JYX JYVÄSKYLÄN YLIOPISTO UNIVERSITY OF JYVÄSKYLÄ

This is a self-archived version of an original article. This version may differ from the original in pagination and typographic details.

Author(s): Terziyan, Vagan; Malyk, Diana; Golovianko, Mariia; Branytskyi, Vladyslav

Title: Encryption and Generation of Images for Privacy-Preserving Machine Learning in Smart Manufacturing

Year: 2023

Version: Published version

Copyright: © 2022 The Authors. Published by Elsevier B.V.

Rights: CC BY-NC-ND 4.0

Rights url: https://creativecommons.org/licenses/by-nc-nd/4.0/

Please cite the original version:

Terziyan, V., Malyk, D., Golovianko, M., & Branytskyi, V. (2023). Encryption and Generation of Images for Privacy-Preserving Machine Learning in Smart Manufacturing. In F. Longo, M. Affenzeller, A. Padovano, & S. Weiming (Eds.), 4th International Conference on Industry 4.0 and Smart Manufacturing (217, pp. 91-101). Elsevier. Procedia Computer Science. https://doi.org/10.1016/j.procs.2022.12.205

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 217 (2023) 91–101

Procedia Computer Science

www.elsevier.com/locate/procedia

4th International Conference on Industry 4.0 and Smart Manufacturing

Encryption and Generation of Images for Privacy-Preserving Machine Learning in Smart Manufacturing

Vagan Terziyan [a]*, Diana Malyk [b], Mariia Golovianko [b], Vladyslav Branytskyi [b]

[a] *Faculty of Information Technology, University of Jyväskylä, 40014, Jyväskylä, Finland*
[b] *Department of Artificial Intelligence, Kharkiv National University of Radio Electronics, 61166, Kharkiv, Ukraine*

Abstract

Current advances in machine (deep) learning and the exponential growth of data collected by and shared between smart manufacturing processes give a unique opportunity to get extra value from that data. The use of public machine learning services actualizes the issue of data privacy. Ordinary encryption protects the data but could make it useless for the machine learning objectives. Therefore, "privacy of data vs. value from data" is the major dilemma within the privacy preserving machine learning activity. Special encryption techniques or synthetic data generation are being in focus to address the issue. In this paper, we discuss a complex hybrid protection algorithm, which assumes sequential use of two components: homeomorphic data space transformation and synthetic data generation. Special attention is given to the privacy of image data. Specifics of image representation require special approaches towards encryption and synthetic image generation. We suggest use of (convolutional, variational) autoencoders and pre-trained feature extractors to enable applying privacy protection algorithms on top of the latent feature vectors captured from the images, and we updated the hybrid algorithms composed of homeomorphic transformation-as-encryption plus synthetic image generation accordingly. We show that an encrypted image can be reconstructed (by the pre-trained Decoder component of the convolutional variational autoencoder) into a secured representation from the extracted (by either the Encoder or a feature extractor) and encrypted (homeomorphic transformation of the latent space) feature vector.

© 2022 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0)
Peer-review under responsibility of the scientific committee of the 4th International Conference on Industry 4.0 and Smart Manufacturing

Keywords: Industry 4.0; data privacy; anonymization; syntetic data generation; image processing; autoencoders

* Corresponding author. *E-mail address:* vagan.terziyan@jyu.fi

1877-0509 © 2022 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0)
Peer-review under responsibility of the scientific committee of the 4th International Conference on Industry 4.0 and Smart Manufacturing
10.1016/j.procs.2022.12.205

1. Introduction

"Top Secret. Burn before reading!"
Boris & Arkady Strugatsky (1965).
"Monday Begins on Saturday"
en.wikipedia.org/wiki/Monday_Begins_on_Saturday

Smart manufacturing under the umbrella of Industry 4.0 has dramatically increased its efficiency with the advancement in artificial intelligence (AI) and, particularly, in machine learning (ML) techniques and related applications [1]. Such applications enable smart production lines with improved flexibility of the product designs and configurations and further analysis of the usage data. The industry 4.0 paradigm encourages the usage of smart sensors, devices and machines to enable smart factories that continuously collect the production process and the product usage data. ML techniques enable the discovery of complex patterns used by decision support systems for many manufacturing tasks. However, many open questions and challenges remain regarding the use of ML applications [2], e.g., big data management and understanding, especially data-related security and privacy concerns. Unfortunately, smart manufacturing is one of the most targeted industries for cyber-attacks, costing billions of dollars worldwide. Attackers often aim to steal either intellectual property or private information (e.g., employees' or customers' data) and sell it to competitors or blackmail the victims. Unlike other industrial revolutions, Industry 4.0 measures, generates, collects, preserves, and shares lots of private information. Personal data has already turned out to be a new domain of warfare against the key Industry 4.0 components such as AI, ML, internet of things, cyber-physical systems and autonomous vehicles, cloud computing, etc. [3]. Personal data nowadays, in addition to customer data and sensitive process data, may even include captured models of workers' behavior while accident handling [4]. Knowledge sharing, which is an important enabler for optimized manufacturing processes (e.g., predictive maintenance among many others), suffers a lot from sensitive data disclosure issues. Potentially sensitive data may not be evident within the maintenance reports and must be identified, removed or anonymized before being shared with other actors [5].

