com/science/article/pii/S1566253521002360.

[19] M.M. Salim, A.E. Azzaoui, X. Deng, J.H. Park, FL-CTIF: A federated learning based CTI framework based on information fusion for secure IIoT, Inf. Fusion 102 (2024) 102074, http://dx.doi.org/10.1016/j.inffus.2023.102074, URL https://www.sciencedirect.com/science/article/pii/S1566253523003901.

[20] I.A. Khan, I. Razzak, D. Pi, N. Khan, Y. Hussain, B. Li, T. Kousar, Fed-Inforce-Fusion: A federated reinforcement-based fusion model for security and privacy protection of IoMT networks against cyber-attacks, Inf. Fusion 101 (2024) 102002, http://dx.doi.org/10.1016/j.inffus.2023.102002, URL https://www.sciencedirect.com/science/article/pii/S1566253523003184.

[21] M.J. Idrissi, H. Alami, A. El Mahdaouy, A. El Mekki, S. Oualil, Z. Yartaoui, I. Berrada, Fed-ANIDS: Federated learning for anomaly-based network intrusion detection systems, Expert Syst. Appl. 234 (2023) 121000, http://dx.doi.org/10.1016/j.eswa.2023.121000, URL https://www.sciencedirect.com/science/article/pii/S0957417423015026.

[22] J. Li, X. Tong, J. Liu, L. Cheng, An efficient federated learning system for network intrusion detection, IEEE Syst. J. 17 (2) (2023) 2455–2464, http://dx.doi.org/10.1109/JSYST.2023.3236995.

[23] A. Tabassum, A. Erbad, W. Lebda, A. Mohamed, M. Guizani, FEDGAN-IDS: Privacy-preserving IDS using GAN and federated learning, Comput. Commun. 192 (2022) 299–310, http://dx.doi.org/10.1016/j.comcom.2022.06.015, URL https://www.sciencedirect.com/science/article/pii/S0140366422002171.

[24] J.B. Awotunde, S.O. Folorunso, A.L. Imoize, J.O. Odunuga, C.-C. Lee, C.-T. Li, D.-T. Do, An ensemble tree-based model for intrusion detection in industrial internet of things networks, Appl. Sci. 13 (4) (2023) http://dx.doi.org/10.3390/app13042479, URL https://www.mdpi.com/2076-3417/13/4/2479.

[25] T. Chen, C. Guestrin, XGBoost: A scalable tree boosting system, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’16, ACM, New York, NY, USA, 2016, pp. 785–794, http://dx.doi.org/10.1145/2939672.2939785, URL http://doi.acm.org/10.1145/2939672.2939785.

[26] T.K. Ho, Random decision forests, in: Proceedings of 3rd International Conference on Document Analysis and Recognition, 1, 1995, pp. 278–282 vol.1, http://dx.doi.org/10.1109/ICDAR.1995.598994.

[27] P. Geurts, D. Ernst, L. Wehenkel, Extremely randomized trees, Mach. Learn. 63 (1) (2006) 3–42.

[28] R.E. Schapire, Explaining AdaBoost, in: B. Schölkopf, Z. Luo, V. Vovk (Eds.), Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 37–52, http://dx.doi.org/10.1007/978-3-642-41136-6_5.

[29] N. Moustafa, M. Ahmed, S. Ahmed, Data analytics-enabled intrusion detection: Evaluations of ToN_IoT linux datasets, in: 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 2020, pp. 727–735, http://dx.doi.org/10.1109/TrustCom50675.2020.00100.

[30] M. Mohy-eddine, A. Guezzaz, S. Benkirane, M. Azrour, An effective intrusion detection approach based on ensemble learning for IIoT edge computing, J. Comput. Virol. Hacking Tech. 19 (4) (2023) 469–481, http://dx.doi.org/10.1007/s11416-022-00456-9.

