CGAPP: A Continuous Group Authentication Privacy-Preserving Platform for Industrial Scene

Juan Manuel Espín López[a,*], Alberto Huertas Celdrán[b], Francisco Esquembre[c], Gregorio Martínez Pérez[a], Javier G. Marín-Blázquez[a]

[a]*Department of Information and Communications Engineering (DIIC), University of Murcia, Murcia, 30100, Murcia, Spain*
[b]*Communication Systems Group (CSG), Department of Informatics (IfI), University of Zurich UZH, Zurich, CH-8050, Zurich, Switzerland*
[c]*Department of Mathematics, University of Murcia, Murcia, 30100, Murcia, Spain*

Abstract

In Industry 4.0, security begins with the workers' authentication, which can be done individually or in groups. Recently, group authentication is gaining momentum, allowing users to authenticate as group members without the need to specify the particular individual. Continuous authentication and federated learning are promising techniques that might help group authentication by providing privacy, by its own design, and extra security compared to traditional methods based on passwords, tokens, or biometrics. However, these techniques have not previously been combined or evaluated for authenticating workers in Industry 4.0. Thus, this paper proposes a novel continuous group authentication privacy-preserving (CGAPP) platform that is suitable for the industry. The CGAPP platform incorporates statistical data from workers' smartphones and employs federated learning-based outlier detection for group worker authentication while ensuring the privacy of personal data vectors. A series of experiments were performed to measure the framework's suitability and address the following research questions: i) What's the cost of using FL compared to full data access in industrial scenarios? ii) How robust is federated learning against adversarial attacks, specifically, how much malicious data is required to deceive the model? and iii) How much noise is required to disrupt the authentication system? The results demonstrate the effectiveness of the CGAPP platform in the industry since it provides factory safety while preserving privacy. This platform achieves an accuracy of 92%, comparable to the 96% obtained by traditional approaches in the literature that do not address privacy concerns. The platform's robustness is tested against attacks in the second and third experiments, and various countermeasures are evaluated. While the CGAPP platform exhibits certain vulnerabilities to data injection attacks, straightforward countermeasures can alleviate them. Nevertheless, the system's performance experiences a notable impact in the event of a data perturbation attack, and the countermeasures investigated are ineffective in addressing this issue.

Keywords: Continuous Authentication, Group Authentication, Federated Learning, Adversarial Attack, Industry 4.0

1. Introduction

Production lines in today's Industry 4.0 are automated and controlled by electronic devices, which are, in turn, operated or programmed by highly trained employees. Since processes are geared to maximize production, any incorrect or mischievous operation can cause significant time and/or money losses and seriously affect production. This makes security a central issue in modern factories [1]. Security begins with authentication from dedicated devices [2] of workers who operate the production machines. Allowing a worker to perform a task that is not under his/her prescribed duties or for which he or she has yet to be trained can pose a significant risk of failure. Even worse, allowing access to the factory or its systems of unauthorized persons or attackers (saboteurs) is an ever-present severe security risk.

Authentication is a broad and well-studied topic that still poses open research questions. Although authentication is considering the use of some futuristic techniques, such as DNA sampling, the most commonly used mechanisms in industries for user authentication are identification cards, passwords, and biometrics [3]. Identification cards allow quick access to premises through doors or turnstiles, but they can be lost, stolen, cloned, or exchanged among workers. Passwords can provide moderately fast access to the factory and computer systems, but they can also be stolen or exchanged (and frequently forgotten). Finally, biometric systems, like facial or fingerprint, are high-speed access mechanisms that are more difficult to impersonate, and definitely not forgettable. Still, their use requires dedicated sensors, which can sometimes interfere with the personal protective equipment worn by workers, such as glasses or face masks, making their use unpractical or even impossible in some situations. Therefore, a more passive, non-intrusive,

[*] Corresponding author.

This research was partially funded by the Spanish Ministry of Science and Innovation, State Research Agency, FEDERfunds, through projects RTI2018-095855-B-I00, MTM2017-84079-P and PID2019-108377RB-C32, by the Swiss Federal Office for Defense Procurement (armasuisse) with the Cyber-Tracer (CYD-C-2020003) project, and by the University of Zürich UZH.

Email addresses: juanmanuel.espin1@um.es (Juan Manuel Espín López), huertas@ifi.uzh.ch (Alberto Huertas Celdrán), fem@um.es (Francisco Esquembre), gregorio@um.es (Gregorio Martínez Pérez), jgmarin@um.es (Javier G. Marín-Blázquez)

and non-equipment-restrictive authentication method is necessary for these situations.

In many factories, workers operate the production line using electronic devices. These devices can continuously collect information of the worker's behavior derived from generic sensors or usage statistics. The acquisition of all this data over time enables the utilization of continuous authentication (CA) systems [4] for worker verification and intrusion detection. A CA system, like any authentication systems, requires an initial data acquisition phase (enrollment) to subsequently authenticate users. During this period, which in this study lasted 14 days, workers must be authenticated through alternative methods specified by the company. Once enrollment is complete, CA systems enhance the security of the factory by continuously authenticating users [5], not just on an occasional basis. If an intrusion is detected, the CA system has the capability to report it to the management, block the device, or even, when feasible, undo the last actions, effectively preventing potential security breaches or production line failures.

Most conventional user authentication schemes are based on individual or one-to-one authentication, where the system checks if the device is operated by a particular legitimate user . In general, to obtain effective models for individual authentication, the system would require access to the personal and private data of these individuals. As individual users become increasingly concerned and assertive about their privacy rights, even in their work environments, there is a growing interest in authentication systems that prioritize the protection of these rights. In certain industry scenarios, group-oriented tasks exist where security concerns do not really necessitate individual identification, and group membership alone suffices. A suitable scheme for such group-oriented applications is group authentication [6], wherein the system verifies the user's affiliation with a specific group. To fully respect user privacy, these systems must allow for the indication of such group membership while being trained in a manner that guarantees the protection of user's private personal data. Members of the group must have identical privileges, such as access to locations, responsibilities, tasks and more. Group authentication enhances the privacy of users by solely confirming their membership within the group without disclosing private personal data.

Therefore, combining continuous and group-oriented approaches makes continuous group authentication (CGA) systems an ideal solution for enhancing security in industrial environments while also safeguarding the personal data privacy of individual workers. A CGA system can be designed for different workers' devices, such as smartphones, laptops, tablets or PDAs. The CGA system generates a model containing all users' behavior, and use this collective information to authenticate each user as a group member. Unfortunately, as previously mentioned, to develop such a model many learning algorithms require extracting personal user data from the devices and sending this private data to some repository or server, which may conflict with personal data users' privacy. However, in recent years, a new paradigm for training machine learning models, called Federated Learning (FL) [7] has emerged, which eliminates the need for sending private data. FL methods allow for building collective models of different participants without sharing their data. Instead, to train a model, each federated participant shares training weights or gradients. These parameters are then merged into a single global model and distributed back to each participant. After several rounds of this process, a final model trained with all clients' knowledge is obtained. Therefore, this training paradigm solves the issue of data leakage and data privacy in some algorithms used to create CGA systems, as user private data no longer need to leave the device to build the model.

This solution, which provides privacy of workers' personal data, can suffer different attacks on the server and the clients, [8]. Since the company provides the server, it is considered honest and reliable, and therefore adversarial attacks can only be carried out on the client side. There are a variety of attacks, some more complex and some more straightforward. The most worrisome attacks can be carried out without technical knowledge of the application, such as injection or perturbation during data collection or enrollment. In an injection attack, a malicious user can use a legitimate worker's device to introduce fabricated or impersonated behavioral data. Conversely, in a data perturbation attack, a compromised worker intentionally alters his behavior to induce a system failure.

In particular, this work focuses on an application scenario for Industry 4.0 that uses continuous group authentication trained in an unsupervised way (it only trains with users' data). This specific scenario raises some interesting research questions that need to be addressed:

1. In an industrial scenario, what is the cost in the accuracy of using systems that learn with this increased data privacy protection, such as FL, versus systems that require full data access in training and operation?
2. How robust is the federated approach when confronted with adversarial attacks? In particular, how much data from an unauthorized worker must be maliciously injected to make the model consider a member of the group?
3. If a group worker becomes compromised or attempts to disrupt the authentication process, what degree of data perturbation should the worker introduce? How much noise is required to compromise the integrity of the system?

To address these questions, with the aforementioned application scenario in mind:

- A new continuous group authentication platform, the CGAPP platform, has been designed and developed. This new platform is aimed to provide industrial companies with a CGA scheme for smartphones using an outlier detection approach. The code for the CGAPP platform server is available in [9].

- The validation of the CGAPP platform took place within an industry-centered scenario where workers utilized smartphones as their work devices. To facilitate this validation, an existing public dataset was employed in this study. The users in the dataset have been first analyzed to find a group of workers sharing enough characteristics to be considered a workforce.

- To answer the first research question, the first experiment aimed to validate the CGAPP platform and evaluate its performance in terms of security and privacy.

- Finally, the two last experiments address the second and third research questions by evaluating the system robustness against two types of adversarial attacks: an injection, in which an impostor tries to inject data into a worker's record, and a poisoning attack, in which some workers deliberately send corrupted data.

The paper is structured as follows. Section 2 analyses related work. Section 3 describes the new platform, a CGA trained with FL. Section 4 reviews the adversarial attacks considered and possible countermeasures. Section 5 describes the scenario and the results obtained from the different experiments to discuss the results finally. To end, Section 6 draws conclusions and sketches possible future work.

2. Related Work

CGA using machine learning while maintaining data privacy is a novel research topic that has not been previously studied. Therefore, a general state-of-the-art review of the different constituent techniques applied in this work is carried out below. A summary of the related works can be found in Table 1.

