ANNEX 3

Deliverable 3: Implementing and training the immune system (attack detector).

Implementation of the Components of the Artificial Immune System.

(Milestone 3)

V.Terziyan[1], M. Golovianko[2], V. Branytskyi[2], D. Malyk[2], S. Gryshko[3]

[1]Faculty of Information Technology, University of Jyvaskyla, FI
[2]Department of Artificial Intelligence, Kharkiv National University of Radio Electronics, UA
[3]Department of Economic Cybernetics and Management of Economic Security, Kharkiv National University of Radio Electronics, UA

1 Experimental setup

A cyber-physical environment is configured as an adversarial learning ecosystem. The physical component of the environment is provided by the logistic laboratory with camera-surveillance over the conveyors within the NATO SPS project "Cyber-Defence for Intelligent Systems" aiming at enhancing civil and military security infrastructures by digital security officers which are immune to adversarial attacks. The core digital component of the environment is provided by Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) with a human-operator as a decision-making trainer, a digital officer (clone) as a trainee (a discriminator) and a smart digital adversary as a challenger (generator of sophisticated decision situations, emergencies and attacks supposedly accelerating the cloning process). The digital workers (powerful convolutional classifiers) "observe" the interroll cassette conveyer, an analogy of those used in airports for luggage distribution and inspection. Their task is to prevent any potential danger caused by the loads of objects in the cassettes on the conveyer applying personal judgement and expertise of the "cloned" security experts.

The adequacy of the experiments and the results` interpretability are highly influenced by the quality and the size of the training datasets. Different image types require different processing methods. Therefore, the experiments were conducted with different data collections – the ones produced within the cyber-physical environment configured for the purpose of the current research, and well-known high-quality datasets for computer vision and deep learning, such as, MNIST (Deng, 2012).

The models` performance is evaluated based on the prediction accuracy calculated as the percentage of the number of the correct predictions (decisions) out of the total number of predictions. This evaluation, however, is not enough to decide on the appropriateness of an artificial worker as a replacement for the human. With this aim we also find important to measure (i) the

correlation between an artificial worker and a human expert indicating the percentage of the matching decisions, (ii) the actual correctness of the decisions by human experts calculated as the classification accuracy and (iii) the correctness of the human experts after he/she knows the decision made by an artificial worker.

2 The essence of experiments

Techniques based on adversarial content generation can increase both the accuracy and the robustness of intelligent models. Artificial generators, particularly GANs, act as providers of either (i) new challenging conditions for a trainee: a model (Discriminator) is then trained in a minimax game by a confrontation with a strong, constantly evolving, hardly predictable artificial adversary (Generator), or (ii) new challenging training content (sophisticated domain-specific data augmentation): a model is retrained in adversarial training on artificially generated or perturbed adversarial examples.

Adversarial training is seen as a form of active learning based on selective sampling, or selection of the most informative or representative instances reflecting personal features (biases) of decision-making. In active learning, models can request labels on new samples from a heuristic labeller replacing the human expert. However, measuring the training value of examples is still a subject of research, and only a few selection criteria have been proposed (Hjelm et al., 2017; Weinstein et al., 2019). Various studies prove that the most informative data samples lie close to the decision boundary (Ertekin et al., 2007; Dasgupta et al., 2008).

Knowledge of samples populating decision space close to the decision boundary is primarily used for learning more accurate classification models. For instance, Zhou et al. (2020) suggest a new learning algorithm which improves the classification performance of neural networks by strengthening boundary samples. Recent studies have shown that knowledge of local neighbourhoods around the decision boundary can also be a key to a variety of other tasks, far beyond basic classification. Among them explaining predictions of complex machine learning models, such as those based on deep neural networks (Vlassopoulos et al., 2020), interpreting the generalization error of these models, and their robustness to adversarial attacks (Yousefzadeh et al., 2019), deepen knowledge about behaviour of decision-making systems (Karimi et al., 2019), knowledge distillation (Heo et al., 2019). In our research we apply it to the tasks of cognitive cloning and intelligent model security (reference removed for double-blind review).

That`s why all the experiments are divided into two groups:

1. The first one corresponds to the development of the new GAN architectures applicable for the cloning and the security tasks.
2. The second group aims at validating the methods for adversarial driven selective sampling based on decision boundary identification running detection, selection and generation of samples populating regions close to decision boundaries of pre-trained classifiers. Boundary samples are used for cognitive cloning simulations and decision-making behaviour modelling.

3 Adversarial driven selective sampling

We suggest new methods for adversarial-driven selective sampling for data augmentation in the tasks sensitive to the size and informativeness of training data. Our objective is to organize sampling in the most informative adversarial areas close to the decision boundary.

Boundary samples are data instances populating local neighbourhoods of intelligent model`s decision boundary.

In terms of this research, *adversarial sample*s are artificially created boundary samples populating feature space areas with the high probability of deep learning models misbehaviour (confusion).

The number and the size of such areas, ignorance zones (Terziyan et al., 2019), rises with the complexity of the targeted decision boundary (see Fig. 1.).

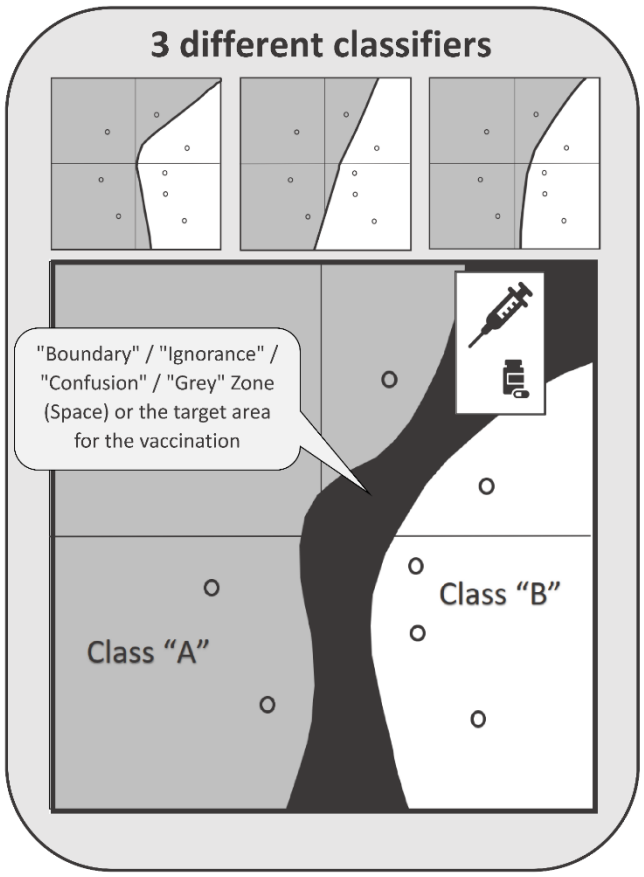

Fig. 1. The nature of the confusion zone along the decision boundary between two classes distributions

Well-identified adversarial samples act as border-guards of classes shaping the decision boundary (see Fig. 2).

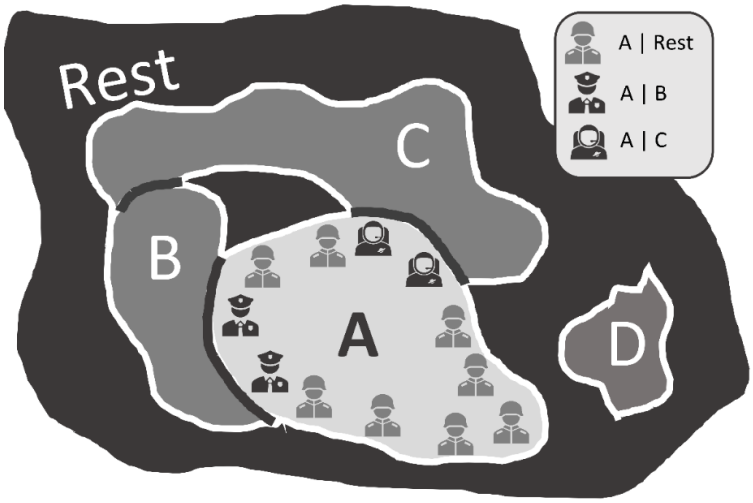

Fig. 2. Boundary samples aka "border-guards" shaping and securing the decision boundaries.