The exponential growth of big data collected by smart manufacturing processes and qualitatively new methods to process the data gives a unique opportunity to get extra value from the data. ML in general and deep learning in particular are increasingly being adopted everywhere in Industry 4.0. Usually, a well-performing ML model relies on a large volume of training data and high-powered computational resources. Therefore, ML-as-a-Service (MLaaS), which leverages deep learning techniques provided by public clouds for predictive analytics to enhance decision-making, has become a hot commodity [6]. Providing a training set to some external storage-as-a-service cloud [7] and further to MLaaS can be difficult when sensitive user data is involved and the General Data Protection Regulation (GDPR) or similar regulations prohibit gathering and processing such sensitive data [8]. The privacy concerns are well justified due to the potential risks of leakage of highly privacy-sensitive information and due to the vulnerability of trained ML models to adversarial attacks [9]. Therefore, privacy-preserving ML (PPML) has emerged to enable the value from data without violating its privacy [10]. The existing work on PPML goes mainly into two directions: the variations of privacy-preserving cryptography including homomorphic encryption and secure multi-party computing [10], [8], [11], and the variations of data perturbations including synthetic data generation with differential privacy [12], [13], [14], each having its advantages and disadvantages.

Machine vision, which is often called "the eyes of Industry 4.0" [15], uses ML algorithms embedded in advanced cameras to enable smart systems' capability to replace human vision in controlling manufacturing processes and, therefore, improving product and process monitoring through inspection [16]. Machine vision tools enhanced with the ML-driven image processing algorithms have a huge application potential in Industry 4.0 [17] as they are capable of identifying products, objects, components and materials being handled through the manufacturing process and making certain decisions as needed. Images may contain sensitive or private information and, therefore, PPML concerns are also applied to images. However, specifics of image data representation (values for the features are hidden behind the pixels) require special approaches towards encryption and synthetic image generation for PPML [18], [19], [20].

In [21], the homeomorphic data space transformation algorithm has been developed as a kind of homomorphic encryption. Its specificity is that the anonymization of a sensitive dataset is performed by a secret neural network with weights as kind of encryption keys. This algorithm worked well with numeric tabular data to protect its privacy and enabling effective supervised ML (classification with deep neural networks) on top of encrypted data. In this paper,

we update and extend the application scope of this method also towards sensitive image data. In addition, we show that this algorithm can be combined with synthetic image generation, aiming to create a hybrid privacy protection method with the high level of security.

The rest of the paper is organized as follows: Section 2 presents two PPML approaches (homeomorphic data space transformation and synthetic image generation) and recommends using them together as a hybrid; Section 3 updates (by including convolutional autoencoders and pre-trained feature extractors into the loop) the PPML algorithms to be applied to image data; Section 4 summarizes the corresponding experiments; and we conclude in Section 5.

2. Privacy-Preserving Supervised ML

The generic and simplified schema of supervised ML (for classification problem) is as follows.

We have as an input:

Data (m samples of private data): $d_1, d_2, \dots, d_m$ with *distribution* D;

Data sample (n-dimensional vector of features): d_i: $\{f_1, f_2, \dots, f_n\}$ within feature *space* F^n;

Labels (k class labels): $l_1, l_2, \dots, l_k$;

Labelled Data sample: $\langle d_i, l_j \rangle = \{f_1, f_2, \dots, f_n \rightarrow l_j\}$, (i.e., Data sample d_i is assigned to the class l_j by some expert/supervisor);

Supervised ML is a function:

ML (*training* subset of Labelled Data samples) → MODEL; such that

MODEL (Data sample d_i from *test* subset of Labelled Data samples with hidden Label) → l_j ; (automatic labelling of test data samples, comparing with the hidden labels, loss evaluation and model's performance assessment).

As we have mentioned in the previous section, there are two basic options (homomorphic encryption and synthetic data generation) to care about the dilemma "privacy of data vs. value from data" or the PPML.

2.1. Homeomorphic transformation as a kind of homomorphic encryption

As an example of the first PPML option (homomorphic encryption), we consider the recently developed [21] homeomorphic (topology-preserving) data space transformation (see Fig. 1).

Such encryption (Fig. 1) can be applied to tabular numeric data. As a result, the private data samples will change their original locations due to the topologically modified (using secret function) data space. After the encryption, the further ML process (e.g., deep learning with neural networks) can be performed remotely on MLaaS public cloud and will result in the encrypted model, which can be decrypted and used later in a safe place when necessary. A simplified formal model of the homeomorphic data space transformation is as follows:

Secured Data: m samples of data $(\tilde{d}_1, \tilde{d}_2, \dots, \tilde{d}_m)$ with *distribution* $\tilde{\boldsymbol{D}}$, which are the result of secret, locked by the *KEY* homeomorphic (i.e., topology-preserving) transformation (defined by an invertible function *TOP_TRANS*, in which *KEY* is a set of secret parameters), of the feature space of the original (private) data $(d_1, d_2, \dots, d_m)$.

Important properties of a homeomorphic transformation:

$\tilde{d}_i$: $\{\tilde{f}_1, \tilde{f}_2, \dots, \tilde{f}_n\} = TOP_TRANS(KEY, d_i: \{f_1, f_2, \dots, f_n\})$;

d_i: $\{f_1, f_2, \dots, f_n\} = TOP_TRANS\ (KEY^{-1}, \tilde{d}_i: \{\tilde{f}_1, \tilde{f}_2, \dots, \tilde{f}_n\})$;

ML (training set: $\{\forall i \langle \tilde{d}_i, l_j \rangle\}$) → SECURED_MODEL; such that

SECURED_MODEL (secured test data sample $\tilde{d}_i$) → l_j ← MODEL (test data sample d_i).