Group authentication (GA) is specially designed for group-oriented applications, in which it is unnecessary to know the particular identity of users [6]. It is quite common to find works using group authentication in the fields of Network Communications [10], Internet of Things (IoT) [11], and Internet of Vehicles (IoV) [12]. A comprehensive review of recent work can be found in [13]. Again, group authentication applied to people authentication has not been thoroughly studied. In [14], Shaoning et al. present a method for authenticating an individual's membership to a dynamic group without revealing the individual's identity and without restricting the group size or the members of the group. These authors use a facial recognition system with an SVM classifier and a private dataset with 1355 face images of 271 people (5 face images per person). The proposed system achieves a success rate of over 96% for different group sizes, from 10 to 40 users.

Since CGA has not being previously studied in depth, some of the most relevant and recent works in continuous authentication in smartphones are detailed below. If grouped by the source of information used, it is possible to find various works that use device sensors, such as accelerometers, gyroscopes, or magnetometers. In [15], Li et al. use a two-stream Convolutional Neural Network to extract the top 25 features and a one-class support vector machine as a classifier. In experimental results using a private dataset and the BrainRun Dataset, their Scanet system achieves a 90.04% accuracy and a 5.14% equal error rate. In [16], Li et al. propose the CAGANet system, which includes a generative adversarial network to do data augmentation and use four different one-class classifiers. The CAGANet system achieved, with the Isolation Forest classifier, the lowest equal error rate (EER) of 3.64%. Other sources have also been used, such as touchscreen data. Shuwandy et al., in [17] propose the BAWS3TS system, which uses a template-matching algorithm. Their test was conducted with three adult volunteers, and the system presented a high accuracy rate of 98%.

The increasing availability of many data sources on the same device, combined with its growing computing power to process them, enables the combination of two or more of these data sources to create a more robust solution. The most typical combination involves all or some of the previously mentioned sensors along with a new, distinct source. Sensors, statistics, and voice are used in [18], where the authors evaluate, in an unsupervised way, different algorithms using the S3-Dataset, achieving more than 90% accuracy when all the sources are available. The same authors apply in [5] continuous authentication for the Industry 4.0 following a supervised approach. Using sensors and statistics, Jorquera et al. in [19] present an Intelligent and Adaptive Continuous Authentication System that it is validated in a online app bank . Also using sensors and statistics, Sánchez et al. present in [20] a multi-device platform for continuous authentication using XGBoost. The system presents up to an 89,35% improvement in the FPR compared to the single-device approach. Sensors and touch screen data are used in [21] to present the DAKOTA system to authenticate users in a banking app. The Dakota system uses a SVM model and achieves an 11.5% equal error rate in a private dataset of 30 users. More work on continuous authentication can be found in the recent survey in [22].

Regarding privacy-preserving, in biometric authentication methods such as facial or speaker recognition, developers typically focus on protecting the pattern that is extracted from the biometric sample with its pattern extractor algorithm. Two recent examples are [23, 24], where the authors encapsulate the pattern as soon as it is extracted and leaves the device. In [25], the authors propose a novel user active authentication training, Federated Active Authentication (FAA), that utilizes the principles of FL and Split Learning to fine-tune a pattern extractor with the users' data. More details on privacy-preserving techniques can be found at [26]. All these options imply the existence of a pattern extractor previously trained with a vast database. Currently, these massive databases do not exist in continuous authentication, leaving these techniques inapplicable for the time being.

As seen in this section, the list of works dealing with CGA systems trained explicitly in a federated way is somewhat limited. The rest of the related works that share some characteristics do not address the problems considered in this paper, either. For this reason, this paper proposes a new platform where workers are authenticated continuously in a group authentication scheme. The system proposed below uses only statistical information but can be adapted to other available device sources with minor changes.

3. CGAPP Platform

This section describes the design and implementation details of the CGAPP platform. The platform task is to continuously authenticate a group of industry workers while preserving their

Table 1: Comparison of related works

Ref.	Continuous Auth.	Group Auth.	Privacy Preserving	Industry
[5] (2022)	✓	x	x	✓
[10] (2015)	x	✓	x	x
[11] (2020)	x	✓	x	x
[12] (2022)	x	✓	x	x
[14] (2003)	x	✓	x	x
[15] (2020)	✓	x	x	x
[16] (2021)	✓	x	x	x
[17] (2022)	✓	x	x	x
[18] (2021)	✓	x	x	x
[19] (2018)	✓	x	x	x
[20] (2020)	✓	x	x	x
[21] (2020)	✓	x	x	x
[23] (2020)	x	x	✓	x
[24] (2021)	x	x	✓	x
[25] (2021)	x	x	✓	x
This work	✓	✓	✓	✓

data privacy. To do this, the CGAPP platform generates an outlier detection model that contains the workers' behavior, trained following a federated approach to preserve data privacy. The platform is designed to use smartphones as workers' work devices. For the use of other different devices, such as laptops or tablets, it could be easily extended by slightly varying some of the modules that compose it.

3.1. Architecture

The CGAPP platform client-server architecture consists of an application client running on the workers' devices and a central server. Each client application collects its worker's data and processes it in order to send limited information to the server. The server then builds the group authentication model from these parameters and distributes it to all connected application clients. After receiving the model, the applications are in charge of authenticating the workers. All communications between server and client are end-to-end encrypted to ensure communication security. The general details of the application and the server are specified as follows. A schematic description of the main functionalities is displayed in Figure 1.

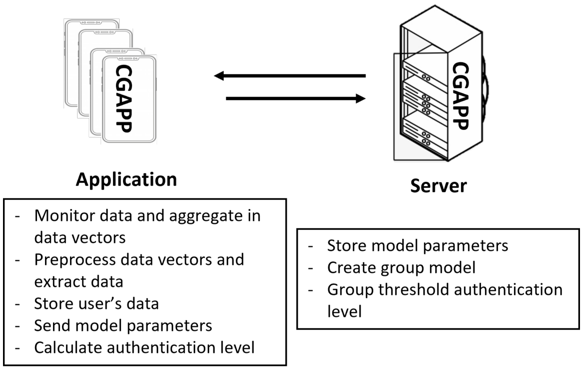
CGAPP
CGAPP
Application
Server
- Monitor data and aggregate in data vectors
- Preprocess data vectors and extract data
- Store user's data
- Send model parameters
- Calculate authentication level
- Store model parameters
- Create group model
- Group threshold authentication level

Figure 1: Main parts and functionalities of the CGAPP platform

3.1.1. Client Application

The client application creates a training dataset by monitoring the worker's data, trains a local machine learning model, and sends the local model parameters to the server. The server then aggregates the models from all group members, and sends the updated group model back to the client application. This process repeats for multiple rounds as needed, and when the training is complete, the application receives the final group aggregated model. Finally, the application monitors the worker, and evaluates the data against the model to authenticate him/her as a group member. To this end, the modules that make up this application are (see also Figure 2):

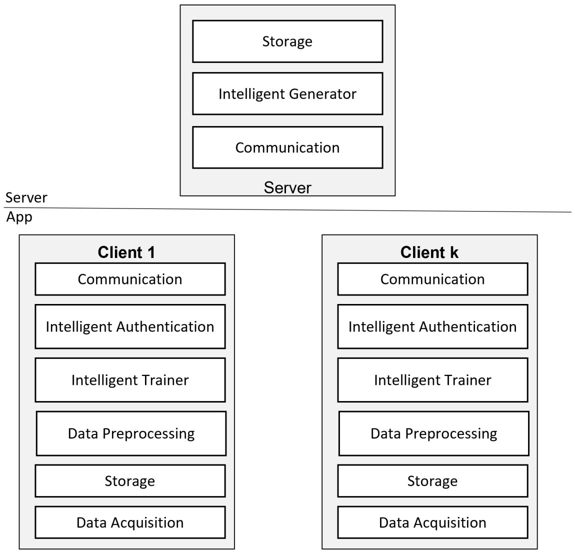

Figure 2: Main parts and functionalities of the CGAPP platform

- **Data Acquisition** This module is responsible for acquiring workers' behavioral data when he/she interacts with industrial smartphones. The data monitored are detailed in Subsection 3.3.1

- **Storage**. This module creates and maintains the training dataset for each worker. Data vectors never leave the device.

- **Data Preprocessing**. This module pre-processes the data (e.g., applying proper scaling).

- **Intelligent Trainer**. This module is in charge of training the model locally.

- **Communication**. This module sends and receives the encrypted information from the server.
- **Intelligent Authentication**. This module checks new sample data against the generated group model to authenticate the worker.

3.1.2. Server

The server acts as a central entity responsible for facilitating communication among all clients within the system. Its primary role involves aggregating a variety of information to construct the behavioral model for the group. To create the group behavioral model, the server receives only the minimum necessary information from each client application. Once the model is constructed, the server distributes the group model back to each client application. In addition to receiving weights and gradients for the federation, the server also acquires other pertinent information, such as preprocessing parameters or thresholds, as dictated by the implementation details. Figure 2 shows the three modules of which the server is composed:

- **Communication**. This module receives the model parameters from each client application and sends them back the aggregated model. It also receives and sends the preprocessing and threshold values.
- **Intelligent Generator**. This module generates the federated group authentication model from the received model parameters.
- **Storage**. This module stores the model parameters of each worker. It also contains other information received and any other necessary for the correct functioning of the server.