Adversarial-driven sampling learning is organized as a search of adversarial samples in ignorance zones. The generative model starts the search by learning distribution of boundary samples which

can be either (i) previously selected as a smaller representative subset from the training data, or (ii) crafted by overlaying some adversarial input on the available data instances (shifting them toward previously discovered decision boundaries, thus, manipulating data characteristics in the feature space).

3.1 Searching for boundary samples in data

For boundary samples selection we suggest a new, and computationally efficient algorithm of data relabelling based on the assumed proximity of data samples to various neighbouring classes. Unlike other selection methods, dealing with high dimensionality of feature space, this one is based on the analysis of the model decision space. Given that we have a basic multi-label classifier pretrained to generalize over the training data, the only information we need is unscaled probabilities in its output layer. The decision about the affiliation of each data sample to the boundary subset is made based on the confusion of the classifier. The higher is the confusion, the higher is the probability of the affiliation.

The overall idea is to divide mutually exclusive k classes {A, B, C, … , Z} of the decision space into k^2 subclasses {A|A, A|B, … , Z|Z} and relabel all data samples accordingly. For instance, a data sample of class A located in the neighbourhood of the boundary with class B gets label "A|B" according to the new partitioning (Fig. 2).

The relabelling process is implemented by a Confusion Classifier, a classifier with a specific output layer - confusion-layer (see Fig. 3).

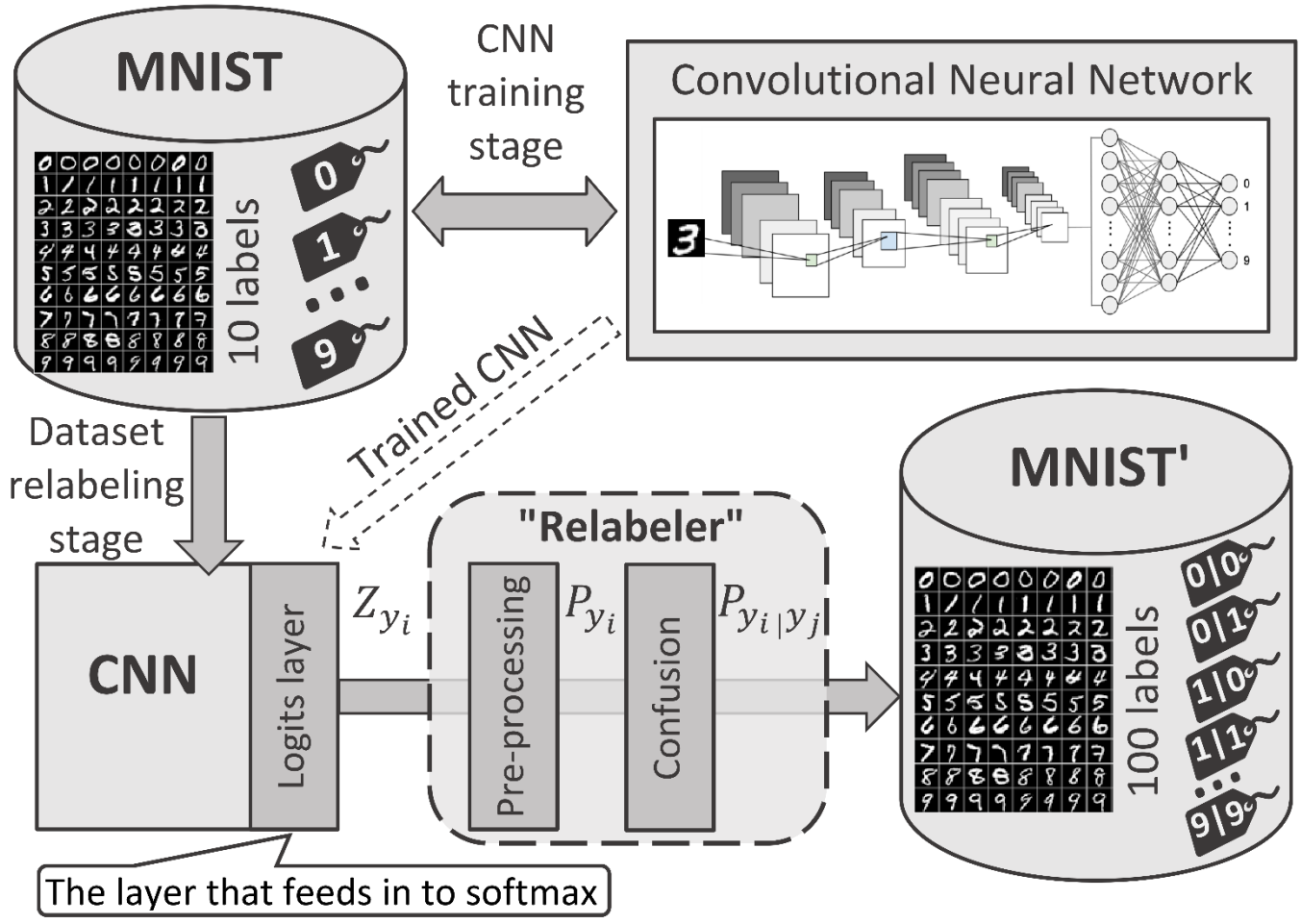

Fig. 3. The relabelling process implemented by Confusion Classifier.

The SoftMax activation function used traditionally for multiclass probability distribution in convolutional classifiers gives "winner takes all" effect, in many cases hindering the relabelling. We suggest several other functions for probability distribution, i.e., normalization, among them *Z-ScoreSoftMax*, *Z-Score-HardMax* and *Z-Score-HardSquareMax*.

As an input, the normalization layer takes k unscaled probabilities $Z_{y_1}, Z_{y_2}, \ldots, Z_{y_k}$ from the output (logits) layer of the neural network, where each value Z_{y_i} is defined on the interval (-∞,+ ∞) and corresponds to the y_i class. The normalization objective is to derive (normalized) probability distribution among the classes, so that:

$$P_{y_1} + P_{y_2} + \ldots + P_{y_k} = 1. \quad (1)$$

At first, the traditional Z-Score normalization is performed as follows:

$$\overline{Z_{y_i}} = \frac{Z_{y_i} - MEAN}{DEV}, \quad (2)$$

where $MEAN$ is the mean value calculated as:

$$MEAN = \sum_{r=1}^{k} \frac{Z_{y_r}}{k}, \quad (3)$$

and DEV is a standard deviation:

$$DEV = \sqrt{\frac{\sum_{r=1}^{k} (Z_{y_r} - MEAN)^2}{k}}. \quad (4)$$

Final normalization is performed by applying one of the following functions.

Z-Score-SoftMax:

$$P_{y_i} = \frac{e^{\overline{Z_{y_i}}}}{\sum_{r=1}^{k} e^{\overline{Z_{y_r}}}}. \quad (5)$$

Z-Score-HardMax:

$$P_{y_i} = \frac{\overline{Z_{y_i}} + M}{k \cdot M}. \quad (6)$$

Where M is calculated as follows:

$$M = \sum_{i=1}^{k} \left|\overline{Z_{y_i}}\right|. \quad (7)$$

And Z-Score-HardSquareMax:

$$P_{y_i} = \frac{\overline{Z_{y_i}} + 2}{2 \cdot k}. \quad (8)$$

3.1.1 Confusion-driven relabelling

According to one of two algorithms either basic confusion relabelling, or naïve confusion relabelling is performed.