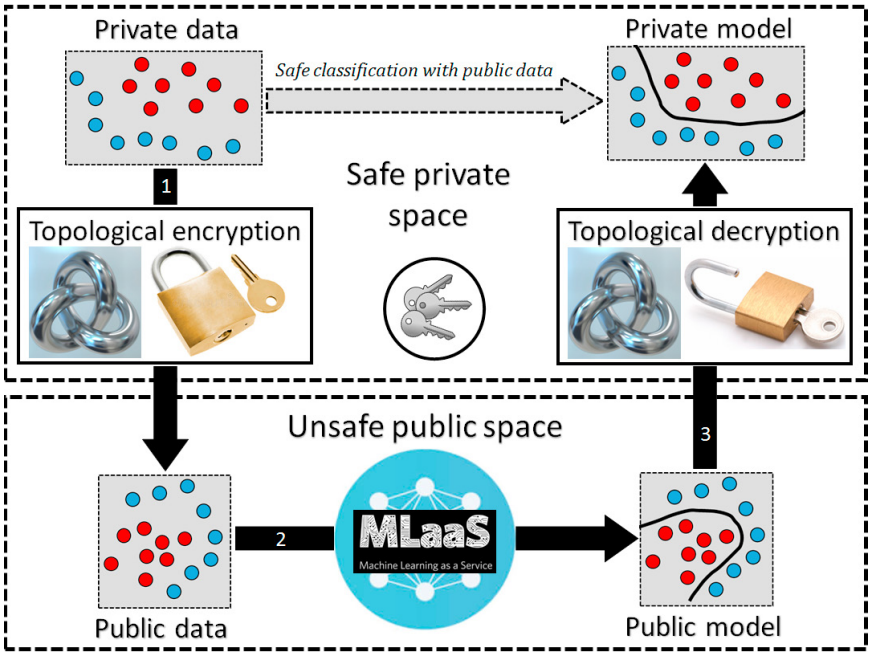

Fig. 1. Generic schema of supervised PPML based on homeomorphic data space transformation (topology-preserving encryption).

The complex Fig. 2 will be used to show: the example of homeomorphic transformation; the example of synthetic data generation; and the hybrid of these both. Particularly for the needs of this subsection, in Fig. 2 (a, b, c, d), an example of a transformation is shown for a 2D data space and three class labels ("red", "green" and "blue" data samples). The transformation from the original data (Fig. 2 (a)) to the secured data (Fig. 2 (d)) is actually also transforming the original (private) data distribution (hidden model or decision boundaries between three classes as shown in Fig. 2 (b)) into the new data distribution (secured model) as shown in Fig. 2 (c).

The advantages of such a homeomorphic transformation for PPML are as follows:

1) Uncovering (or guessing) any sample of the original private data $(d_1, d_2, \dots, d_m)$ from $(\tilde{d}_1, \tilde{d}_2, \dots, \tilde{d}_m)$ without knowing the KEY is unfeasible (for a properly defined transformation function);
2) There is no need to keep anywhere the original private data $(d_1, d_2, \dots, d_m)$ after the transformation (it can be deleted);
3) Machine learning process over the labelled secured training data could be performed within remote and potentially unsecure premises of the MLaaS providers and the trained secured models could be kept there and securely queried when needed within their clouds;
4) After the model is built and in use, there is no need to keep the secured data $(\tilde{d}_1, \tilde{d}_2, \dots, \tilde{d}_m)$ anymore and it can be deleted;
5) The transformation could be potentially (with some effort discussed in the following sections) adapted to the private image data transformation (i.e., making the homeomorphic transformation on the level of the latent feature vectors of the images).

The potential drawbacks of the homeomorphic transformation are as follows:

A. The stronger ("deeper") the transformation KEY would be, the longer the machine learning process could be needed to capture an effective (secured) classification model out of the secured data. For example (in terms of Deep Learning), the deeper the KEY (distributed sequence of "locks"), the deeper architecture of a neural network (as a potential classifier) and more training time will be needed to achieve the inteded accuracy;
B. If all the distributed components (unlikely but anyway) of the KEY leak and become known to the MLaaS provider (which also gets secured data to build the model), then the original private data could be uncovered.