3.2. Federated Learning

Federated Learning paradigms are used to train the machine learning (ML) and deep learning (DL) models required by the CGAPP platform. Model parameters/weights are calculated locally during training in the device by the client application, using the worker's private data vectors that are only present in that device . It is important to note that these private data vectorsremain confined to the device, thereby preserving privacy. These model weights are then sent to the server to be aggregated with those from other clients, thus building a single global model for all the clients. The model is then returned to the clients to repeat the training process for a few rounds. Finally, when a given number of iterations is reached, the final model is sent to the clients, ready for group authentication.

3.3. Implementation Details of the Modules

For the sake of reproducibility, the next subsections follow the process flow, providing details about each module implementation and the subsequent experiment development.

3.3.1. Data Acquisition

The Data Acquisition module is in charge of monitoring the smartphone events in each of the selected dimensions at time intervals or whenever they are produced. In this work, app usage statistics is the only dimension used because other data sources are often unavailable. (For example, in an industrial scene, the device is frequently attached to the machine, or the voice or facial are unavailable if the worker uses personal protective equipment.) The design and most configuration values of the data acquisition are based on results from previous work [20, 19].

The data vectors contain information about the different applications used by the workers in the last 60 seconds. Each vector of statistics is then calculated every 60 seconds and contains:

- Foreground application counters (number of different and total apps) for the last minute and the last day.
- Most common app ID and the number of usages in the last minute and the last day.
- ID of the currently active app.
- ID of the last active app prior to the current one.
- ID of the application most frequently utilized prior to the current application.
- Bytes transmitted and received through the network interfaces.

It was decided to use the existing S3 Dataset [27] to address the open research questions stated in Section 1. The S3 Dataset contains the behavior statistics of applications of 21 volunteers interacting with their smartphones for more than 60 days. (The dataset also contains sensors and voice data, which are not interesting for this work.) The type of users is diverse: males and females aged 18 to 70 were included in the dataset generation. The wide range of age is a crucial aspect due to the impact of age on smartphone usage.

The data in this dataset can be extrapolated to the situation considered in this work because the dataset shares important similarities with it: the type of device it uses (smartphone) can be provided by the company as a work assistant and authentication device, the dataset contains statistics app usage data – which are always available–. It considers regular use of the device. (This dataset would be even more suitable if it contained different user devices, the users shared the device, the number of users was more significant, or if it contained adversarial data attacks.)

For completeness, the portions of the database used in this work are detailed below, as well as an abbreviated explanation of how the information contained in the statistics vector has been calculated. More specific details can be found in [27].

3.3.2. Data Preprocessing

Once the data have been captured and stored, a simple preprocessing is applied to the data vectors before model training. Specifically, Min-Max scaling, i.e., $x' = \frac{x - x_{min}}{x_{max} - x_{min}}$ is applied to each of the elements. To apply this scaling it is necessary to calculate the x_{min} and x_{max} values. In earlier stages of this research, the performance of calculating them individually for each worker or globally was evaluated. The results showed that a global calculation of these values gives better results than an individual one. Therefore, this module is in charge of calculating these values x^k_{min} and x^k_{max} for each client k and, using the Communication module, of sending them to the server. The server will then calculate the global ones, x^G_{min} and x^G_{max}, and send them back to each client. The min-max values are computed locally with the worker's data, and these min and max values, which contain worker-specific indirect data, are shared with the server during this stage.

3.3.3. Intelligent Trainer

This module's primary function is to train, with the workers' data, a round of the Federated Learning model desired for group authentication. The training begins once the required worker's data has been acquired and preprocessed. The trigger to start the training can be either having collected a fixed number of vectors, or the end of an initial time period previously configured for data capturing only. For this research, a time period of 14 days of device use was set (see Section 5.1). The module trains an initial model sent by the server via the Communication module.

Three types of ML models were finally chosen among those suitable for a federated approach, and these were chosen among those in widespread use for outlier detection to ease comparison. Other types were also pre-evaluated during the preparatory work of this research, but Autoencoders (A), Variational Autoencoders (VA), and k-nearest neighbor (KNN) presented the best results in that preliminary exploration and are the ones fully researched and reported here. In the same way, the hyperparameters (that is, the parameters of the learning algorithms and the structural parameters that are fixed in the learning) presented below were obtained during that preparatory work:

- **Autoencoder**. Two 1-hidden-layer models, with 16 and 8 neurons each, and four 2-hidden-layers models with 16-8, 18-4, 8-4, and 8-2 neurons.
- **Variational Autoencoder**. Two 1-hidden-layer models, with 16 and 8 neurons each, and four 2-hidden-layers models with 16-8, 18-4, 8-4, and 8-2 neurons.
- **k-nearest neighbor**. With 1, 2, 5, 10, 20, 30, and 50 clusters.

The Variational and Autoencoder models use RELU in each hidden layer and sigmoid as activation function in the output. No Batch Normalization or other type of regulation or dropout was used in any layer, and the training was performed using the stochastic gradient descent (SGD) algorithm with a fixed learning rate fixed of 0.01.

Note that Autoencoders and KNN use slightly different methods to calculate the score (sc). An encoder and a decoder form an Autoencoder. The encoder transforms the input by reducing the number of dimensions to a coding dimension, and the decoder attempts to map the encoded output back to the original input. The training aims to reduce the Mean Squared Error (MSE) between the input and the reconstructed features. In an authentication scene, an Autoencoder is trained with the worker data. Given a new sample, a low reconstruction error indicates that the sample is from a genuine worker, while a high reconstruction error indicates that the sample belongs to an impostor. For Autoencoders, the MSE of a sample is considered the authentication score:

$$\mathrm{sc}(x|A) = \mathrm{MSE}(x, x') \tag{1}$$

where x' is the Autoencoder's reconstruction of x.

On the other hand, the KNN model groups all the training samples into clusters to represent the workers' behavior. Therefore, the distance of one sample to the different clusters indicates the degree of membership of that sample to the workers and thus gives an authentication score. Thus, a low score indicates that the new sample belongs to one of the clusters of the KNN model and, thus, to the worker. Conversely, a high score indicates that it belongs to an impostor.

$$\mathrm{sc}(x|KNN) = \mathrm{dist}(x; KNN) \tag{2}$$

Finally, the threshold must be calculated. In this work, this calculation used the training data of all workers. No data set aside from the training set was used. The formula used was:

$$thr = mean(sc(x)) + std(sc(x)). \tag{3}$$

Note that this is a federated approach, which means that neither the server nor any client application has full access to all the training data from all workers to calculate the threshold. Each client has access only to its private worker data. Therefore, each client application calculates the mean and standard deviation of the samples of that device and sends it to the server along with the number of samples used for its calculation. These values, are not considered private data, and sharing them with the server does not compromise user privacy. Once the server receives all this information, it calculates the global threshold using (3).

3.3.4. Intelligent Generator

This server module is in charge of aggregating the weights sent by each client application and obtaining a global model that includes the behavior of all workers. The Weighted FedAvg Algorithm [7] was selected because each client application may use a different amount of vector data in the training round. Therefore it makes sense to counterbalance it. Once the global model has been built, it is passed to the server's communication module to be sent to all the client applications.

3.3.5. *Intelligent Authentication*

Once the group model is trained and ready, the client application starts the Intelligent Authentication module. During operation, the Data Acquisition module gets a new sample, the Pre-processing module prepares it, and the Intelligent Authentication module uses it to perform the group authentication. A sample is authenticated to belong to the group if the score obtained (when applying the group model to it) reaches or exceeds a particular threshold value (which must be properly set).

4. Adversarial Attacks and Countermeasures

Even when privacy is an issue of great concern, security should not be neglected. Federated Learning models can be the target of a particular type of attack that must be considered: Adversarial Attacks. This section details some of the different adversarial attacks that can be addressed in a CGA scene with a federated model: how they operate and how to avoid or prevent them.

4.1. *Adversarial Attacks*

In FL, for its proper functioning, it is assumed that all participants, clients, and server are honest and therefore do not aim to harm the common good. Unfortunately, this is not always the case, and both clients and server could be malicious. The effect of one of these parties becoming malicious implies a degradation in the functioning, harming the whole system. Since the server is a service controlled by the company's security department, this work does not consider its malicious potential and will be considered completely honest. On the other hand, the security department cannot guarantee that the application client is always behaving honestly. In a group authentication, there may be a compromised worker who wants to carry out an attack, either by corrupting the system or by allowing an attacker to gain access. The study of attacks and countermeasures, therefore, will be focused on the client application part of the CGAPP platform.

The list of attacks is classified into different groups according to where the attack occurs [8]. The focus of this work is on *Data poisoning*, a type of attack in which a client intentionally introduces malicious data to the system, causing a behavioral change in the models indirectly through the data used to train them. These attacks can be carried out through the worker's device or by accessing the device's code or hardware. Among the various types of data poisoning attacks, two prominent ones are:

- **Data Injection**. This attack occurs when a compromised worker aims to impersonate an outside worker as a group member. To achieve this, the compromised client sends the behavioral data of the impostor to the federated model to authenticate the fake worker. Although the global model remains fully functional, it now erroneously authenticates an outside worker as a member of the group.

- **Data Perturbation**. Unlike the previous attack, this one aims to render the model useless by sabotaging it. To achieve this, one or more compromised workers send contaminated data to confuse the model and make it non-functional. As a result, the model either authenticates anyone who tires to access it or, to reject all workers altogether.

4.2. *Countermeasures*

To mitigate the adversarial attacks mentioned above, slight modifications can be made to the aggregation function used by the FedAvg algorithm, which typically uses the average weights to build the overall model. Among the available possibilities [28], this work evaluates two alternative functions: median and clipping [29]. The reasons for selecting these functions were the following: i) they are the simplest functions and require the least amount of computation, which in an industrial environment is appreciated. ii) these functions are robust to the distribution of the data, which is useful because each FL participant is only trained with the data of one worker, and iii) they are robust even when the number of FL participants is small. The aggregation methods compared are then:

- **Mean**. This is the usual aggregation function, where the mean (or average) of the weights is used. This function does not resolve adversarial attacks but is the baseline for comparison with the other functions.