Confusion-layer accepts samples with k class labels $(y_1, y_2, \ldots, y_k)$ and computes the k^2-dimensional output as follows: $P_{y_1|y_1}, P_{y_1|y_2}, \ldots, P_{y_k|y_k}$, so that:

$$P_{y_1|y_1} + P_{y_1|y_2} + \cdots + P_{y_k|y_k} = 1. \quad (9)$$

Each of the new labels $y_i|y_j$ corresponds to a subset of class y_i neighbouring with the class y_j. Label $y_i|y_i$ is assigned to the subset of y_i class samples, which are deep within the y_i class samples' distribution (not close to the boundary with other classes).

The relabelling calculations go as follows:

$$P_{\substack{y_i|y_j \\ (i\neq j)}} = \frac{P_{y_i} \cdot P_{y_j}}{1-P_{y_i}} \cdot TAX, \quad \text{IF}(\ P_{y_i} = 1\ ), \text{THEN}\ (P_{y_i|y_j} = 0\ ), \tag{10}$$

here:

$$TAX = 1 - V, \tag{11}$$

and

$$P_{y_i|y_i} = P_{y_i} \cdot V. \tag{12}$$

In case of basic confusion:

$$V = \sqrt{\frac{k}{k-1} \cdot \sum_{r=1}^{k} \left(P_{y_r} - \frac{1}{k}\right)^2}. \tag{13}$$

In case of naïve confusion:

$$V = \frac{k}{k-1} \cdot \sum_{r=1}^{k} \left(P_{y_r} - \frac{1}{k}\right)^2. \tag{14}$$

In Confusion layer V is an estimate of the standard deviation of class probabilities multiplied by the root of k (number of classes) to bring it to range [0, 1].

In Naïve Confusion layer V is an unbiased estimation of the standard deviation of class probabilities multiplied by k (number of classes) to bring it to range [0, 1]. In both cases, the mean value is assumed to be $\frac{1}{k}$.

$V \to 0$, if all the class probabilities are equal. $V \to 1$, if the probability for some class is 1 and others – 0.

$P_{y_i|y_i}$ is the probability that a particular sample belongs to class y_i and is located in the area, which is deep within the class y_i distribution.

$P_{y_i|y_j}$ is the probability that a particular sample belongs to class y_i distribution, but it is located in the area close to the class y_j distribution, i.e., the sample is close to the decision boundary between classes y_i and y_j from the side of class y_i. These calculations are based on two important components: $P_{y_i} \cdot P_{y_j}$ – the probability of simultaneous occurrence of the two independent events (the sample belongs to the distribution of class y_i and it also belongs to the distribution of class y_j); and $1 - P_{y_i}$ – the probability that the sample belongs to any other distribution but not of y_i class.

The more the deviation of the original probabilities is (i.e., clear winners, larger V and smaller TAX), the bigger share of the probabilities (in the new distribution) goes to the "deep-within class" label(s) of the winning class(es). On the contrary, the less is the deviation of the original probabilities (i.e., not clear winners, smaller V and larger TAX), the bigger share of the probabilities (in the new distribution) goes to the boundary class labels.

3.1.2 Multichannel relabelling

Another straightforward method of the relabelling is shown in Fig.4. According to this approach, the samples of each class in k classes (channels) are processed independently and synchronously. For each i-th channel, the dataspace is divided into two classes: i, containing all the samples of class i of the original dataset, and not-i (or $\bar{\imath}$), containing all the other samples. Every channel has a unique subset of samples, each containing only two classes. After that, the relabelling procedure described in 4.1.1 is applied to each channel independently, dividing decision space into k^2 subclasses: i; $\bar{\imath}$; $i|\bar{\imath}$; $\bar{\imath}|i$. In this way we find samples, which are:

- deep inside the distribution of the target class (i);
- deep inside the distribution of the union of all other classes ($\bar{i}$);
- boundary sample of class i ($i|\bar{i}$);
- boundary sample of class $\bar{\imath}$ ($\bar{\imath}|i$).

The goal is to extract boundary samples $i|\bar{\imath}$ for each channel. Thus, we get a boundary subset for of each class in k classes. The advantage of this approach is to split the complexity of the $k \to k^2$ relabeling into k synchronously running processes, assuming that set of $i|\bar{\imath}$ samples is actually the union of the appropriate subsets $\bigcup_{j=\overline{1,k},j\neq i} i|j$.

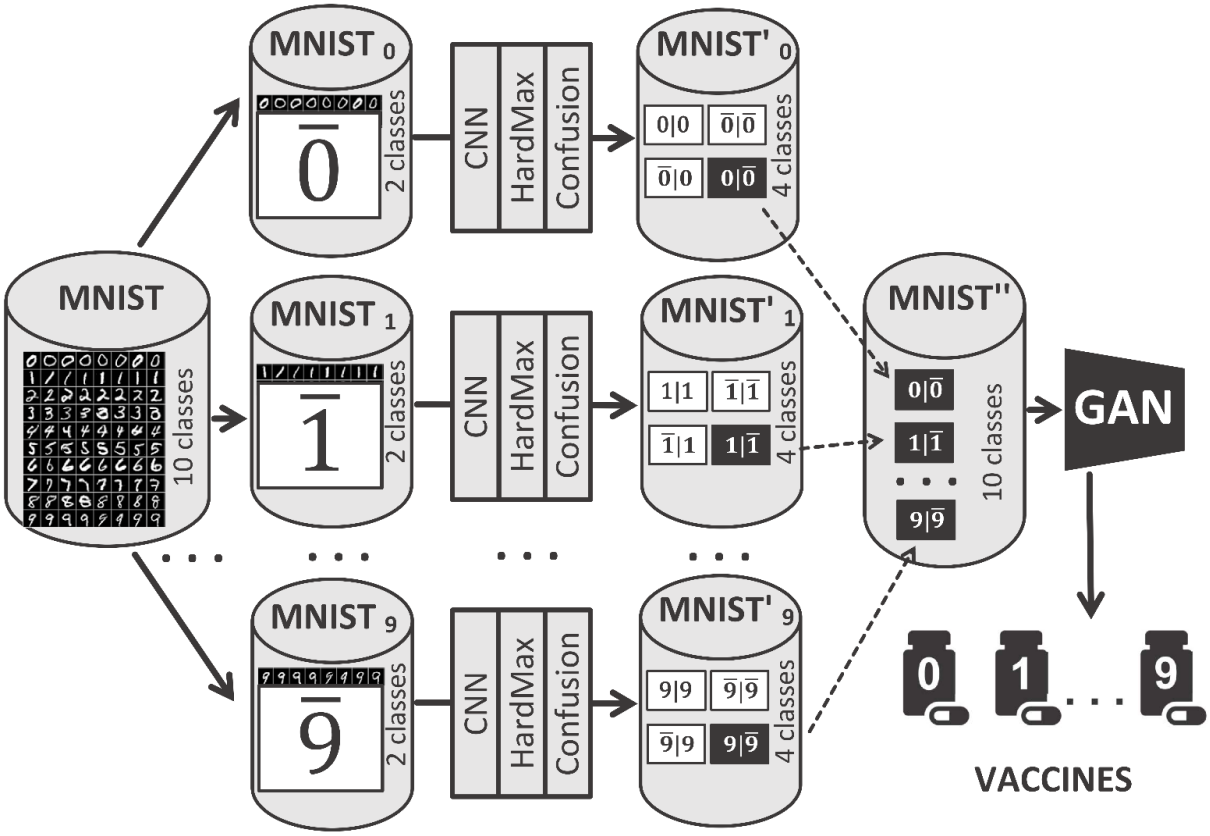

Fig. 4. Relabelling with the multichannel procedure.