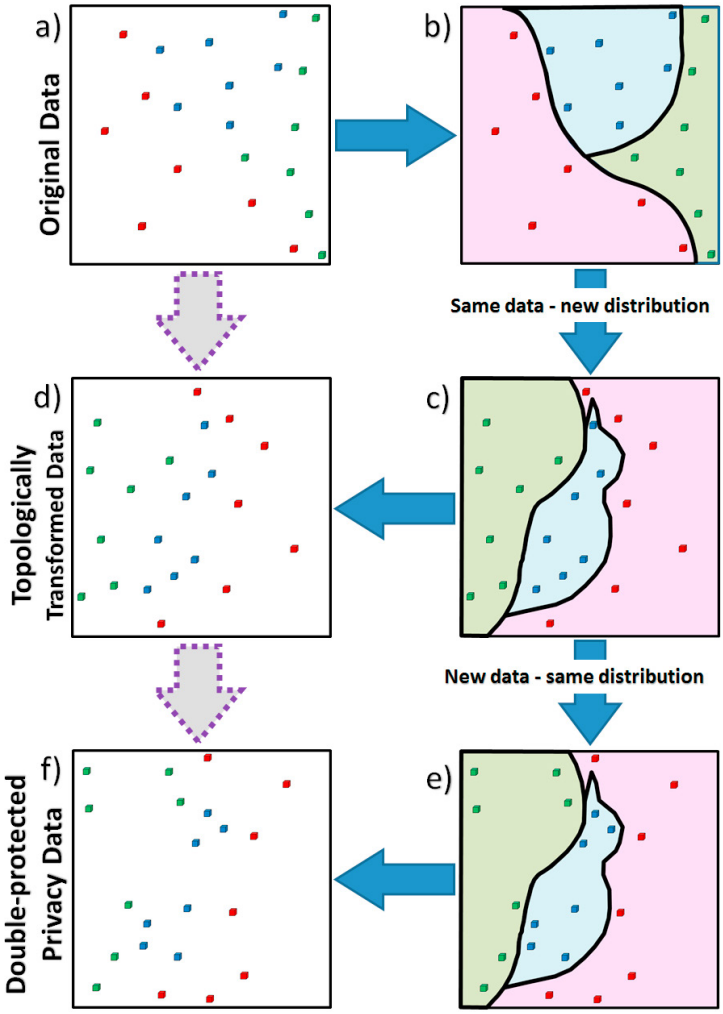

Fig. 2. Illustration of the homeomorphic data space transformation and synthetic data generation processes and their sequential use as a hybrid PPML process: (a) example of the original (private) data labelled into three classes ("green", "blue" and "red") in 2D space; (b) distribution of the original data is illustrated by the decision boundaries between the classes, which could be potentially learned by some ML algorithm; (c) the result of the homeomorphic data space transformation, which changes the distribution of the original data, relocates each data sample and corresponding decision boundaries; (d) shows the original data samples but relocated due to homeomorphic transformation for privacy reasons; (e) the synthetic data generation is illustrated, which preserves the distribution of data (i.e., creates new data samples for each class in new locations but not changing the decision boundaries); (f) the result of synthetic data generation is shown, which guarantees the double protection of the original private data (a) due to relocation of private samples and then replacing them with the arbitrary number of synthetic samples.

2.2. Synthetic data generation to enable PPML

As we have mentioned above, the second PPML option would be a synthetic data generation. A simplified formal model of the synthetic data generation process has the following input:

Data (m samples of private data): $d_1, d_2, \dots, d_m$ (with *distribution* D) labelled to k classes $l_1, l_2, \dots, l_k$ so that we have:

m_1 samples (with distribution D_1) labelled as l_1;
m_2 samples (with distribution D_2) labelled as l_2;
…

m_k samples (with distribution D_k) labelled as l_k,
where: $m = m_1 + m_2 + \cdots + m_k$.

The task of the synthetic data generation process is as follows:

Generate $\bar{m}$ new samples of synthetic data: $\bar{d}_1, \bar{d}_2, \ldots, \bar{d}_{\bar{m}}$ (within the same *distribution* D) labelled to k classes $l_1, l_2, \ldots, l_k$ so that we have:

$\bar{m}_1$ new samples labelled as l_1 (within the same distribution D_1 as the original m_1 samples with the same label);

$\bar{m}_2$ new samples labelled as l_2 (within the same distribution D_2 as the original m_2 samples with the same label); …

$\bar{m}_k$ new samples labelled as l_k (within the same distribution D_k as the original m_k samples with the same label); … where: $\bar{m} = \bar{m}_1 + \bar{m}_2 + \cdots + \bar{m}_k$.

The main requirements for the synthetic data generation process are as follows:

The classification $\overline{MODEL}$ built (trained) from the set of synthetic data $\bar{d}_1, \bar{d}_2, \ldots, \bar{d}_{\bar{m}}$ will classify the new test samples similarly (as much as possible) as the $MODEL$, which could be built from the set $d_1, d_2, \ldots, d_m$ of the original [private] data, i.e.:

MACHINE LEARNING (training set: $\{\forall i \langle d_i, l_j \rangle\}$) $\rightarrow MODEL$;

MACHINE LEARNING (training set: $\{\forall i \langle \bar{d}_i, l_j \rangle\}$) $\rightarrow \overline{MODEL}$;

$\overline{MODEL}$ (test data sample d_t) $\rightarrow l_q \leftarrow MODEL$ (test data sample d_t).

In Fig. 2 (c, d, e, f), an example of synthetic data samples' generation for the three classes ("red", "green" and "blue") is shown for 2D data space. Assume that the original data is the one which is shown on the screen in Fig. 2 (d) and its distribution is shown in Fig 2 (c). The new synthetic data samples generation process will result in the renewed set of data samples like it is shown in Fig. 2 (f), which have the same distribution (Fig. 2 (e) is the same as Fig. 2 (c)) as the original data from Fig. 2 (d) has.

The advantages of such synthetic data generation for PPML are as follows:

1) There is no one-to-one (neither explicit nor implicit) relationship between any sample d_i of the original [private] data and any sample $\bar{d}_j$ of the newly generated [synthetic] data;
2) There is no need to keep anywhere the original private data $(d_1, d_2, \ldots, d_m)$ after the new synthetic dataset(s) is (are) generated (i.e., private data can be deleted);
3) After the model is built from the synthetic data and in use, there is no need to keep the synthetic data $(\bar{d}_1, \bar{d}_2, \ldots, \bar{d}_{\bar{m}})$ anywhere as well, and it can be deleted.

The drawbacks of synthetic data generation are as follows:

A. If the classification model is going to be trained within potentially unsecure premises of the MLaaS providers, then it is important that synthetic data samples are located (within the feature space) far enough from any of the original [private] samples so that the private entities cannot be identified. It is very difficult (if possible at all) to keep this requirement when we have a multidimensional decision space (n is large) and we do not have enough original samples within some classes (m_i is small). In such cases (to guarantee at least some reasonable generalization performance of the model trained from synthetic data), the newly generated samples may appear very close to the original samples, violating their privacy (especially if various modifications of (conditional) GANs are used. However, also the differential privacy approaches may suffer);
B. The trained classification model cannot be safely kept and queried within potentially unsecure premises of the MLaaS providers because each query itself exposes the private entity;
C. It is difficult to adapt easily to the [private] image datasets due to the (A)-concern.

2.3. The hybrid approach to PPML

As we can see, both PPML approaches have their different pros and cons. Therefore, we suggest combining these two: transformation (i.e., same data - different distribution) and generation (i.e., same distribution – different data) processes into one hybrid process so that it will have minimal drawbacks but, in the same time, will inherit advantages of both. Fig. 6 as a whole illustrates such a hybrid, where first the homeomorphic data transformation is performed (i.e., original data samples change their locations together with the potential decision boundaries) and then new synthetic data samples are generated so to fit the new distribution of the data. Evidently, such a hybrid process will neither have the (B) drawback of the homeomorphic data transformation process applied alone nor the (A), (B) and (C) drawbacks of the synthetic data generation process applied alone. Therefore, such a hybrid, which inherits the advantages of both PPML options, could potentially provide an extremely high level of privacy protection of various data and of the models learned out of it (even within the unsafe MLaaS clouds), and, at the same time, enable high generalization performance of the models (both individual and federated).