[31] J.M. Peterson, J.L. Leevy, T.M. Khoshgoftaar, A review and analysis of the Bot-IoT dataset, in: 2021 IEEE International Conference on Service-Oriented System Engineering, SOSE, 2021, pp. 20–27, http://dx.doi.org/10.1109/SOSE52839.2021.00007.

[32] M. Zolanvari, WUSTL-IIOT-2021 dataset, 2021, http://dx.doi.org/10.21227/yftq-n229.

[33] S. Ruiz-Villafranca, J. Roldán-Gómez, J. Carrillo-Mondéjar, J.M.C. Gómez, J.M. Villalón, A MEC-IIoT intelligent threat detector based on machine learning boosted tree algorithms, Comput. Netw. 233 (2023) 109868, http://dx.doi.org/10.1016/j.comnet.2023.109868, URL https://www.sciencedirect.com/science/article/pii/S1389128623003134.

[34] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T.-Y. Liu, LightGBM: A highly efficient gradient boosting decision tree, in: I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 30, Curran Associates, Inc., 2017, URL https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

[35] L. Prokhorenkova, G. Gusev, A. Vorobev, A.V. Dorogush, A. Gulin, CatBoost: unbiased boosting with categorical features, in: Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS '18, Curran Associates Inc., Red Hook, NY, USA, 2018, pp. 6639–6649.

[36] J.H. Friedman, Greedy function approximation: A gradient boosting machine, Ann. Statist. 29 (5) (2001) 1189–1232, http://dx.doi.org/10.1214/aos/1013203451.

[37] F.S. Melícias, T.F.R. Ribeiro, C. Rabadão, L. Santos, R.L.D.C. Costa, GPT and interpolation-based data augmentation for multiclass intrusion detection in IIoT, IEEE Access 12 (2024) 17945–17965, http://dx.doi.org/10.1109/ACCESS.2024.3360879.

[38] J. Quinlan, Decision trees and decision-making, IEEE Trans. Syst. Man Cybern. 20 (2) (1990) 339–346, http://dx.doi.org/10.1109/21.52545.

[39] M. Ahmad, Q. Riaz, M. Zeeshan, H. Tahir, S. Haider, M. Khan, Intrusion detection in internet of things using supervised machine learning based on application and transport layer features using UNSW-NB15 data-set, EURASIP J. Wireless Commun. Networking 2021 (2021) http://dx.doi.org/10.1186/s13638-021-01893-8.

[40] I. Tareq, B.M. Elbagoury, S. El-Regaily, E.-S.M. El-Horbaty, Analysis of ToN-IoT, UNW-NB15, and Edge-IIoT datasets using DL in cybersecurity for IoT, Appl. Sci. 12 (19) (2022) http://dx.doi.org/10.3390/app12199572, URL https://www.mdpi.com/2076-3417/12/19/9572.

[41] X. Zhang, L. Hao, G. Gui, Y. Wang, B. Adebisi, H. Sari, An automatic and efficient malware traffic classification method for secure Internet of Things, IEEE Internet Things J. 11 (5) (2024) 8448–8458, http://dx.doi.org/10.1109/JIOT.2023.3318290.

[42] S.H. Javed, M.B. Ahmad, M. Asif, W. Akram, K. Mahmood, A.K. Das, S. Shetty, APT adversarial defence mechanism for industrial IoT enabled cyber-physical system, IEEE Access 11 (2023) 74000–74020, http://dx.doi.org/10.1109/ACCESS.2023.3291599.

[43] A. Aleesa, M. Younis, A.A. Mohammed, N. Sahar, Deep-intrusion detection system with enhanced UNSW-NB15 dataset based on deep learning techniques, J. Eng. Sci. Technol. 16 (1) (2021) 711–727.

[44] A. Thakkar, R. Lohiya, Fusion of statistical importance for feature selection in Deep Neural Network-based Intrusion Detection System, Inf. Fusion 90 (2023) 353–363, http://dx.doi.org/10.1016/j.inffus.2022.09.026, URL https://www.sciencedirect.com/science/article/pii/S1566253522001646.