3.2 GAN architecture for cognitive cloning and digital immunity development

Two types of learning models are combined in the GANs architecture: a generative deep neural network (Generator) plays against a discriminative deep neural network (Discriminator) in a specifically organized adversarial game. The main purpose is to train strong Generator which can create new data instances indistinguishable from those sampled from the true data distribution. Discriminator is secondary in the basic GANs architectures; it learns the boundaries of the reality and classifies samples as either real or fake (generated) to act as a good teacher or an assistant for the learning generative model.

However, decision-making is based on discriminative classification models: in GANs terms it is Discriminator that plays a role of a learning cognitive clone. It implies changes in the GANs architecture and the overall logic of the adversarial game. First, the main focus is now shifted to Discriminator while Generator becomes more of an assistant facilitating its further development. Second, traditional Discriminator cannot be used for cloning purposes since it does not learn to make any other decision than to distinguish real from fake data. Our task, however, is to apply adversarial learning to the classification problem.

The first cloning experiments were performed with Auxiliary Classifier GANs (AC-GANs) (Odena et al., 2017). This type of class-conditional image synthesis models contains modified Discriminator extended with an auxiliary decoder network that, besides deciding whether data is fake or real, outputs the class label for the training data. During the adversarial game the model improves not only its generative but also the classification ability. However, AC-GAN is a Lagrangian to a constrained primal objective function that down-samples points near the classifier's decision boundary and explicitly pushes the density of the generator distribution away from it (Shu et al., 2017). For training classification models in adversarial settings, we introduce a family of Turing-GAN (T-GAN) architectures (see Fig. 5) containing an extra player: Trainer, a pre-trained large classification model (or an ensemble of models). Thus, Discriminator of Turing-GANs (Turing Discriminator) contains 3 components:

- Trainer, classification model used for knowledge transfer;
- Trainee, an auxiliary classifier learning to benchmark decision-making of Trainer;
- basic discriminator.

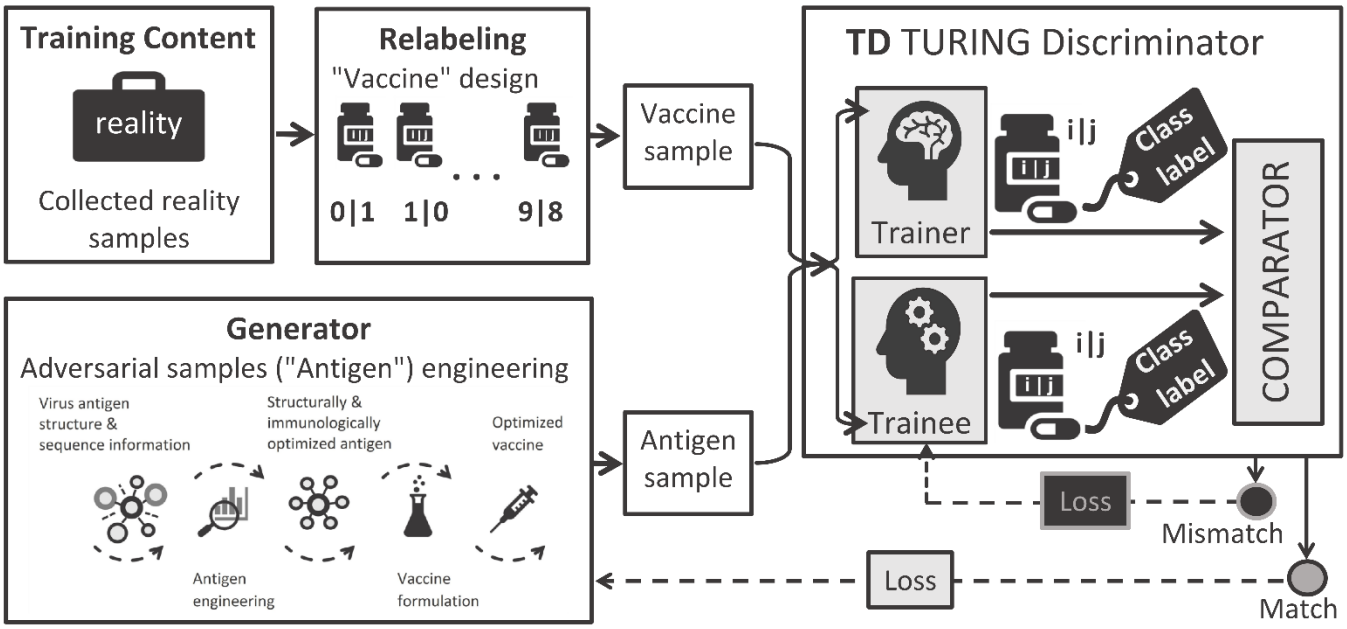

Fig. 5. New GAN architecture for adversarial training of classification models.

The rules of the new game are the following: both Trainer and Trainee are addressing the same data input independently.

The task of Trainee is to guess the label provided by Trainer, i.e., to minimize the difference in Trainer-Trainee behaviour in the same decision-making situations. Generator seeks to minimize the ability of Discriminator to discriminate real and fake images, maximizing the ability of Trainer to correctly predict the class label, and minimizing the ability of Trainee to predict the class label given by Trainer. Generator samples data in the local neighbourhoods of the classifier`s decision boundary to find the challenging data instances defining specific features of the Trainer`s decision boundary. The mismatch between Trainer and Trainee is measured by Comparator which transfers feedback to Trainee`s and Generator`s loss functions.

This scheme pushes the density of the generator distribution towards decision boundaries of the classifier. Thus, Generator learns how to generate challenging adversarial training content for

Trainee and, therefore, accelerates the training (cloning) process, while Trainee iteratively improves its imitation performance.

T-GAN-enabled adversarial training can be applied to different tasks that require learning sophisticated classification models, a.o., cognitive cloning and digital immunity development. Each application domain, however, implies its own specifics (see Fig. 6).

In security tasks Trainer is usually a security supervisor, Trainee is a discriminative classification model becoming more immune (robust) to various adversarial attacks. A pool of artificial adversarial data generated in the confusion zones, close to decision boundary is a "*digital vaccine*" for intelligent models.

Injection of new challenging training content (artificially generated or perturbed adversarial examples) into training datasets and re-training of intelligent models is called in this research "*smart vaccination*".

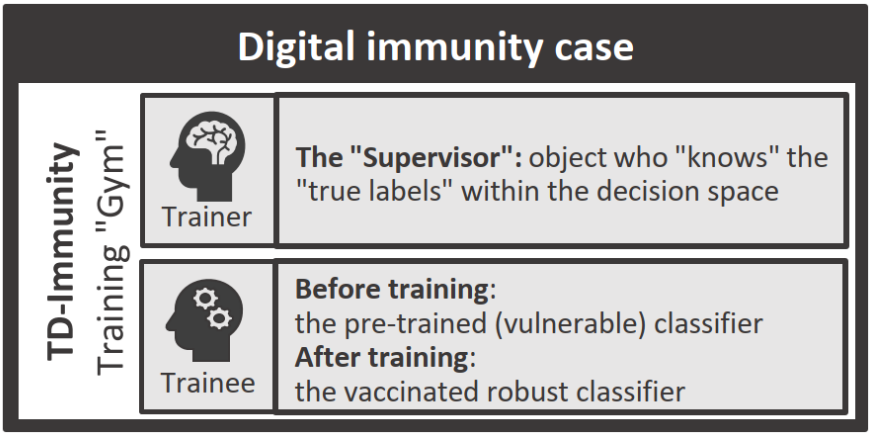

Fig. 6. Digital immunity development based on the new GAN.