3. Approaching image data with PPML

Digital images are made up of pixels, and the features of an image (needed for processing with ML algorithms like ordinary numeric tabular data) are not explicit within the image representation. Therefore, the straightforward use of the PPML processes described above over the image data is not possible. Certain image pre-processing will be needed to enable both homeomorphic image data transformation and synthetic data generation.

To enable homeomorphic image data transformation, we suggest using the autoencoder concept and its various modifications. An autoencoder is a type of deep learning network that is trained to replicate its input (usually image) data. A basic or simple autoencoder consists of two networks: Encoder and Decoder (see Fig. 3 (a)). The training process is organized so that the Encoder (having some image as an input) learns a set of hidden-within-image features, shown as a latent vector in Fig. 3 (a). Simultaneously, the Decoder is trained to reconstruct the original image based on the latent vector. The more image-generation-specific option of an autoencoder, which is often used as an alternative to popular Generative Adversarial Networks (GANs), is a variational autoencoder [22] shown in Fig. 3 (b). The basic idea behind the image generation process with the variational autoencoders assumes the development of a low-dimensional latent vector space of representations (see Fig. 3 (b)) where each point can be mapped to a realistically looking image. The role of such mapping (latent vector – grid of pixels for an image) is played by the Decoder. A similar basic objective with certain specifics could be achieved also with the convolutional autoencoders [23], adversarial autoencoders [24], attention autoencoders [25], hybrids of autoencoders with GANs [26], etc.

In this study, we assumed that, if an image can be encoded by such an autoencoder, which can also extract features of an image needed for future classification (e.g., convolutional autoencoder [23]), into the corresponding latent vector, then further PPML manipulations with such vectors (homeomorphic encryption, synthetic image generation) could be performed within the latent (feature) space in a similar way like with other tabular data.

Consider the process of homeomorphic image data transformation driven by a convolutional autoencoder in Fig. 4. First, within a safe place, each private image goes through an autoencoder (i.e., feature extractor) architecture (upper part of the figure) until the Decoder will be trained enough to achieve an acceptable reconstruction quality. This would also mean that the Encoder becomes capable of giving an acceptable latent vector representation of the image, which also includes the extracted features needed for potential classification tasks. After that, the homeomorphic latent space transformation (as a kind of homomorphic encryption) is being performed as it is described in Subsection 2.1. Such transformation outputs the encrypted (with secret keys) latent vector representation of the original private image. Finally, if to apply the already trained (at the first stage) Decoder to that encrypted latent vector, then some new image will be reconstructed, which can be considered as an encrypted version of the original image (see lower part of Fig. 4). Such an encrypted image is supposed to hide all private information from the original image, which cannot be uncovered without knowing secret keys. As described in [21], such encryption-decryption keys are the secret weights of a (deep) neural network used for homeomorphic transformation of the data space.

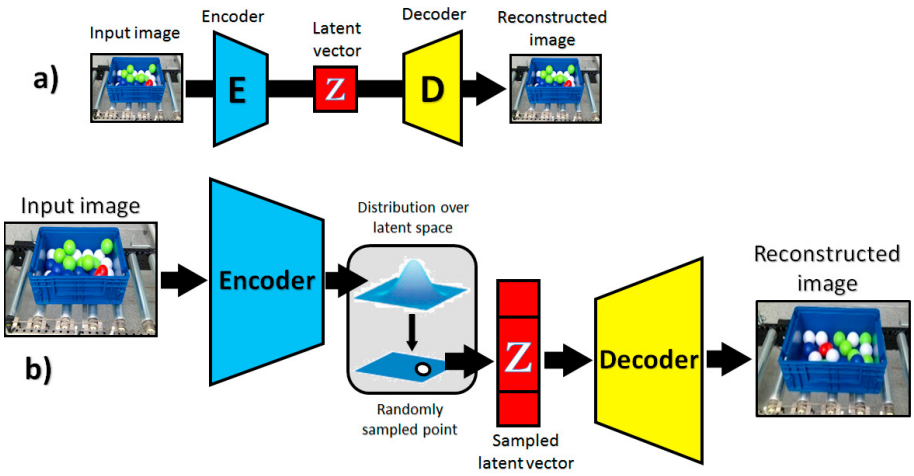

Fig. 3. Autoencoders: (a) the basic (e.g., convolutional) autoencoder; (b) [convolutional] variational autoencoder.