- **Median**. Using this aggregation function, the mean of the weights is replaced by the median to exclude outliers effectively.

- **Clipping values (X^{th} percentile)**. In this case, the 60^{th} and 80^{th} percentiles are calculated for each of the weights and removed. Once the extreme values are removed, the mean is used.

5. Experiments

This section first details the scenario and the metrics used in the experiments for attack detection. Then, the results of the three experiments are explained in detail. The experiments conducted are: 1) evaluation of the CGAPP platform and comparison with other approaches that relax privacy, 2) evaluation of injection attacks and the robustness of countermeasures, and 3) data perturbation attacks. Finally, a discussion about the results is given.

5.1. *Scenario*

The key characteristics of this work scenario are the following: i) an automated industrial environment, ii) workers operating the production line using smartphones, iii) only a group of authorized workers can perform specific tasks, iv) the intrusion of an unauthorized worker or external person poses a severe risk, v) workers wear personal protective equipment that may prevent the capture of some biometric characteristics, and vi) the electronic device may be anchored at certain times and not portable. Attending to these characteristics, the S3-Dataset (previously detailed in Section 3.3.1) was used for all the experiments.

To emulate a group authentication situation, two groups of users were selected from all the S3-Dataset. The first group consisted of legitimate workers authorized to operate a given process who wished to be authenticated anonymously. The second group consisted of attackers who were also workers of the same company but lacked the proper authorization. The workers' group is formed by the identities 1, 2, 8, 11, 20, and 21 of the S3-Dataset. The second group of users, with identities 3, 4, 12, and 19, is considered to be the attackers. All other users in the dataset were discarded. This user selection was performed using a KNN classifier during the preparatory phase of the research. The users forming the workers' group showed a great deal of similarity and could then be considered as workers in the same operation line. The attackers showed some similarities among them, but with important differences with the workers. They could then be considered as workers of other areas of the company or operators of different lines. Finally, the discarded users were very different from those two groups and could be considered as not belonging to the same company.

Figure 3 shows a visual T-SNE representation of the training data. In this figure, the legitimate workers are colored with blue and green tones, while the four impostors are colored yellow, orange, red, and pink. As can be seen in the group of legitimate workers, their samples are very much intermixed, and clusters per worker can hardly be differentiated. On the contrary, the impostors do show very well differentiated clusters, with three types of workers: worker 12, very far from the group; workers 3 and 19, adjacent to the group; and worker 4, who is separated but is more surrounded by the samples of the group.

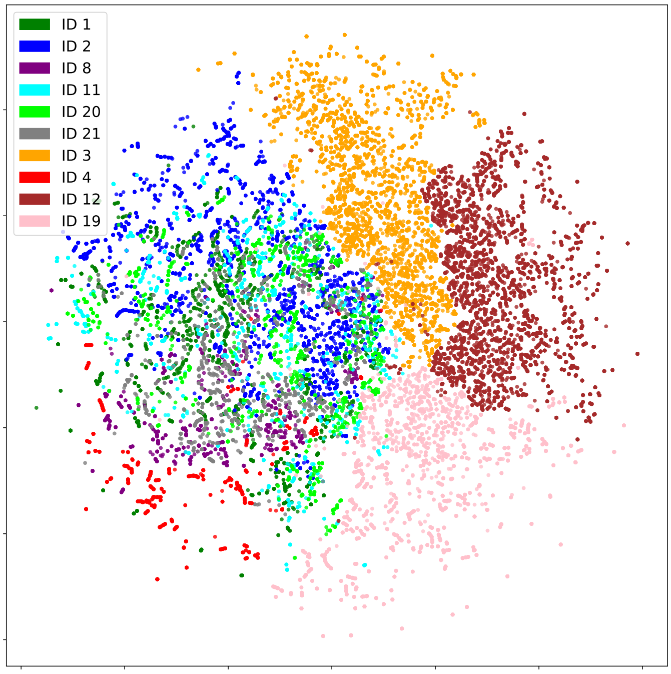

Figure 3: T-SNE for worker selected. Tones blue and green are the workers in the group, while yellow, orange, red, and pink are the impostors

The dataset has been split into train and test parts for the experiments. More precisely, the data of the first 14 days has been selected for training, which will generate the workers' profiles, and data from days 15th up to 60th has been used for testing. The choice of the 15th day for partitioning the data is based on the results of [18], which showed that unsupervised systems needed that many days of data to generate profiles with high enough precision.

It is important to note that although there are both positive and negative samples, this approach is unsupervised because the models are trained solely with negative examples, which are assumed in this work to belong to the group of legitimate workers. Positive examples are used only to evaluate performance during testing and represent people who are not in the group of legitimate workers. Therefore, even though both positive and negative examples are included in the evaluation, the approach is still considered unsupervised in the sense that there is no explicit labeling or annotation of the examples used for training (only negative ones). The labeling of positive samples as non-members of the legitimate group follows an attack detection interpretation. Hence, true positives are successfully detected attacks, and true negatives correspond to non-attacks that did not raise the alarm.

Likewise, $P_{attacker}$ is the number of outsider samples, N_{worker} is the number of legitimate worker samples, True Negatives (TN) denote correct authentications, True Positives (TP) show correct denials of authentication (rejections) to outside workers, False Negatives (FN) correspond to outside workers authenticated as legitimate, and False Positives (FP) represent a failure to authenticate a legitimate worker. *Accuracy* shows the rate of correct rejections for outside samples and correct authentication for genuine samples, *Precision* represents the rate of all rejections of the system that were truly outsider samples, and *Recall* is the true positive rate; that is, the rate of total outsider samples that were rejected by the system. *F1* is the geometric mean of Recall and Precision.

5.2. *Experiments*

The details and results of each experiment are now given. The experiments have focused on evaluating the performance of the CGAPP platform in terms of security and privacy and the robustness of the CGAPP platform against data injection attacks and also data perturbation attacks.

5.2.1. *Experiment 1: Performance of Group Authentication*

The objective of this first experiment is twofold: to evaluate the performance of the CGAPP Platform in the use case explained above and to compare it against other existing approaches in the literature that relax privacy concerns. To this end, two unsupervised approaches (outlier detection) with fewer security/privacy constraints are described below, providing the necessary details for their implementation without repeating redundant information from the federated approach.

Privacy Level 0: centralized approach. This approach has no privacy concerns, and all private data is sent to the server. The group authentication model (outlier detection model) is trained on the server, and the final model is sent back to each client. This approach has some significant advantages, such as the availability of greater computing power on the server for training models and the flexibility to use other artificial intelligence models beyond those providing FL capabilities. However, the main disadvantage is the complete breach of privacy resulting from private data leaving the devices and being sent to a centralized framework.

Privacy Level 1: Individual approach. In this approach, each legitimate group worker (i.e., client application) trains their outlier detection model on their own device using only their data. Once the model is trained, it is sent to the server where it is anonymized. After all models of group members have been collected in the server, they all are sent back to all the client applications. Scores and thresholds are calculated similarly as in the federated approach.

Privacy Level 2: Federated Learning. As described in section3.2, is an approach in which the model parameters/weights are calculated locally on the client's device during training, using the worker's private data. This data is only present on the personal device and never leaves it, ensuring full data privacy.

Once the approaches have been presented, they can be evaluated and compared with the FL approach to answer the first research question of Section 1. The Autoencoder, Variational Autoencoder, and KNN models are used for this experiment. These are models that have an FL algorithm for training.

Different model hyperparameters and scaling strategies were tested in a preliminary experimentation phase. Likewise the hyperparameters of each model have been optimized using the centralized approach and can be found in Table 2. The comparison of the different approaches keeps these hyperparameters fixed.

Regarding scaled strategies, for the centralized approach, the one that works best is min-max. For the individual approach, it was observed that individual scaling works best for each model. Finally, for the federated approach, the best is min-max, detailed in Section 3.3.2. For the federated approach (Privacy Level 2), the FedAvg algorithm was used, and different numbers of epochs were tested in a preliminary phase. The best results were found using 200 steps, which were then used in the final experiments.

Table 2: Hyperparameters for each model

Model	Best Params
Autoencoder	16 neurons
Variational A.	16 neurons
Autoencoder 2 layers	16-8 neurons
Variational A. 2 layers	16-8 neurons
KNN	n-neighbors = 50

The authentication results, using the previously defined metrics for each security level, are shown in Table 3. The global best metrics are shown in bold, and the best metrics for each privacy level are underlined (not done for results in bold).

The results show that the CGAPP platform (Privacy level 2) can achieve acceptable accuracy for operation in an industrial environment. The 2-layer Autoencoder has metrics better than 90% in all metrics but the lower bound of Precision, and it is the best model for accuracy and the F1 metric. This model, Autoencoder of 2 layers, is selected and it will be used for experiments 2 and 3. This model encompasses 725 trainable parameters and weighs approximately 80 kilobytes. To establish a baseline reference, the training and inference times of the model were evaluated in a federated environment on a PC equipped with an Intel Core i7-6800 3.40GHz processor. The training process was completed within 40 seconds, while the inference time amounted to a mere 0.03 milliseconds. To provide context, previous work [20], conducted measurements on a similar application running on the device. The results revealed an authentication time of approximately 1 millisecond, a dataset size of less than 1 megabyte, and an estimated battery consumption of approximately 150 milliampere-hours per hour of runtime.