4 Experiments on adversarial training of classification models

4.1. Boundary samples search and images relabelling in MNIST

The experiments were conducted in the deep learning environment created on top of 2 GPUs NVIDIA GeForce RTX 2080 Ti.

For the initial proof of our hypotheses we choose MNIST (LeCun, 1998), an available dataset with sufficient number of images requiring minimal efforts on pre-processing and formatting.

The basic CNN classifier is trained on the MNIST dataset containing 60,000 samples of handwritten digits stored as 28x28px images corresponding to 10 classes of integer values from 0 to 9. Our classifier has several convolutional layers with kernel 3x3. As an optimization algorithm we use the Adam optimizer with decay 1e-3 and the negative log likelihood loss (NLLLoss).

The task is to partition the obtained decision space into 100 more specific classes representing 100 neighbouring areas of digits written similar one to another (i.e., close to the boundary between two classes) and relabel all the data samples accordingly with respect to their position. For example, an image with initial label "1", which, according to the new partitioning, is located in the neighbourhood of the boundary between class "1" and class "7", gets label: "1|7". That indicates its affiliation to the boundary class "1|7" with respect to the reidentified decision boundaries. A sample deep within the class "1" gets label "1|1" according to the new partitioning.

To decide on the configuration of the Confusion Classifier, responsible for the classes specification, we experiment with 6 different activation functions SoftMax (SM), HardMax (HM), HardSquareMax (HSM), Z-Score-SoftMax (ZSM), Z-Score-HardMax (ZHM) and Z-Score-HardSquareMax (ZHSM) and two types of confusion functions (Confusion (Conf) and Naïve

Confusion (NConf)) for the output layer of the network. Different functions produce neighbouring areas of different size and shape. This is an essential factor that should be considered when choosing an appropriate combination for each specific dataset and classifier. The implemented experiments are listed in Table 1. Some of the most interesting results of these experiments are described below and summarized in Table 6.

Table 1. Experiments conducted with different parameters

	SM	HM	HSM	ZSM	ZHM	ZHSM
Conf	+	+	+	+	+	+
NConf	+	+	+	+	+	+

4.1.1. Relabelling MNIST with Confusion Layer

The output from the last fully connected layer of the classifier goes to one of the specified activation functions and then to the Confusion Layer. The Confusion layer calculates probabilities of the sample`s affiliation to 100 neighbouring areas.

Z-Score-SoftMax. We start with the SoftMax activation function and Z-Score Normalization. Some of the results are demonstrated in Table 2. Application of the SoftMax activation shows the implications of "the winner takes it all" principle when one neuron on the output layer of the basic MNIST-classifier inhibits others. The SoftMax activation hides part of the important information from the previous fully connected layers and impedes relabelling. No boundary areas are revealed in this case. The decision space is still divided into 10 classes of images.

Table 2. Relabelling results for Z-Score-SoftMax and Confusion Layer

						
MNIST label	5	0	4	1	9	2
MNIST Conf-label	5|5	0|0	4|4	1|1	9|9	2|2

Z-Score-Hardmax. We try to avoid the non-linearity effect typical to the SoftMax procedure. Z-Score-Hardmax leaves no "deep-within" areas and capture the smallest bias of the sample toward other classes (decisions). The combination of HardMax and Confusion Layer allows relabelling the dataset completely and fits smaller datasets. It is applicable in case of critical importance to keep all data homogeneous, i.e., corresponding to one type, such as, boundary samples. The results of the second experiment for the same few image samples are illustrated In Table 3.

Table 3. Relabelling results for Z-Score-HardMax and Confusion

						
MNIST label	5	0	4	1	9	2
MNIST Conf-label	5|3	0|2	4|1	1|4	9|4	2|1

Z-Score-HardSquareMax. The experiment is an attempt to compromise between SoftMax and HardMax functions. The boundaries of the neighbouring areas are defined differently, but we still have only boundary samples in the relabelled dataset (Table 4).

Table 4. Relabelling results for Z-Score-HardMax and Confusion

						
MNIST label	5	0	4	1	9	*2*
MNIST Conf-label	5|3	0|2	4|1	1|4	9|4	2|1

5.1.2. Relabelling MNIST with Naïve Confusion Layer

More uniform partitioning of the decision space into distinctive boundary and deep-inside areas is achieved due to application of the naïve version of the confusion function.

Z-Score-SoftMax. The combination of the SoftMax activation function, Z-Score Normalization and Naïve Confusion allows partitioning MNIST into 100 different classes with 86 of them populated with samples from the dataset (see Fig. 7). This combination gives 17,39% of boundary samples and 82,61% deep-within-classes samples. Some of the relabelling examples are shown in Table 5.

Table 5. Relabelling with Z-Score-SoftMax and Naïve Confusion

						
MNIST label	5	0	4	1	9	2
MNIST Conf-label	5|3	0|0	4|4	1|1	9|4	2|2

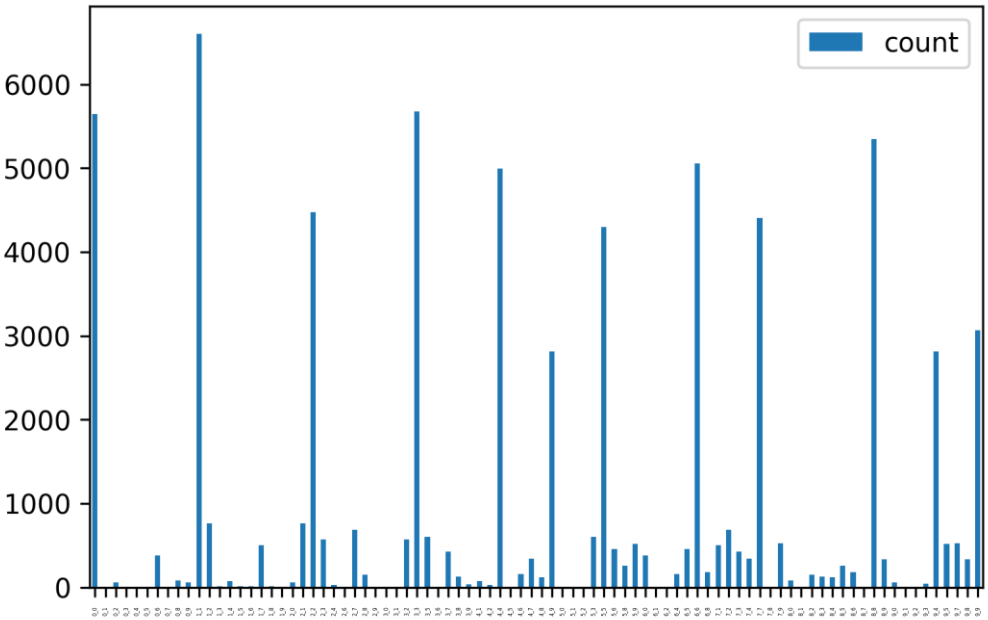

Fig. 7. Distribution of MNIST samples between 100 neighbouring classes (Z-Score-SoftMax and Naïve Confusion).

Table 6 summarizes the experiments and shows which types of decision space are obtained after the application of different activation and confusion functions. D denotes a dataset with deep-within-class samples, B – a dataset divided into neighbouring areas with boundary samples only, and BD corresponds to the hybrid decision spaces populated by samples that are considered both boundary and deep-within-class ones.