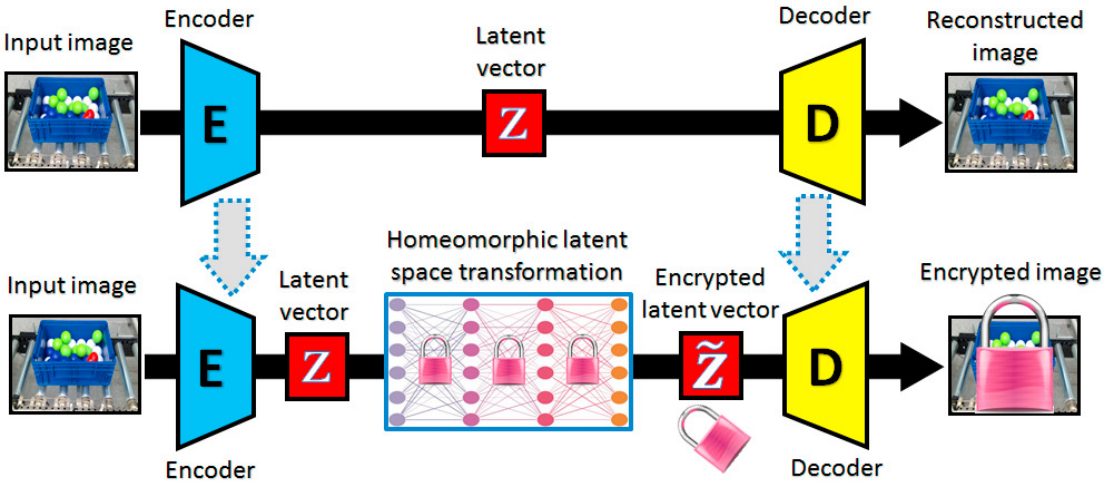

Fig. 4. The homeomorphic latent space transformation as a PPML approach to encrypt the images using a convolutional autoencoder.

Synthetic image data generation can be arranged using, e.g., convolutional variational autoencoders (www.tensorflow.org/tutorials/generative/cvae) and randomly sampled points within the latent space distribution as shown in Fig. 3 (b).

Finally, the safest option would be a hybrid PPML process (see Subsection 2.3) where we first encrypt the latent space using homeomorphic latent space transformation and then generate new images within the encrypted space.

If there is no need to visualize the images in encrypted form, one can use more powerful (than convolutional autoencoders) pre-trained feature extractors and use the extracted and encrypted feature vectors for combining privacy protection with good classification accuracy.

4. Experiments

In order to prove the concept of hybrid PPML (encryption plus generation) with real data from cyber-physical environments of Industry 4.0, we use the capacities of a logistics system and particularly of an interroll cassette conveyor. These industrial installations are related to the project IMMUNE: “Cyber-Defence for Intelligent Systems”, which is a NATO SPS project (http://recode.bg/natog5511). The conveyor is used for simulating various critical or

adversarial situations with military cargo, supplies or luggage delivered by air and for automated video inspection at the security checkpoints. The decisions about the distribution of the cassette loads (bags, boxes, packages, etc.) are made automatically by AI/ML components [27] trained using labeled images. Such artificial "airport workers" assess in real time each load's safety, aiming to prevent any potential danger caused by the items in the load. Needed models for artificial decision-makers are built and constantly updated using remote MLaaS and, therefore, privacy of images is an issue here. We experimented with 2198 labelled images of cassettes taken by the cameras installed on the critical distribution points of the conveyer. We applied a supervised ML to build a load classification model (a convolutional neural network) from the image data before and after the encryption (homeomorphic latent space transformation using convolutional autoencoders or feature extractors). The objective was to compare how the level (depth) of encryption influences the quality of a learned classifier. The results show that the loss of classification accuracy due to the PPML measures is within the 1-3% interval depending on the depth of the transformation, which is an acceptable loss.

The experimental settings of the described logistics laboratory enable us to simulate various types of attacks [28] on the backbone of AI algorithms, including attempts to decrypt the secured images. We have checked the robustness of the homeomorphic data space transformation as an encryption algorithm and have shown that, even after a simulated leakage of >90% of the original private images together with the encrypted model, the rest 10% of secured images cannot be uncovered (reconstructed to disclose the private information).

Below we give a few illustrative examples of our experiments with the less sensitive to disclosure, public and popular image dataset Kaggle (Cats and Dogs Dataset: www.microsoft.com/en-us/download/details.aspx?id=54765), which contains 50K colored images of various cats and dogs.

Fig. 5 illustrates the result of applying the convolutional (driven by the simplest from the VGG-11 family of neural architectures: www.kaggle.com/datasets/pytorch/vgg11) autoencoder for image encryption via homeomorphic transformation of the latent space. One can see that the private information from the original image can be safely hidden. Classification with the images encrypted in this way is possible yet gives lower classification accuracy comparably to the pre-trained feature extractors.

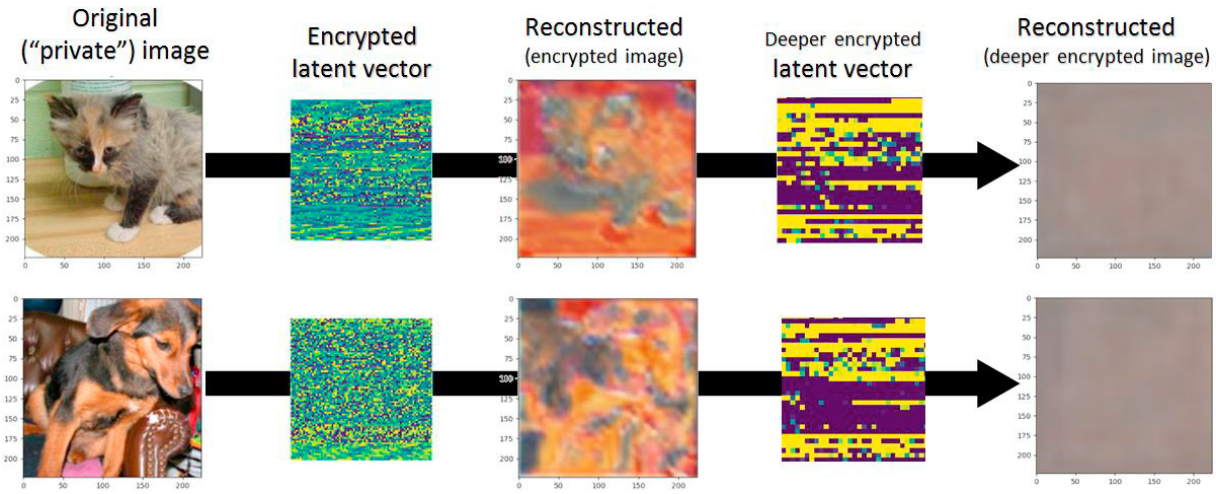

Fig. 5. Example of the homeomorphic latent space transformation and the encrypted image visualization using a convolutional autoencoder.