Compared to the centralized approach (Privacy level 0), where there is no restriction to guarantee privacy, the CGAPP platform presents lower results but is close to the centralized approach. Regarding Privacy Level 1, where the individual approach is used, the results clearly show that the system's performance degrades significantly. In most algorithms, there is a drop of nearly 20 percentage points in accuracy, precision, and F1 can be seen. This may be due to the aggregation of different individual models. Remember that in this approach, whenever one of the individual models authenticates a user, they are authenticated as a member of the group. Hence, a false negative in any single individual model triggers an error in group authentication. If an individual model is poor, the entire group model is also poor. Furthermore, when the number of members in the group increases, there will be more individual models to use, and as a result, the chances of one model failing increase, and, consequently, the entire group authentication would fail.

5.2.2. *Experiment 2: Robustness Against Adversarial Attacks - Injection Attack*

This experiment studies the impact of Injection Attacks on the proposed CGAPP platform (privacy level 2) and the robustness of the countermeasures discussed in Section 4.2. To carry out this attack, the attacker will inject their data into the authentication system during the training phase using a compromised worker. Although this paper does not focus on the specific method of introduction, it could be performed using the worker's device during the training phase. In the experiment, all attackers will be cross-referenced with all workers, and different injection levels will be evaluated.

The most interesting metric to analyze this experiment is the False Negative rate, which can be calculated as $FNR_{attacker} = FN_{attacker}/P_{attacker}$. This rate shows the percentage of samples with which the attacker manages to fool the system after perpetrating the attack. Additionally, the true negative rate $TNR_{worker} = TN_{worker}/N_{worker}$ will be examined as well, which shows the rate of correct authentications for legitimate worker samples. This rate will illustrate how the behavior of the attacked system degrades. This experiment employs a ratio of

Table 3: Results of experiment 2. Comparison of the different approaches (privacy levels). The results are shown as 95% confidence intervals

Model	Privacy	Accuracy	Precision	Recall	F1
Autoencoder	Level 0	**96.03 - 96.29**	**97.55 - 98.26**	94.54 - 95.55	**96.32 - 96.57**
Autoencoder	Level 1	81.45 - 82.79	74.92 - 76.37	**99.33 - 99.57**	85.47 - 86.37
Autoencoder	Level 2	91.04 - 91.52	91.04 - 91.53	92.39 - 93.55	91.87 - 92.35
Autoencoder 2 layers	Level 0	95.33 - 95.60	95.40 - 95.97	95.86 - 96.24	95.75 -95.99
Autoencoder 2 layers	Level 1	74.73 - 78.26	68.94 - 72.20	98.42 - 98.84	81.11 - 83.31
Autoencoder 2 layers	Level 2	91.92 - 92.21	89.82 - 90.21	95.94 - 96.44	92.87 - 93.13
KNN	Level 0	87.80 - 91.78	83.31 - 88.97	96.85 - 98.04	89.93 - 92.78
KNN	Level 1	78.09 - 78.09	88.22 - 88.22	69.27 - 69.27	77.60 - 77.60
KNN	Level 2	87.67 - 90.58	86.65 - 89.37	91.02 - 94.73	89.00 - 91.66
Variational A.	Level 0	95.71 - 96.21	96.05 - 97.11	95.65 - 96.45	96.09 - 96.52
Variational A.	Level 1	76.80 - 78.47	70.66 - 72.34	98.39 - 98.79	82.35 - 83.38
Variational A.	Level 2	91.36 - 91.60	89.28 - 89.48	95.56 - 96.10	92.38 - 92.61
Variational A. 2 layers	Level 0	95.06 - 95.26	95.03 - 95.51	95.76 - 96.12	95.51 - 95.69
Variational A. 2 layers	Level 1	73.77 - 75.87	68.07 - 69.94	98.15 - 98.57	80.45 - 81.73
Variational A. 2 layers	Level 2	91.98 - 92.10	89.40 - 89.55	96.74 - 96.98	92.97 - 93.08

attacker samples to compromised but genuine worker samples, i.e., samples from outside sources over legitimate ones. Using only a single compromised worker, the metrics analyzed in the previous stage will be examined across a range of ratios to illustrate how the intensity of the injection could potentially degrade the system. The ratio ranges between 0.005 to 1.00. Note that 1.00 means the presence of as many outsider data samples as from the legitimate worker. A higher sample ratio has not been considered since it means that there would be more training samples from the attacker than from the legitimate worker. From the unsupervised modeling viewpoint in general, and anomaly detection in particular, the group with more samples would naturally be interpreted as the normal one.

Figure 4 shows the attacker's FNR (left column) and worker's TNR (right column) metrics for each attack. The attacks are conducted by different attack workers (IDs 3, 4, 12, and 19) using a particular legitimate worker compromised (IDs 1, 2, 8, 11, 20, and 21) to inject its data. Analyzing first the FNR plots, three different behaviors can be observed. In the first place, attacker 4, which in Figure 3 was isolated from the other attackers, shows a high miss rate from the beginning. However, even when it injects all of its data, its FNR is at most 30 percent. On the other hand, attacker 12 needs a ratio higher than 0.2 to get a noticeable FNR. Finally, workers 3 and 19 show some FNR from the beginning (with higher values for worker 3), and the percentage of miss rates increases as the ratio increases.

The FNR plots indicate that worker 2 (dark blue line) is the most vulnerable worker in the group, as injection of attacker data causes the biggest disruption. Attackers 3, 12, and 19 are able to perpetrate a significant number of authentications with a lower sample rate when worker 2 is the compromised one. In contrast, worker 21 is the most robust against this type of attack, with very few successful attacks from worker 12.

Additionally, in the second column of Figure 4, the worker's TNR can be observed when attackers inject their data. This metric shows either an increment or remains very stable. Only for attacker 4 and workers 2 and 20 a worsening of the metric is observed as the ratio increases, but minimal. This system behavior would imply that the compromised worker would not notice that they are being the victim of an attack, which would aggravate the situation. After analyzing these metrics, the global TNR of the system was also evaluated to check if the rest of the workers would be harmed, but the results were similar. This suggests that when new data is injected, the system becomes more permissive and accepts more samples as genuine, causing dubious samples of a genuine worker that would have been previously rejected (a false positive) to now be correctly recognized (true negatives). Hence the TNR improves.

Given the danger of this type of attack, the different countermeasure aggregation methods mentioned in Section 4.2 are evaluated below to mitigate the effects of these attacks. Figure 5 shows the same data as Figure 4 but now uses the median of the weights as the aggregation method. The other aggregation function proposed in Section 4.2 has also been evaluated, but it did not show a noticeable improvement over using the median. Figure 5 shows a significant reduction of the FNR for attackers 3, 12, and 19. The behavior in attacker 4 remains similar. It is worth noting that for attacker 12, except for worker 2, the FNR barely exceeds 15%, for all ratios.

Figure 6 shows a bar chart for the most vulnerable worker of the group, worker 2. In addition, the sample ratio has been set to 0.5 since, at that point, the worker causes a miss rate higher than 30 percent for all attackers. If analyzed, worker 2 has about 2900 samples in the training set, which means about 48 hours of device use in the 15 days, an average of 3.5 hours per day. Therefore, at the 0.5 ratio, the impostors should enter about 24 total hours or 1.75 hours per day. As seen in the bar chart, the proposed aggregation methods improve on the standard *mean*, with the *median* being the best-performing system, reducing the miss rate caused by the attacker to about half. The

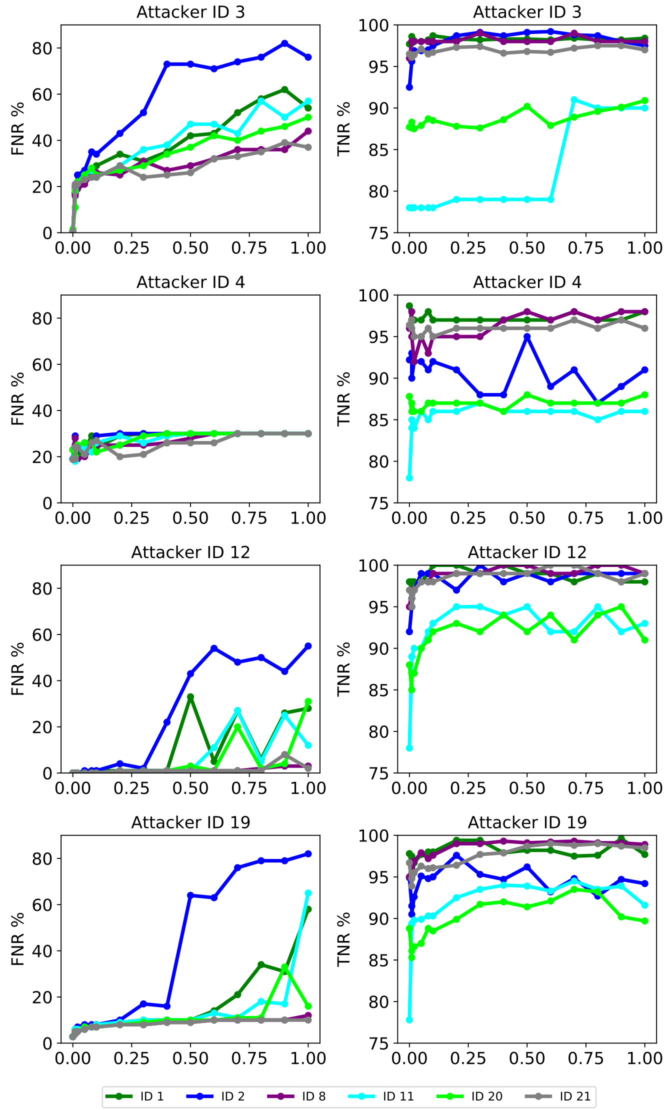

Figure 4: The left column of the y-axis is the attacker's FNR, and the right column is the worker's TNR of the authentication system for a backdoor attack when an attacker is able to inject its own data as if it were data of one compromised genuine worker. The X-axis is the ratio of data between the attacker and the compromised worker. Each row is a different attacker, and each color line represents the same compromised worker in all the graphics. The aggregation method used for this plot is the mean