Table 6. Experimental results for MNIST

	SM	HM	HSM	ZSM	ZHM	ZHSM
Conf	D	B	B	D	B	B
NConf	BD	B	B	BD	B	B

4.2 Generation of challenging content for future adversarial re-trainings

The experiments in this part illustrate three different approaches to new challenging content creation for adversarial re-training of classification models: (i) generation of images in a framework of an attack-inspired GAN, (ii) corruption of real samples by generation of adversarial noise shifting them explicitly towards boundaries, and (iii) generation of adversarial samples using revealed boundary samples as a training dataset.

Confusion Classifier (distributing images among 100 specific classes of digits) is built with the Z-Score-SoftMax activation function and a Naïve Confusion output layer. The accuracy of the classification on the basic dataset is 99%.

4.2.1 Generating adversarial images with attack-inspired GAN

The implementation is based on AI-GAN implying that a generator, a discriminator, and an attacker are trained jointly (Bai et al., 2020). Once trained, the GAN is able to quickly process the entire dataset and interpolate between image classes. Unlike many other GANs it generates perceptually realistic adversarial examples by adding meaningful adversarial perturbations to initially real images. Depending on the parameters, we can create both adversarial and corrupted dataset in a short time. The success rate of the white-box attack performed by the GAN on MNIST dataset exceeds 98% given any targeted class (see Fig. 8).

Fig. 8. The original subset of real images and adversarial images with perturbations created with an attack-inspired GAN.

Comparison of the decision spaces on the original dataset and on the perturbed images proves that the structure of the samples` distribution changes, since the confusion level of the images increases essentially. Only 7,81% of the perturbed images stay deep-within-classes out of 82,61% in

the original dataset, while 74,8% of the images are shifted towards boundaries by perturbations, so that 92,19% of all the adversarial images get in the local neighbourhoods of the decision boundaries.

5.2.2 Shifting training data samples towards decision boundaries

The idea of this experiment is to simulate data with the higher level of ambiguity, in case of MNIST, imitate human illegible handwriting, and compare the distributions of the original and the new more ambiguous data. We corrupt original images with a series of black-box attacks performed by the attack-inspired GAN.

We experiment with different configurations of the generative model testing different levels of perturbations, changing activation functions and their parameters in several layers of the model. We choose the most successful combinations and parameters of the activation functions: the activation function for Generator – traditional ReLU, Swish, AReLU, Rational activation function; the activation function for the generated noise – LeakyRelu, two values of the slope parameter for the noise activation function – 0,3 and 0,7; two values for the coefficient of perturbations – 2 and 3. Results for several combinations are demonstrated below (see Fig. 9). Corrupted examples, unlike the adversarial ones, are created with visually perceptible changes. Despite their name, they are not needed for our model`s corruption, we only use them to approach the decision boundary and build Confusion Reality for the GAN training.

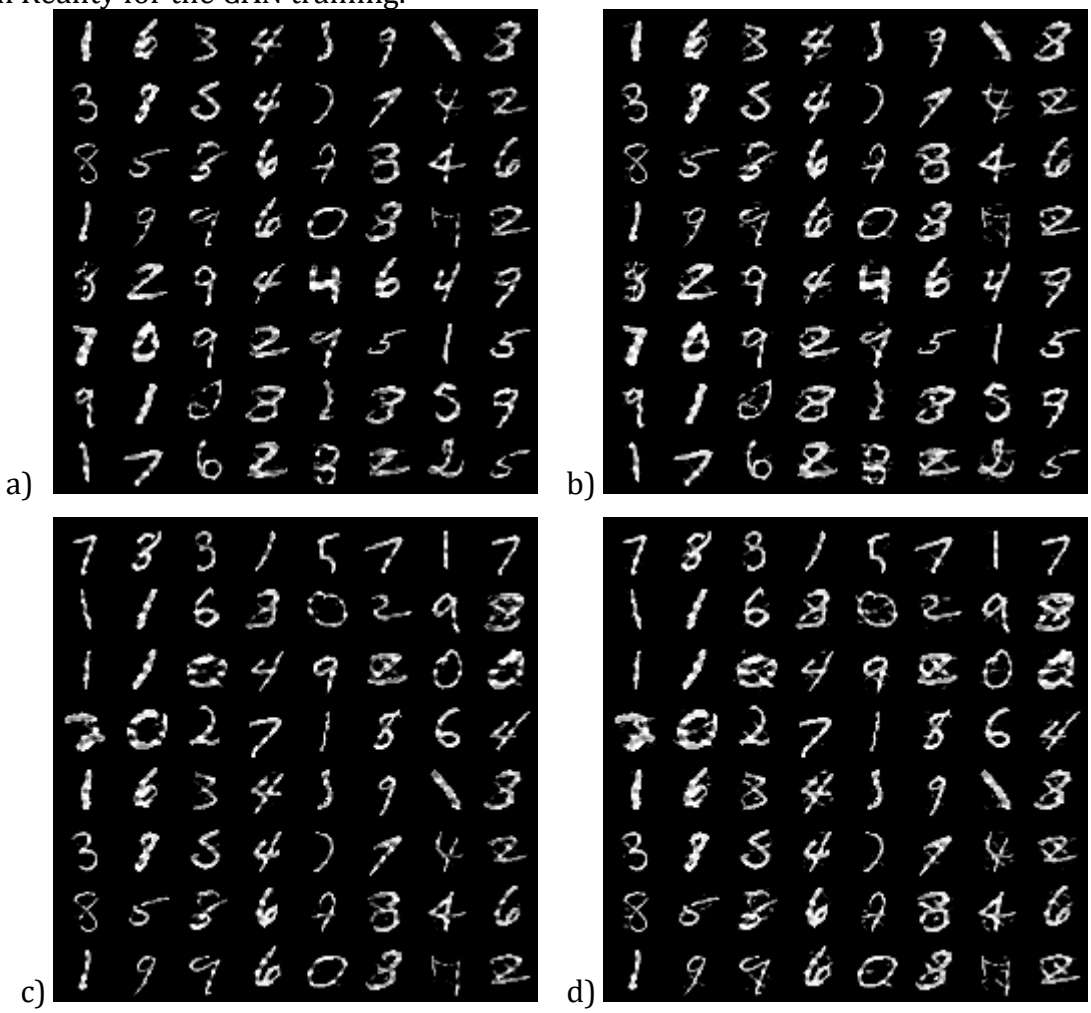

Fig. 9 Corrupted MNIST. Generator Activation function: a) Activation function: Swish, slope: 0,7 coefficient of perturbations: 2; b) Activation function: Swish, slope=0,3 coefficient of perturbations: 3; c) Activation function: ReLU, slope=0,7 coefficient of perturbations: 2; d) Activation function: ReLU, slope=0,3 coefficient of perturbations: 3.

As a prototype of an ambiguous dataset for further experiments, we use MNIST corrupted by the attack-inspired GAN with the Swish activation function (slope: 0,7; the coefficient of perturbations: 2).

Table 7 summarizes the experiments on ambiguous MNIST and shows how decision space looks after the application of different combinations of the activation and confusion functions in the Confusion Classifier. Although the results in the table resemble the original MNIST partitioning, there is an obvious shift of the examples towards the boundaries. Decision space of the Confusion Classifier on ambiguous MNIST consists of 82,48% images in the local neighbourhoods of the decision boundaries. They are distributed among 92 specific classes compared to 86 classes on original MNIST (see Fig. 10).