For the experiments with the much more powerful (than VGG-11) and pre-trained feature extractors, we have chosen the ResNet architecture [29] according to the configuration ResNet-18, see, e.g. [30], which has been pretrained for the Kaggle (www.kaggle.com/datasets/pytorch/resnet18). Extracted and encrypted feature vectors for Kaggle dataset have been used for image classification and show better classification accuracy comparably to the feature vectors extracted by the convolutional autoencoder, as can be seen from Table 1. One can see that there is almost no accuracy loss within the classification tasks in safe (three layers of deep encryption) dataset comparably to the original dataset. Therefore, if there is no need to visualize and show the encrypted images, then the feature extractors are the better choice to assist PPML with the homeomorphic transformation algorithm.

The source code and more details related to the Kaggle dataset encryption experiments together with the corresponding plots and figures can be found in https://github.com/Adversarial-Intelligence-Group/image-encryption.

Table 1. Comparisons of classification accuracy over the original and the encrypted Kaggle dataset images.

Homeomorphic transformation (encryption): assistive model for latent (feature) vector extraction	Classification accuracy before the encryption	Classification accuracy after the encryption
Convolutional (VGG-11) autoencoder	76.12 %	74.89 %
ResNet feature extractor [29], [30]	97.92 %	97.34 %

5. Conclusions

Lots of private data collected to be processed by various MLaaS to get hidden value out of it for the needs of industry 4.0 makes the domain of PPML an emergent one. The algorithms, which are able to secure private data so that ML will still be able to capture the needed value (diagnostic, prediction, etc., models) are of great interest nowadays. Two orthogonal options for the PPML are homomorphic encryption and synthetic data generation. Image data (compared to the traditional tabular data) has certain specifics (features of images are hidden behind the pixels), which is the challenge for the mentioned PPML options. In this study, we have expanded the scope of the homeomorphic data space transformation algorithm [21] to be also applied to image data. Privacy of an image is achieved by special encryption of the correspondent latent (feature) vector, obtained using convolutional autoencoders or more powerful pre-trained feature extractors. Such a PPML approach guarantees security of private data in the images and, at the same time, does not exclude the possibility to apply ML on top of the encrypted images. In this paper, we also show that combining a homeomorphic data space (latent vector space for the images) transformation with a synthetic data generation (via convolutional variational autoencoders for the images) into one hybrid PPML process will essentially improve privacy. This is due to the fact that such a hybrid inherits the advantages of both components and eliminates most of the shortcomings of each one. For example, such a hybrid is capable of generating the well-protected synthetic images even on the basis of a relatively small number of the available original data samples, which is a problem if you apply the synthetic image generation alone without the encryption phase. Also, the leakage of the encryption keys, which is a problem with the encryption algorithm applied alone, will not be a problem for a hybrid due to the additional (synthetic image generation) PPML stage.

The manufacturing domain is constantly facing a wide range of problems related to cybersecurity, and encryption is one of the most important ways to protect manufacturers from cyberattacks [31]. Furthermore, deep learning models are still not resistant to attacks and can leak important information to an attacker. Standard encryption algorithms like AES (Advanced Encryption Standard) or RSA (Rivest, Shamir, and Adleman) have one (symmetric encryption) or two (asymmetric encryption) keys to encrypt and decrypt data [32], and here comes another problem of storing securely the private keys. Our algorithm resolves these problems by having the encrypted ML models and synthetic data. One example of using synthetic data in manufacturing could be video surveillance, activity monitoring and recognition, anomaly detection, diagnostics and predictive maintenance, etc. ML trains such models based on images, which may include secret objects, processes or individuals, while use of synthetic data will make the ML models GDPR compliant. We conducted our experiments with simple images to demonstrate the advantages of using our algorithm over standard encryption algorithms in MLaaS, but it can be applied to any image dataset.

The source code and more additional illustrations for some of our image encryption experiments are available online at https://github.com/Adversarial-Intelligence-Group/image-encryption.

References

[1] Gourisaria, M. K., Agrawal, R., Harshvardhan, G. M., Pandey, M., and Rautaray, S. S. (2021). “Application of machine learning in Industry 4.0”. In: *Machine Learning: Theoretical Foundations and Practical Applications* (pp. 57–87). Springer, Singapore.

[2] Rai, R., Tiwari, M. K., Ivanov, D., and Dolgui, A. (2021). “Machine learning in manufacturing and industry 4.0 applications”. *International Journal of Production Research*, **59(16)**: 4773–4778.

[3] Onik, M. M. H., Kim, C.-S., and Yang, J. (2019). “Personal data privacy challenges of the fourth industrial revolution”. In: *Proceedings of the 21st International Conference on Advanced Communication Technology* (pp. 635–638). IEEE.

[4] Longo, F., Nicoletti, L., and Padovano, A. (2019). ”Modeling workers’ behavior: A human factors taxonomy and a fuzzy analysis in the case of industrial accidents”. *International Journal of Industrial Ergonomics*, **69**: 29-47.

[5] Hossayni, H., Khan, I., and Crespi, N. (2021). “Privacy-preserving sharing of industrial maintenance reports in Industry 4.0”. In: *Proceedings of the Fourth IEEE International Conference on Artificial Intelligence and Knowledge Engineering* (pp. 17–24). IEEE.

[6] Tanuwidjaja, H. C., Choi, R., Baek, S., and Kim, K. (2020). “Privacy-preserving deep learning on machine learning as a service—a comprehensive survey”. *IEEE Access*, **8**:167425-167447.