[45] S. Shen, C. Cai, Z. Li, Y. Shen, G. Wu, S. Yu, Deep Q-network-based heuristic intrusion detection against edge-based SIoT zero-day attacks, Appl. Soft Comput. 150 (2024) 111080, http://dx.doi.org/10.1016/j.asoc.2023.111080, URL https://www.sciencedirect.com/science/article/pii/S1568494623010980.

[46] S. Yu, X. Wang, Y. Shen, G. Wu, S. Yu, S. Shen, Novel intrusion detection strategies with optimal hyper parameters for industrial internet of things based on stochastic games and double deep Q-Networks, IEEE Internet Things J. 11 (17) (2024) 29132–29145, http://dx.doi.org/10.1109/JIOT.2024.3406386.

[47] S. Yu, R. Zhai, Y. Shen, G. Wu, H. Zhang, S. Yu, S. Shen, Deep Q-Network-based open-set intrusion detection solution for industrial internet of things, IEEE Internet Things J. 11 (7) (2024) 12536–12550, http://dx.doi.org/10.1109/JIOT.2023.3333903.

[48] S. Shen, C. Cai, Y. Shen, X. Wu, W. Ke, S. Yu, MFGD3QN: Enhancing edge intelligence defense against DDoS with mean-field games and dueling double deep Q-network, IEEE Internet Things J. 11 (13) (2024) 23931–23945, http://dx.doi.org/10.1109/JIOT.2024.3387090.

[49] M. Karabacak, P. Jagtiani, A. Carrasquilla, R.K. Shrivastava, K. Margetis, Advancing personalized prognosis in atypical and anaplastic meningiomas through interpretable machine learning models, J. Neurooncol. 164 (3) (2023) 671–681, http://dx.doi.org/10.1007/s11060-023-04463-8.

[50] A.K. Dasari, S.K. Biswas, D.M. Thounaojam, D. Devi, B. Purkayastha, Ensemble learning techniques and their applications: An overview, in: A. Kumar, S. Mozar, J. Haase (Eds.), Advances in Cognitive Science and Communications, Springer Nature Singapore, Singapore, 2023, pp. 897–912.

[51] F. Jemili, R. Meddeb, O. Korbaa, Intrusion detection based on ensemble learning for big data classification, Cluster Comput. 27 (3) (2024) 3771–3798, http://dx.doi.org/10.1007/s10586-023-04168-7.

[52] R. Mussabayev, N. Mladenovic, B. Jarboui, R. Mussabayev, How to use K-means for big data clustering? Pattern Recognit. 137 (2023) 109269, http://dx.doi.org/10.1016/j.patcog.2022.109269, URL https://www.sciencedirect.com/science/article/pii/S0031320322007488.

[53] Z. Ai, W. Zhang, M. Li, P. Li, L. Shi, A smart collaborative framework for dynamic multi-task offloading in IIoT-MEC networks, Peer- To- Peer Netw. Appl. 16 (2) (2023) 749–764.

[54] Y. Gorishniy, I. Rubachev, V. Khrulkov, A. Babenko, Revisiting deep learning models for tabular data, in: M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, J.W. Vaughan (Eds.), Advances in Neural Information Processing Systems, Vol. 34 (2021) 18932–18943, URL https://proceedings.neurips.cc/paper_files/paper/2021/file/9d86d83f925f2149e9edb0ac3b49229c-Paper.pdf.

[55] S.M. LaValle, M.S. Branicky, On the relationship between classical grid search and probabilistic roadmaps, in: J.-D. Boissonnat, J. Burdick, K. Goldberg, S. Hutchinson (Eds.), Algorithmic Foundations of Robotics V, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 59–75, http://dx.doi.org/10.1007/978-3-540-45058-0_5.

[56] G. Afendras, M. Markatou, Optimality of training/test size and resampling effectiveness in cross-validation, J. Statist. Plann. Inference 199 (2019) 286–301, http://dx.doi.org/10.1016/j.jspi.2018.07.005, URL https://www.sciencedirect.com/science/article/pii/S0378375818301514.