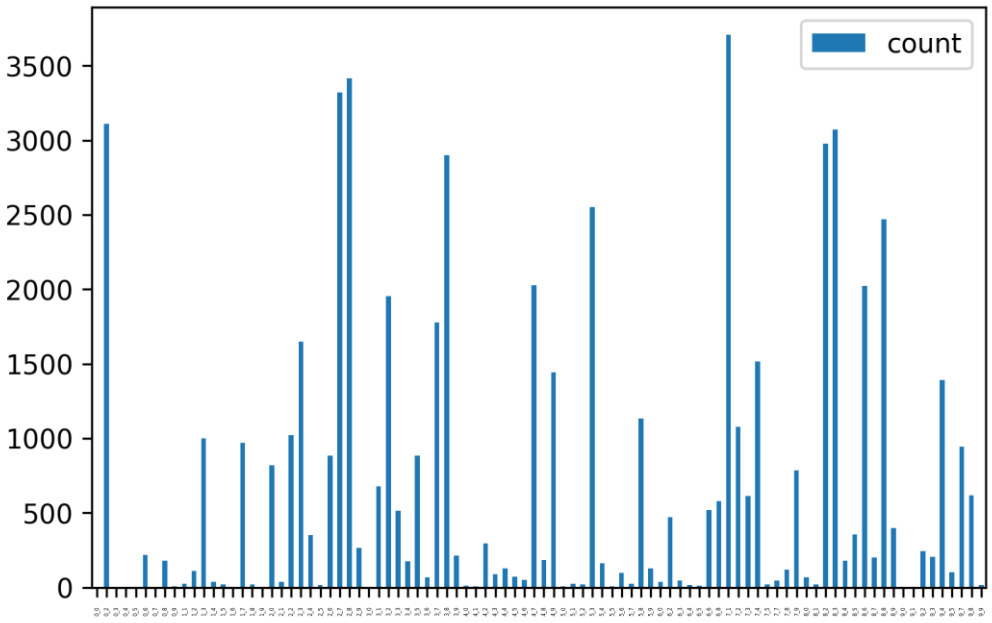

Fig. 10. Distribution of MNIST samples between 100 neighbouring classes by application Z-Score-SoftMax and Naïve Confusion.

Table 7. Experimental results on Ambiguous MNIST

	SM	HM	HSM	ZSM	ZHM	ZHSM
Conf	D	B	B	D	B	B
NConf	BD	B	B	BD	B	B

4.2.3 Adversarial sampling

Generation of new samples in the ambiguous areas close to the decision boundaries is performed with GAN trained on preselected boundary samples as a model of reality to be recreated. The results of generation of the new challenging content with the StyleGAN2 model (Karras et al., 2020) is shown on Figure 11.

Fig. 11. Artificially generated images in the local neighbourhood between class 1 and class 2.

4.3 Experiments on adversarial training of deep classification models

To test adversarial learning in a real industrial environment we use capacities of a logistic system based on an inter-roll cassette conveyor. The conveyor is used for the simulation of an airport luggage system and for the automated inspection at the security checkpoint. The sensors and actuators of the conveyer are interconnected via the respective programmable logic controllers under supervision of a Supervisory Control and Data Acquisition System. Decisions about the distribution of the cassette loads ("bags") are made by artificial intelligent components – digital security officers ("airport workers"). They decide whether the load's content is safe aiming to prevent any potential danger caused by the items in the load. The correspondent decision depends not only on the image recognition quality but also on the personal judgement and intuition of the security officers – their personal bias.

To train digital workers to distribute and dispatch the cassettes, an airport inspection procedure held by three different human experts is simulated. The Conveyer-v2 dataset created for training contains 2198 images of cassettes taken by the cameras installed in the critical distribution points of the conveyer. The images present various configurations of the cassettes and items in them (see Fig. 12).

The experts label each image either as "dangerous" if it is suspected to contain some potentially unsecure object(s) or as "not dangerous" otherwise.

For the experiments we imitate non-hazardous items with plastic balls and pills of different colours and hazardous items with red hearts which are considered a "bomb".

5.3.1. Airport luggage inspection

Three classifiers based on a deep convolutional neural architecture mimic decision behaviour of human inspectors. The transfer learning technique is used for the optimization purposes.

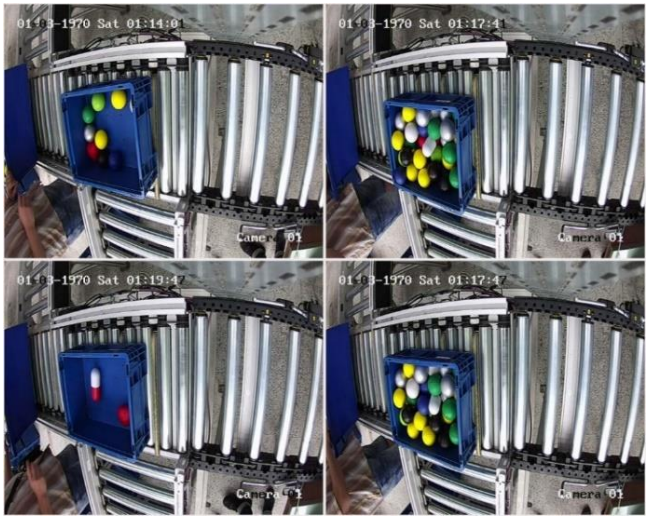

Fig. 12. Image samples representing various configurations of cassettes from Conveyer-V2 dataset

The image feature extraction module uses the Inception V3 architecture trained on ImageNet (image-net.org). The accuracy of the trained artificial security officers is evaluated with respect to four parameters: (1) the actual correctness of the classification obtained on the test and validation sets (see Tab. 8); (2) correlation between the artificial inspectors' and the human experts' decisions (see Tab. 9); (3) the actual correctness of the decisions from human experts (see Tab. 10); (4) the correctness of the human experts in case of artificial decision advice (see Tab. 11).

Although human decision-makers show better performance in threat recognition, the results are promising due to the high accuracy of the artificial predictions and the high human-clone correlation

of the decisions. Moreover, artificial experts appear to be capable of valuable advice since the human donors improve their accuracy after getting an advice from the artificial inspectors.

Table 8. Classification accuracy

Artificial decision-maker	Test set	Validation set
Classifier_1	92.05%	87.5%
Classifier_2	97.95%	96.5%
Classifier_3	95.23%	94%

Table 9. The correlation of artificial and human security inspectors

Decision-makers	Test set	Validation set
Classifier_1 vs DM_1	92.05%	92.1%
Classifier_2 vs DM_2	99.53%	99.48%
Classifier_3 vs DM_3	95.66%	94.94%

Table 10. The actual correctness of the human experts

Human decision-makers	Test set	Validation set
DM_1	95.45%	95%
DM_2	98.41%	97%
DM_3	99.55%	99%

Table 11. The correctness of the human experts after an artificial advice

Human decision-makers	Test set	Validation set
DM_1	97.27%	95%
DM_2	98.86%	98%
DM_3	99.77%	97.5%

4.3.2 Airport luggage inspection under adversarial attacks

An artificial security officer is trained to be prepared to new tasks and complex contexts. A disrupted reality and unexpected confusing conditions are simulated by poisoned images coming from the camera so that they are expected to be misclassified by the artificial predictors. Two attack scenarios are considered:

- Attack Scenario 1: there is “a bomb in the bag” but the image is “poisoned” to be potentially misclassified as being “not dangerous”. The intent of this attack is to allow dangerous load coming undetected “on board”.
- Attack Scenario 2: there is “no bomb in the bag”, but the image is “poisoned” to be potentially misclassified as being “dangerous” causing a false alarm. The intent of this scenario is to cause disruption in the normal operation of the system, causing delays and panic.

Applying different types of adversarial noise during the so-called white-box attacks we can compromise the work of the deep-learning convolutional classifiers completely and decrease their accuracy to less than 1%. Experiments show that both artificial and human workers tend to misclassify poisoned images (see Table 12).

Table 12. The accuracy of the tampered images recognition

Human decision-makers	Digital decision-makers
75%	70.05%
90%	71%
85%	65.5%

To foster resilience of decision-making in the logistic system under cyber-attacks, the clone is trained to develop a new capability – an artificial cognitive immunity against potential adversarial attacks.