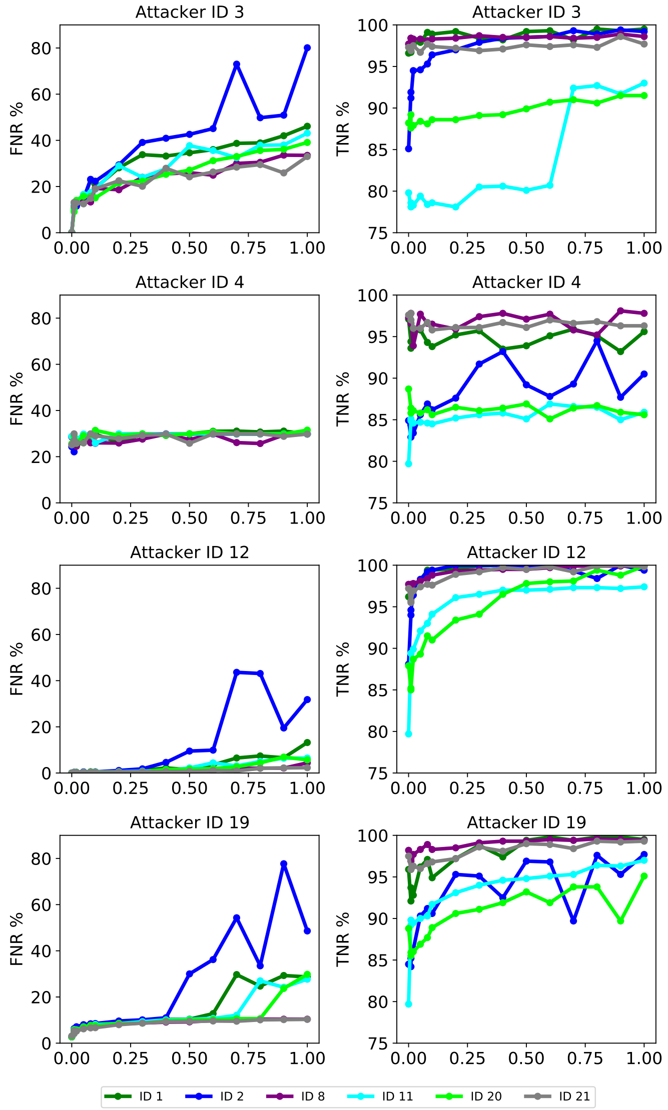

Figure 5: The y-axis is the attacker's FNR, left column, and the worker's TNR, right column, for a backdoor attack when a compromised member of the group introduces data for an attacker. The X-axis is the ratio of data between the attacker and the compromised worker. Each row is a different attacker, and each color line represents the same compromised worker in all the graphics. The aggregation method used for this plot is the median

exception is attacker 4, where any aggregation function shows no improvement.

As observed in the results of this experiment, an injection attack poses a severe risk to the system. Not only can the attacker access the system, but the group worker's usability remains unaffected, and the TNR even improves, making it much more challenging to detect the compromised worker to stop the injection attack. Fortunately, the results suggest that if the considered countermeasures are applied, the FNR rate will be reduced, increasing the system's robustness against injection attacks. Among the countermeasures shown, the median offers the best robustness against these attacks.

5.2.3. *Experiment 3: Robustness Against Adversarial Attacks - Data Perturbation*

This final experiment studied the impact of Data Perturbation Attacks on the proposed CGAPP platform (Privacy Level 2) and the robustness of the countermeasures that were discussed in Section 4.2. To perpetrate this attack, a worker must contaminate its data in a way that breaks the authentication system. In this experiment, two methodologies of perturbing the data are evaluated. In addition, the behavior of several workers carrying out the attack and the level of the attack, considered by the proportion of corrupted samples, are tested.

A simple-to-achieve and hard-to-detect attack would consist in contaminating a worker's data using its own dispersion range or that of the legitimate group as a whole. This experiment

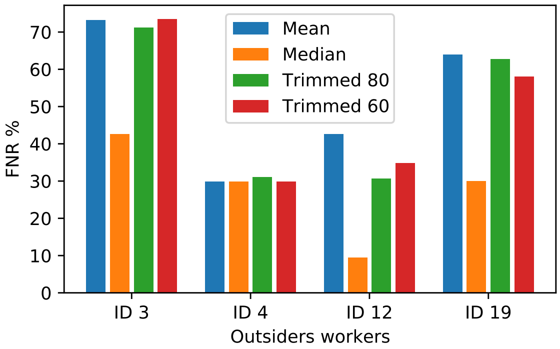

Figure 6: Bar plot for the FNR in an injection attack of the group worker ID 2, at a ratio of 0.5. The different colors represent the different aggregation methods of federated weights

addresses this strong attack, where the data is contaminated by replacing a percentage of the worker's genuine data with random data following a uniform distribution in its dispersion range considering i) the compromised worker's data only and ii) data of all the workers in the group. This will create uniform clouds in the input space, covering the whole range of input values present in-group members, effectively eliminating any complexity in its shape and allowing substantial areas of the input space as genuine.

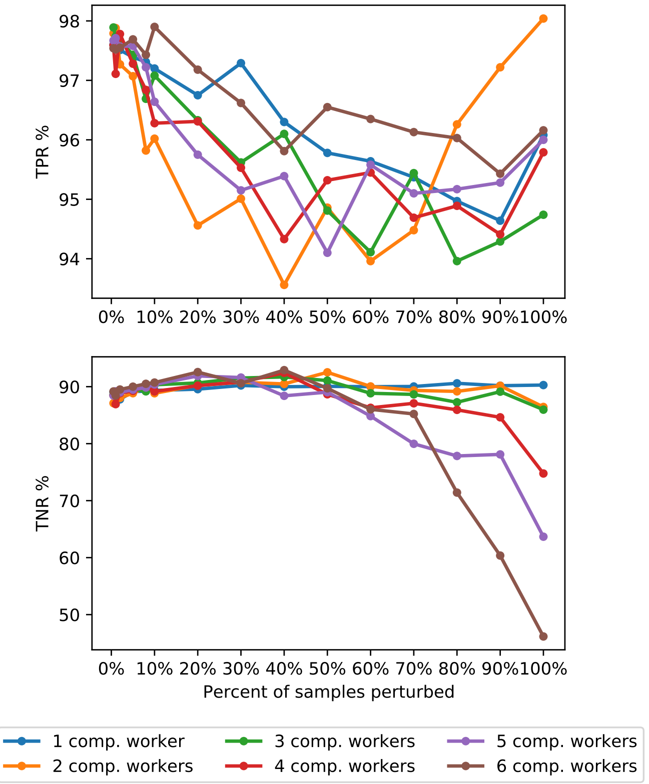

Figure 7: TPR (Upper) and TNR (Lower) for each percent (X-axis) of samples poisoned from the different numbers of compromised workers (color lines), ranging from 1 to 6 compromised workers

The first study considers that a compromised worker contaminates its data by following its own data dispersion range. For this experiment, a ranging percent of samples contaminated are considered, as well as different numbers of compromised workers. Figure 7 shows the obtained TPR and TNR depending on the percentage of contaminated samples and the number of compromised workers. As can be seen, the TPR shows a downward trend and accumulates a loss of less than four percentage points. Meanwhile, the TNR is very stable for a single compromised worker and decreases progressively when the number of compromised workers increases. Even in the case of six compromised workers, when 80 percent of the data is contaminated, the TPR is already reduced to 70 percent. The behavior observed in this graph shows that the system loses a little security, but its usability is greatly affected.

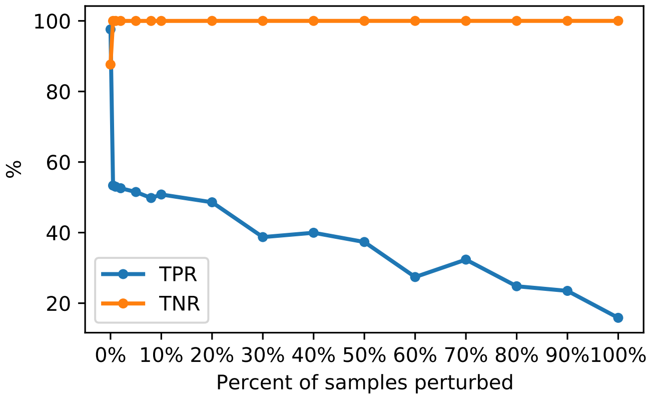

Figure 8: TPR and TNR for each percent (X-axis) of samples poisoned from a single compromised client

Next, the case when an attacker uses data that follows the feature value range of that of all workers is studied. This information can be obtained by the client app of a compromised worker because the scaling module of the system has the global maximum and minimum of each feature of the data. As previously, the experiment would vary the percent of perturbed samples and the number of workers compromised. The results are shown in Figure 8. Since the system degrades rapidly, in this figure, only the case of a single compromised worker is shown. In this type of attack, the above hypothesis of creating uniform clouds in the input hyperspace is tested. The TNR increases to 100% so that no sample from a worker in the group is rejected. In comparison, TPR plummets just introducing a few contaminated data. In this case, security has been breached, and any worker can easily access the system, while usability has not been affected. This attack may not be detected by genuine workers, as they would be authenticated as usual (even with fewer false rejections) as their system authentication performance is unaffected. System administrators could detect it if they perceive that the rate of global rejections drops. However, this can be difficult to detect in environments where the number of attacks is overwhelmingly lower than genuine checks.