[7] Famulari, A., Longo, F., Campobello, G., Bonald, T., and Scarpa, M. (2014). “A simple architecture for secure and private data sharing solutions”. In: *Proceedings of the IEEE Symposium on Computers and Communications* (pp. 1-6). IEEE.

[8] Rechberger, C., and Walch, R. (2022). “Privacy-preserving machine learning using cryptography”. In: *Security and Artificial Intelligence* (pp. 109-129). Springer, Cham.

[9] Xu, R., Baracaldo, N., and Joshi, J. (2021). “Privacy-preserving machine learning: Methods, challenges and directions”. *arXiv preprint arXiv:2108.04417*.

[10] Liu, J., and Meng, X. (2020). “Survey on privacy-preserving machine learning”. *Journal of Computer Research and Development*, **57(2)**: 346-362.

[11] Xu, R., Joshi, J. B., and Li, C. (2019). “Cryptonn: Training neural networks over encrypted data”. In: *Proceedings of the 39th IEEE International Conference on Distributed Computing Systems* (pp. 1199-1209). IEEE.

[12] Abay, N. C., Zhou, Y., Kantarcioglu, M., Thuraisingham, B., and Sweeney, L. (2018). “Privacy preserving synthetic data release using deep learning”. In: *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (pp. 510-526). Springer, Cham.

[13] Xin, B., Geng, Y., Hu, T., Chen, S., Yang, W., Wang, S., and Huang, L. (2022). “Federated synthetic data generation with differential privacy”. *Neurocomputing*, **468**: 1-10.

[14] Hernandez, M., Epelde, G., Alberdi, A., Cilla, R., and Rankin, D. (2022). “Synthetic data generation for tabular health records: a systematic review”. *Neurocomputing*, **493(7)**: 28-45.

[15] Coffey, V. C. (2018). “Machine vision: the eyes of Industry 4.0”. *Optics & Photonics News,* **29(7)**: 42–49.

[16] Silva, R. L., Canciglieri Junior, O., and Rudek, M. (2022). “A road map for planning-deploying machine vision artifacts in the context of industry 4.0”. *Journal of Industrial and Production Engineering*, **39(3)**: 167–180.

[17] Penumuru, D. P., Muthuswamy, S., and Karumbu, P. (2020). Identification and classification of materials using machine vision and machine learning in the context of industry 4.0. *Journal of Intelligent Manufacturing*, **31(5)**: 1229–1241.

[18] Xue, H., Liu, B., Din, M., Song, L., and Zhu, T. (2020). “Hiding private information in images from AI”. In: *Proceedings of the IEEE International Conference on Communications* (pp. 1-6). IEEE.

[19] Webster, R., Rabin, J., Simon, L., and Jurie, F. (2021). “Generating private data surrogates for vision related tasks”. In: *Proceedings of the 25th International Conference on Pattern Recognition* (pp. 263-269). IEEE.

[20] Chen, Y., Ping, Y., Zhang, Z., Wang, B., and He, S. (2021). “Privacy-preserving image multi-classification deep learning model in robot system of industrial IoT”. *Neural Computing and Applications*, **33(10)**: 4677-4694.

[21] Girka, A., Terziyan, V., Gavriushenko, M., and Gontarenko, A. (2021). “Anonymization as homeomorphic data space transformation for privacy-preserving deep learning”. *Procedia Computer Science*, **180**: 867-876.

[22] Kingma, D. P., and Welling, M. (2013). “Auto-encoding variational bayes”. *arXiv preprint arXiv:1312.6114*.

[23] Guo, X., Liu, X., Zhu, E., and Yin, J. (2017). “Deep clustering with convolutional autoencoders”. In: *Proceedings of the International Conference on Neural Information Processing* (pp. 373-382). Springer, Cham.

[24] Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., and Frey, B. (2015). “Adversarial autoencoders”. *arXiv preprint arXiv:1511.05644*.

[25] Oluwasanmi, A., Aftab, M. U., Baagyere, E., Qin, Z., Ahmad, M., and Mazzara, M. (2021). “Attention autoencoder for generative latent representational learning in anomaly detection”. *Sensors*, **22(1)**: 123.

[26] Ahmad, B., Sun, J., You, Q., Palade, V., and Mao, Z. (2022). “Brain tumor classification using a combination of variational autoencoders and generative adversarial networks”. *Biomedicines*, **10(2)**: 223.

[27] Golovianko, M., Gryshko, S., Terziyan, V., and Tuunanen, T. (2021). ”Towards digital cognitive clones for the decision-makers: adversarial training experiments. *Procedia Computer Science*, **180**: 180-189.

[28] Terziyan, V., Golovianko, M., Branytskyi, V., Malyk, D., and Gryshko, S. (2021). “Implementing and Training the Immune System (Attack Detector)”. *Technical Report* (Annex 3, Deliverable 3, pp. 1-19). NATO SPS G511 Project IMMUNE: “Cyber-Defence for Intelligent Systems”. http://recode.bg/natog5511/wp-content/uploads/2021/05/ANNEX-3-Deliverable-3_NURE-and-JYU-G5511.pdf

[29] He, K., Zhang, X., Ren, S., and Sun, J. (2016). “Deep residual learning for image recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770-778).

[30] Wang, S., Xia, X., Ye, L., and Yang, B. (2021). “Automatic detection and classification of steel surface defect using deep convolutional neural networks”. *Metals*, **11(3)**, 388.

[31] Khanezaei, N., and Hanapi, Z. M. (2014). “A framework based on RSA and AES encryption algorithms for cloud computing services”. In: *Proceedings of the IEEE Conference on Systems, Process and Control* (pp. 58-62). IEEE. https://doi.org/10.1109/SPC.2014.7086230

[32] Thames, L., and Schaefer, D. (2017). “*Cybersecurity for Industry 4.0”*. Heidelberg: Springer.