Training a powerful discriminator to recognize tampered images based on StyleGAN2 architecture.

Using data-driven unconditional generative image modelling based on StyleGAN2 architecture configured as shown in Table 13, we pursue two objectives: to train a powerful discriminator capable of detecting a tampered image; and to generate a pool of new high quality images, which are deliberately designed as adversarial samples aka “digital vaccine” (see Fig. 13) for smart vaccination during comprehensive retraining of the digital clones.

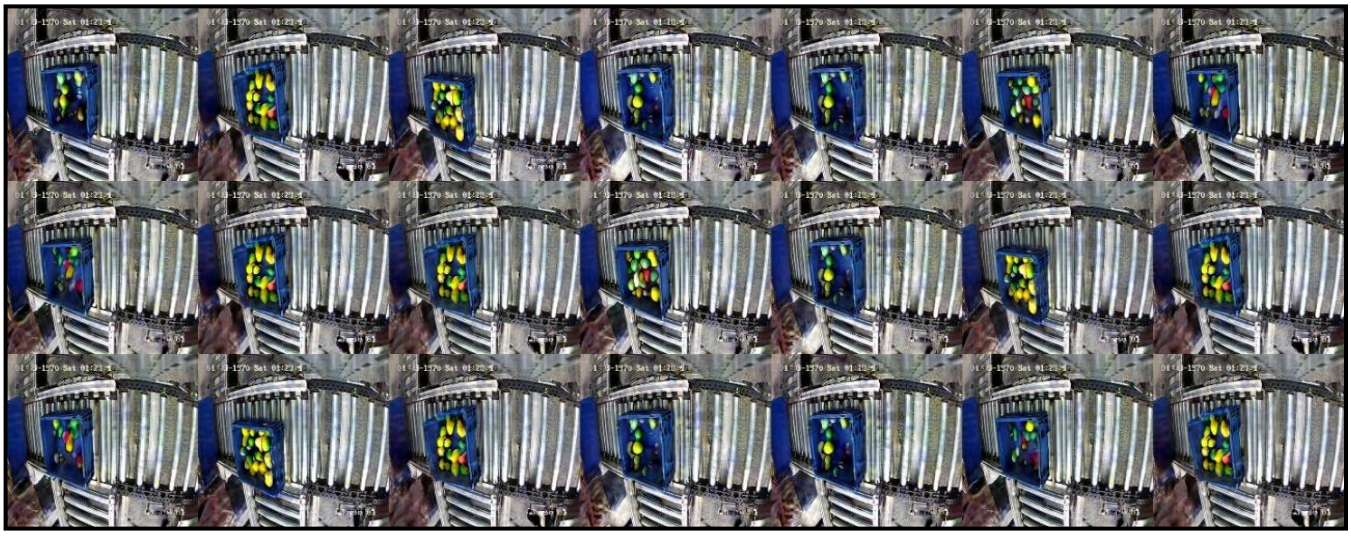

Fig. 13. A pool of artificially generated images used for smart vaccination

Table 13. The parameters of the StyleGAN2

Parameter	Value
Configuration	config-a
Resolution	1024x1024
Total number of images	25000
GPU	2
Duration	45d23h

GPU Memory	7.5 Ghz

From 3000 generated images a batch of 300 samples, the most challenging ones for both human and artificial security officers, is selected as the samples of the disrupted reality used to retrain digital security officers. The experiments show an increase of the discrimination accuracy after retraining/vaccination (see Table 13).

Table 13. The accuracy of the tampered image recognition

Human decision-makers	Digital decision-makers
72%	80.05%
92%	75%
80%	70%

5 Conclusions

We support our assumptions and the approach with the set of experiments on top of public MNIST dataset and also with the data from real industrial processes. Experiments have shown that our analytics is able to successfully discover the target confusion zones for additional self-supervised learning. The suggested architectures appeared to be available to generate additional training content ("personalized vaccines" for data injection) within the target zones (either by from-the-scratch new adversarial images' generation or by adding adversarial noise to the available images and pushing them closer to the decision boundaries).

We show several experiments with the adversarial learning environment designed for the real industrial processes. Current limitation of our implementation is that we cannot yet guarantee that the adversarial samples are covering the close-to-boundary space evenly along the entire decision surface. This is an objective of our future research. Also, we are planning to address the challenge of digital cloning of evolving smart objects with non-deterministic behavior.

References

Bai, T., Zhao, J., Zhu, J., Han, S., Chen, J. and Li, B., 2020. Ai-gan: Attack-inspired generation of adversarial examples. arXiv preprint arXiv:2002.02196.

Dasgupta, S. and Hsu, D., 2008, July. Hierarchical sampling for active learning. In Proceedings of the 25th international conference on Machine learning (pp. 208-215).

Deng, L., 2012. The mnist database of handwritten digit images for machine learning research [best of the web]. IEEE Signal Processing Magazine, 29(6), pp.141-142..

Ertekin, S., Huang, J., Bottou, L. and Giles, L., 2007, November. Learning on the border: active learning in imbalanced data classification. In Proceedings of the sixteenth ACM conference on Conference on information and knowledge management (pp. 127-136).

Goodfellow, I.J., Shlens, J. and Szegedy, C., 2014. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572.

Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., 2014. Generative adversarial networks. arXiv preprint arXiv:1406.2661.

Heo, B., Lee, M., Yun, S. and Choi, J.Y., 2019, July. Knowledge distillation with adversarial samples supporting decision boundary. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 33, No. 01, pp. 3771-3778).

Hjelm, R.D., Jacob, A.P., Che, T., Trischler, A., Cho, K. and Bengio, Y., 2017. Boundary-seeking generative adversarial networks. arXiv preprint arXiv:1702.08431.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J. and Aila, T., 2020. Analyzing and improving the image quality of stylegan. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 8110-8119).

Karimi, H., Derr, T. and Tang, J., 2019. Characterizing the decision boundary of deep neural networks. arXiv preprint arXiv:1912.11460.

LeCun, Y., 1998. The MNIST database of handwritten digits. http://yann. lecun. com/exdb/mnist/.

Odena, A., Olah, C. and Shlens, J., 2017, July. Conditional image synthesis with auxiliary classifier gans. In International conference on machine learning (pp. 2642-2651). PMLR.

Terziyan, V., Gryshko, S. and Golovianko, M., 2018. Patented intelligence: Cloning human decision models for Industry 4.0. Journal of manufacturing systems, 48, pp.204-217.

Terziyan, V. and Nikulin, A., 2019. Ignorance-Aware Approaches and Algorithms for Prototype Selection in Machine Learning. arXiv preprint arXiv:1905.06054.

Terziyan, Vagan, Mariia Golovianko, and Svitlana Gryshko. "Industry 4.0 Intelligence under Attack: From Cognitive Hack to Data Poisoning." Cyber Defence in Industry 4.0 Systems and Related Logistics and IT Infrastructures 51 (2018): 110.

Vlassopoulos, G., van Erven, T., Brighton, H. and Menkovski, V., 2020. Explaining Predictions by Approximating the Local Decision Boundary. arXiv preprint arXiv:2006.07985.

Weinstein, B., Fine, S. and Hel-Or, Y., 2019. Selective sampling for accelerating training of deep neural networks. arXiv preprint arXiv:1911.06996.

Yousefzadeh, R. and O'Leary, D.P., 2019. Investigating decision boundaries of trained neural networks. arXiv preprint arXiv:1908.02802.

Zhou, D., Lu, L., Zhao, J., Wang, D., Lu, W. and Yang, J., 2020. A new learning algorithm based on strengthening boundary samples for convolutional neural networks. In MATEC Web of Conferences (Vol. 327, p. 02004). EDP Sciences.