As in the case of the previous experiment, the data perturbation attacks can significantly compromise the security of the CGAPP platform. Therefore, it was necessary to evaluate whether the different aggregation proposals could mitigate this situation. The experiments conducted to this end showed that all the countermeasure aggregation mechanisms work similarly, and none is an improvement over the original one. The fact that no countermeasure can alleviate the effects of this attack highlights the clear danger it poses and underscores the need for future work to find a solution against this type of attack.

5.3. Discussion

5.3.1. Experiments

The results presented in this work depend highly on the chosen dataset and the distribution of workers. Therefore, the first task performed at the beginning of the research was to select workers to be characterized as the legitimate group. These workers would be those whose behavior was similar to that of workers in an industry with similar jobs or tasks. On the other hand, other group of workers was selected, looking for those workers whose behavior was similar yet different and who could be considered employees of the same company but in different positions or ranks. The rest of the workers who did not fit into these two groups were discarded. This makes the problem harder than if very different workers were selected as the legitimate group versus the non-authorized group.

Experiment 1, Section 5.2.1, evaluated the overall performance and accuracy that could be achieved in this type of problem with different levels of privacy. The best model, the Autoencoder, showed an accuracy of 96%. Subsequently, as data privacy increased with the individual and federated approaches, the system performance degraded, as could be expected. In the federated approach, it suffered a minor degradation, achieving 92% accuracy. However, in the individual approach (Privacy Level 1), the degradation was much more significant, dropping to 82%, more than 14 percentage points. The particularities of the workers and their quite reduced number may cause this fact. As can be seen in Figure 3, which shows the TSNE of the workers, the samples of the workers in the group are distributed in such a way that the workers in the group cannot be distinguished internally. This may result in the individual models being unable to fit each user's behavior well. Therefore, the aggregation of all of them does not constitute a good model. These results clashed with the anticipated results since an intermediate performance between the other two approaches was expected for this approach.

In Experiment 2, Section 5.2.2, the response to injection attacks was evaluated. Figure 4 shows the different behaviors presented by the different attackers. The color lines also show the different workers' behavior. Among the peculiarities, attacker 4 stands out since, in the beginning, independently of the worker that has been compromised, it is able to get accepted with around 30% of the samples. Yet, no matter how much it increases the attack, it does not manage to increase its percentage of success. This points to an attacker that, while some of its own behavior is already and previously similar to the group members (near 20%), the non-similar areas of its characterization cloud are too far from the group model for the learning algorithm to be pushed to enclose it, even with a quite substantial amount of injection. Injecting more attacker data than the genuine data of the compromised worker is out of the scope of this work because, as mentioned previously, that would be a case of a worker replacement attack rather than a data injection attack. Attackers 3, 12, and 19 manage, at different ratios, to perform more successful attacks. Experiment 2 also allows the establishment of a sort of ranking from weakest to strongest among the legitimate workers who are compromised: 2, 1, 11, 20, 8, and 21. The most important thing to note is that none of the attacked workers would detect that they are being compromised because the attackers do not hamper their authentication rates and even improve them.

Finally, in Experiment 3, Section 5.2.3, the group workers' data is perturbed in two different ways: first with their own input ranges, and then with the characteristics range of the whole group. For the first type of perturbation, there is hardly any strong degradation if only one worker is contaminating its data, and several workers need to be compromised to degrade the system. However, with the second type of perturbation, the effect is immediate and very drastic: the TNR rises to 100%, and the TPR decays rapidly. Fortunately, this attack would easily alert workers that an attack is underway and put the system on alert.

The experiment results show the great potential of the CGAPP platform for its use by the industry, even in situations where the number of workers is rather small. The CGAPP platform provides CGA based on FL in a novel way, increasing workers' privacy and their data while keeping an accuracy close to that obtained by systems that do not protect privacy. However, the system shows some weaknesses when confronted with adversarial attacks, albeit some may be substantially alleviated with the evaluated countermeasures. All in all, before incorporating this platform into the productive industrial environment, it would be necessary to carry out a series of works to evaluate and minimize the limitations of this work.

5.3.2. General Comments

Despite the contributions and advancements made in this study, it is important to acknowledge certain limitations that warrant further consideration. Firstly, the transmission of preprocessing and threshold values required for the operation of the system raises concerns regarding privacy preservation. In this work, it has been considered that sharing such data between the client and the trusted server through encrypted communication does not result in privacy compromise. This is due not only to the fact that these values do not directly involve private personal data collected from users, unlike the data vectors that remain exclusively on personal devices, but also because other clients and external entities are unable to access this information anyway, thereby maintaining the confidentiality of the client's data. However, it would be valuable to explore methods that eliminate the explicit sharing of this information or embed it within the parameters of the federation. Additionally, studying the extent to which information can be inferred from these data would be worthwhile.

Next, a validation in a realistic and industrial scenario, evaluating and validating the platform deployed in a company, would provide more robust insights into its practical effectiveness.

Additionally, this work focuses on a single group within the continuous authentication system. However, exploring the potential for multiple groups with different roles and privileges, delving into the study of their interrelationships, and investigating their performance would be valuable directions for future research. Gaining a comprehensive understanding of the dynamics and challenges associated with managing multiple groups would enhance the applicability and scalability of our approach.

Furthermore, an aspect that deserves future further attention in the current work is the limited scope of adversary attacks studied. Specifically, only two types of Data Poisoning attacks have been considered, where it has been assumed that the server is trustworthy and that the clients do not have the knowledge to carry out complex attacks. Other types of adversarial attacks are membership inference or model inversion attacks. Looking at Figure 3, it can be seen how the workers' data are very mixed, which could indicate that this type of attack would have moderate success, but it should be evaluated in the future.

Another notable aspect of the work that requires future consideration is the handling of compromised workers within the continuous group authentication system, an aspect not explicitly addressed. Devising effective strategies to detect and manage the presence of corrupt workers poses a significant challenge. While this paper focuses on developing and validating the authentication framework, the specific mechanisms for dealing with compromised workers are beyond its scope. Multiple options, such as implementing anomaly detection algorithms, employing redundancy mechanisms, or conducting regular audits, can be considered. However, determining the most suitable approach requires expertise from the company's security department. Thus, it is essential for the security department to collaborate with the research team to devise appropriate measures to mitigate the impact of compromised workers, ensuring the overall robustness and security of the authentication system.

6. Conclusions

This work presents the Continuous Group Authentication Platform (CGAPP), an innovative solution that serves as a proof of concept for industrial companies seeking to develop and implement non-intrusive continuous group authentication for their workforce. By adopting the CGAPP platform, companies can enhance factory security while wnsuring the privacy of workers' data. This is achieved through the CGAPP platform design and development, which follows an outlier detection approach and is trained with a federated learning scheme that provides data privacy to the workers.

A series of experiments were conducted to validate the CGAPP platform in an industry-centric scenario using an existing and public dataset, providing promising evidence of feasibility. Dataset S3 has the following features that satisfy the peculiarities of the application scenario: i) an automated industrial environment, ii) electronic devices utilized by workers, iii) specific tasks performed by a designated group of authorized workers, iv) the presence of an unauthorized external individual or worker poses a grave threat, v) personal protective equipment worn by workers may impede the capture of certain biometric traits, and vi) at certain times, the electronic device may be stationary and not portable. The S3 Dataset has behavior stats of 21 volunteers using their smartphones for over 60 days. Various users, both male, and female, between 18 to 70 years old, were involved in the dataset creation.

The first objective of the work was to evaluate the performance of the federated system in an industrial environment and compare it with other approaches that may compromise data privacy. It is related to the first research question, "In an industrial scenario, what is the cost in the accuracy of using systems that preserve data privacy versus systems that require full data access in training and operation?". To answer it, Experiment 1 has been carried out, where the federated, individual, and centralized approaches are compared. The best algorithm found for the federated learning approach, where no private data ever leaves personal devices, is the Autoencoder with two layers, with an accuracy of nearly 92%. Then, the performance when privacy is relaxed was also evaluated. In the individual approach, where full models of each individual worker are sent to the server, the accuracy drops to 82% while, in the centralized approach, where full worker private data is sent to the server, it grows to 96%. To address the next research question concerning security versus attacks, the robustness of the federated approach has been studied by evaluating its susceptibility to different types of attacks and how it would affect it. Specifically, in this work, two poisoning attacks have been considered. The first, evaluated in Experiment 2, is an injection attack in which an external subject compromises a legitimate worker by injecting its own data into the model, thereby trying to be recognized as a legitimate user. Alleviating countermeasures were proposed and tested. Finally, Experiment 3, addresses the last research question, which is also related to security but concerns data perturbation attacks, a type of poisoning attack aimed at breaking down the authentication system. The results of the experiment indicate that the robustness of the authentication system against data perturbation attacks depends largely on the specific characteristics of the worker and of the attacker. In some situations, the attack barely succeeds, while in others, it can have a success rate greater than 80%. The most important conclusions drawn are that the attacker's success rate increases as the ratio of its data samples to those of the compromised workers grows, as can be expected, and that the attack can go unnoticed by the workers of the group. However, the use of alternative aggregation measures, such as the median, can mitigate this situation.

Future work in this area will aim to address the limitations identified and mentioned in Section 5.3, and explore various avenues for improvement. This includes considering additional characteristic features for continuous authentication, such as incorporating sensor data or similar information. Furthermore, the study will be extended to encompass a broader range of Machine Learning models, allowing for a more comprehensive evaluation. Efforts will be made to overcome the limitations associated with the dataset used in this study. This may involve acquiring an actual industrial environment dataset or a dataset that better represents the characteristics of the specific scenario, including the inclusion of real adversarial attacks. By incorporating such data, the research can provide a more realistic evaluation of the proposed approaches and their effectiveness. Lastly, the research will explore techniques to effectively detect and handle corrupted workers within the federated learning framework.

References

[1] Alcaraz, P., Zhang, P., Cardenas, P. & Zhu, P. Guest Editorial: Special Section on Security and Privacy in Industry 4.0. *IEEE Transactions On Industrial Informatics*. **16**, 6530-6531 (2020)

[2] Gao, Z., Castiglione, A. & Nappi, M. Guest Editorial: Biometrics in Industry 4.0: Open Challenges and Future Perspectives. *IEEE Transactions On Industrial Informatics*. **18**, 9068-9071 (2022)

[3] Spolaor, R., Li, Q., Monaro, M., Conti, M., Gamberini, L. & Sartori, G. Biometric Authentication Methods on Smartphones: A Survey.. *PsychNology Journal*. **14** (2016)

[4] Alzubaidi, A. & Kalita, J. Authentication of Smartphone Users Using Behavioral Biometrics. *IEEE Communications Surveys & Tutorials*. **18**, 1998-2026 (2016)

[5] Espin Lopez, J., Huertas Celdran, A., Esquembre, F., Martinez, G. & Marin-Blazquez, J. A supervised ML Biometric Continuous Authentication System for Industry 4.0. *IEEE Transactions On Industrial Informatics*. pp. 1-1 (2022)

[6] Harn, L. Group Authentication. *IEEE Transactions On Computers*. **62**, 1893-1898 (2013,9)

[7] McMahan, H., Moore, E., Ramage, D., Hampson, S. & Arcas, B. Communication-Efficient Learning of Deep Networks from Decentralized Data. *AISTATS*. pp. - (2017)

[8] Jere, M., Farnan, T. & Koushanfar, F. A Taxonomy of Attacks on Federated Learning. *IEEE Security & Privacy*. **19**, 20-28 (2021)

[9] Espín López, J. CGAPP Platform. *GitHub Repository*. (2022), https://github.com/bazako/CGAPP_Platform

[10] Guo, C., Zhuang, R., Yuan, L. & Feng, B. A Group Authentication Scheme Supporting Cheating Detection and Identification. *2015 Ninth International Conference On Frontier Of Computer Science And Technology (FCST)*. pp. 110-114 (2015,8), https://doi.ieeecomputersociety.org/10.1109/FCST.2015.52

[11] Aydin, Y., Kurt, G., Ozdemir, E. & Yanikomeroglu, H. A Flexible and Lightweight Group Authentication Scheme. *IEEE Internet Of Things Journal*. **7**, 10277-10287 (2020)

[12] Liu, J., Zhang, L., Li, C., Bai, J., Lv, H. & Lv, Z. Blockchain-Based Secure Communication of Intelligent Transportation Digital Twins System. *IEEE Transactions On Intelligent Transportation Systems*. pp. 1-11 (2022)

[13] Xu, R., Wang, X. & Morozov, K. Group authentication for cloud-to-things computing: Review and improvement. *Computer Networks*. **198** pp. 108374 (2021), https://www.sciencedirect.com/science/article/pii/S138912862100356X

[14] Pang, S., Kim, D. & Bang, S. Membership authentication in the dynamic group by face classification using SVM ensemble. *Pattern Recognition Letters*. **24**, 215-225 (2003), https://www.sciencedirect.com/science/article/pii/S0167865502002131

[15] Li, Y., Hu, H., Zhu, Z. & Zhou, G. SCANet: sensor-based continuous authentication with two-stream convolutional neural networks. *ACM Transactions On Sensor Networks (TOSN)*. **16**, 1-27 (2020)

[16] Li, Y., Luo, J., Deng, S. & Zhou, G. CNN-Based Continuous Authentication on Smartphones With Conditional Wasserstein Generative Adversarial Network. *IEEE Internet Of Things Journal*. **9**, 5447-5460 (2022)

[17] Shuwandy, M., Aljubory, H., Hammash, N., Salih, M., Altaha, M. & Alqaisy, Z. BAWS3TS: Browsing Authentication Web-Based Smartphone Using 3D Touchscreen Sensor. *2022 IEEE 18th International Colloquium On Signal Processing & Applications (CSPA)*. pp. 425-430 (2022)

[18] Espín López, J., Huertas Celdrán, A., Marín-Blázquez, J., Esquembre, F. & Martínez Pérez, G. S3: An AI-Enabled User Continuous Authentication for Smartphones Based on Sensors, Statistics and Speaker Information. *Sensors*. **21** (2021), https://www.mdpi.com/1424-8220/21/11/3765

[19] Jorquera Valero, J., Sánchez Sánchez, P., Fernández Maimó, L., Huertas Celdrán, A., Arjona Fernández, M., De Los Santos Vílchez, S. & Martínez Pérez, G. Improving the Security and QoE in Mobile Devices through an Intelligent and Adaptive Continuous Authentication System. *Sensors*. **18**, 3769 (2018,11), http://dx.doi.org/10.3390/s18113769

[20] Sánchez Sánchez, P., Fernández Maimó, L., Huertas Celdrán, A. & Martínez Pérez, G. AuthCODE: A privacy-preserving and multi-device continuous authentication architecture based on machine and deep learning. *Computers & Security*. **103** pp. 102168 (2021), http://www.sciencedirect.com/science/article/pii/S0167404820304417

[21] Barlas, Y., Basar, O., Akan, Y., Isbilen, M., Alptekin, G. & Incel, O. DAKOTA: Continuous Authentication with Behavioral Biometrics in a Mobile Banking Application. *2020 5th International Conference On Computer Science And Engineering (UBMK)*. pp. 1-6 (2020)

[22] Santos, U., Costa, C., Mayer, A., Reis, E., Maldonado, J., Barbosa, J., Antunes, R., Righi, R. & Flores, N. Trends in User Identity and Continuous Authentication. *Computer*. **55**, 52-61 (2022)

[23] Phillips, T., Yu, X., Haakenson, B., Goyal, S., Zou, X., Purkayastha, S. & Wu, H. AuthN-AuthZ: Integrated, User-Friendly and Privacy-Preserving Authentication and Authorization. *2020 Second IEEE International Conference On Trust, Privacy And Security In Intelligent Systems And Applications (TPS-ISA)*. pp. 189-198 (2020)

[24] Cheng, Y., Meng, H., Lei, Y. & Tan, X. Research on Privacy Protection Technology in Face Identity Authentication System Based on Edge Computing. *2021 IEEE International Conference On Artificial Intelligence And Industrial Design (AIID)*. pp. 438-449 (2021)

[25] Oza, P. & Patel, V. Federated Learning-based Active Authentication on Mobile Devices. *2021 IEEE International Joint Conference On Biometrics (IJCB)*. pp. 1-8 (2021)

[26] Hernández-Álvarez, L., Fuentes, J., González-Manzano, L. & Hernández Encinas, L. Privacy-Preserving Sensor-Based Continuous Authentication and User Profiling: A Review. *Sensors*. **21** (2021), https://www.mdpi.com/1424-8220/21/1/92

[27] Espín López, J., Huertas Celdrán, A., Marín-Blázquez, J., Esquembre, F. & Pérez, G. S3 Dataset. (figshare,2021,4), https://figshare.com/articles/dataset/S3Dataset_zip/14410229/2

[28] Rodríguez-Barroso, N., Jiménez-López, D., Luzón, M., Herrera, F. & Martínez-Cámara, E. Survey on federated learning threats: Concepts, taxonomy on attacks and defences, experimental study and challenges. *Information Fusion*. **90** pp. 148-173 (2023), https://www.sciencedirect.com/science/article/pii/S1566253522001439

[29] Yin, D., Chen, Y., Ramchandran, K. & Bartlett, P. Byzantine-robust distributed learning: Towards optimal statistical rates. *Byzantine-Robust Distributed Learning: Towards Optimal Statistical Rates*. pp. 5650 - 5659 (2018)

Juan M. Espín López received the M.Sc. degree in mathematics from the University of Murcia. He is currently pursuing his PhD in computer science at the University of Murcia. His research interests are focused on CA, speaker recognition, facial recognition, anti-spoofing systems and the application of machine learning and deep learning to the previous fields.

Alberto Huertas Celdrán received the M.Sc. and Ph.D. degrees in computer science from the University of Murcia, Spain. He is currently with the Communication Systems Group (CSG), Department of Informatics (IfI), University of Zürich UZH. His scientific interests include IoT, brain-computer interfaces (BCI), cybersecurity, data privacy, artificial intelligence, semantic technology, and computer networks.

Francisco Esquembre is Full Professor in the Department of Mathematics of the University of Murcia, Spain. His scientific activity is mostly devoted to mathematical modeling and computer simulation of physical and engineering phenomena, developing the Easy Java Simulations modeling tool. He is currently interested in the application of data analysis to different practical problems.

Gregorio Martinez Pérez is Full Professor in the Department of Information and Communications Engineering of the University of Murcia, Spain. His scientific activity is mainly devoted to cybersecurity and networking. He is working on different national and European IST research projects related to these topics, being Principal Investigator in most of them. He has published 160+ papers in national and international conference proceedings, magazines and journals.

Javier G. Marín-Blázquez received both Computer Science (1994) and Psychology (2012) degrees by the University of Murcia, and a M.Sc. (2001) and PhD (2003) in Artificial Intelligence by The University of Edinburgh. His research interests include: Artificial Intelligence (AI), Machine Learning, Fuzzy Systems, Soft Computing, Cybersecurity, AI for Games and Cognitive Science.