Egyptian Informatics Journal 21 (2020) 241–257

Contents lists available at ScienceDirect

Egyptian Informatics Journal

journal homepage: www.sciencedirect.com

Full length article

An efficient machine-learning model based on data augmentation for pain intensity recognition

Ahmad Al-Qerem

[a] Department of Computer Science, Faculty of information technology, Zarqa University, Zarqa, Jordan

ARTICLE INFO

Article history:
Received 28 November 2019
Revised 20 January 2020
Accepted 19 February 2020
Available online 24 March 2020

Keywords:
Machine-learning
Data augmentation
Pain intensity recognition
Features selection
GANs

ABSTRACT

Pain is defined as "a distressing experience associated with actual or potential tissue damage with sensory, emotional, cognitive and social components", knowing the exact level of pain experienced to have a critical impact for caregivers to make diagnosis and make he suitable treatment plan, but the available methods depend entirely on the patient self-report, which increase the difficulties of knowing the accurate level of pain experienced by the patient. Therefore, automating this process became an important issue, but due to the hardness of acquiring medical data, it became difficult to build a predictive model with good performance. Generative Adversarial Networks is a framework that generates artificial data with a distribution similar to the real data, by training two networks; the generator which tries to generate new samples similar to the real ones, and the discriminator which applies a traditional supervised classification to distinguish the augmented samples, the optimal case is when the discriminator cannot distinguish the augmented samples from the real samples. In this research, we generated data using Least Square Generative Adversarial Networks and the study the effect of applying feature selection on the data before the augmentation. Moreover, the approach was tested on a dataset that contains multi biopotential signals for different levels of pain.

1. Introduction

Pain is an essential signal in the nervous system that indicates something is wrong inside the human body and requires a physician consideration [1], it is an irritating and complicated feeling that disturb the patient making him uncomfortable. Pain is affected by many different factors, like age and gender [2]; it comes in many different ways and levels such as burn, ashes or pricks in a particular area in the human body.

Pain has two main types; first is the acute pain, which might be caused by several reasons like exposing to an injury or having a disease. If the cause behind this type of pain is not appropriately treated, the problem will aggravate causing more significant untreatable problems. The second type is the chronic pain, caused by severe, untreated injury, infection or ongoing disease like cancer, this kind of pain stick with the patient for months or years and is not always curable, but physicians try to manage it so the patient can feel more comfortable. Untreated pain has a significant impact on the patient and his family, making him restless, affect his character, causing a sleeping disorder and may lead to depression.

In order to diagnose the cause of pain, making a proper treatment plan and knowing the best medicine dose to give the patient, the caregiver must know the exact level of pain experienced by the patient. Caregivers usually rely mainly on the patient self-report to specify the level of pain he is experiencing using specific scales designed for this purpose.

There are different scales used to recognise the level of pain for patients, the most used scale is the Visual Analogue Scale (VAS), which is commonly used because of its simplicity, the main idea behind VAS is asking the patient to choose the pain level from 0 to 10, where 0 represented no pain and 10 represent the worst pain imaginable [3], Fig. 1 shows the levels of pain used. VAS has its weaknesses especially with patients that have a mental illness, infants, and traumas who cannot use it properly, moreover, VAS

Peer review under responsibility of Faculty of Computers and Information, Cairo University.

Production and hosting by Elsevier

E-mail address: ahmad_qerm@zu.edu.jo

https://doi.org/10.1016/j.eij.2020.02.006
1110-8665/

depends on the past experience patient had with pain, and fails to detect addicts whose faking pain to get drugs.

In order to determine the pain intensity level accurately, researchers start to automate the pain intensity recognition based on several factors like facial expressions and biopotential signals. Facial expressions are recorded for subjects while experiencing pain through a controlled experiment. Biopotential signals are simply electronic signals produced by the electrochemical activities of a cell type during physiological processes that occur in the body, they are measured by attaching specific type of sensors to the skin called an electrode. Pain recognition automation can be done using either data or both of them. In this work, we are going to use biopotential signals dataset containing four levels of pain; the signals are obtained from Electromyography, Skin Conductance Level, and Electrocardiogram, the dataset is described in details in Section 3.

Pain Intensity levels can be classified using prediction algorithms which feeds on data, as it is known the more data given to the algorithm, the more effective it can be [4]. Medical problems suffer from the lack of data because of its sensitivity, therefore, the performance of the prediction algorithm is affected.

Data augmentation algorithms trains on the real data to generate artificial (augmented) data, by using this kind of data, the performance of the prediction algorithms can be significantly enhanced. Recently, the Generative Adversarial Networks (GANs) have gained attention because of the augmented data quality, and the wide set of applications using it.

GANs train two neural networks; Generator and Discriminator plays a game together, the generator trains on the real data to generate artificial data follows the same distribution as the real data distribution, the generated data is sent along with real data to the discriminator, which applies supervised classification on real and augmented data to distinguish between them.

Automatic pain intensity recognition requires building a machine learning model to predict the level of pain experienced by patients, models feed on previously labelled data to learn how to predict the level of pain for new patients, the more data exist, the better the model learns. Acquiring medical data usually is a problem due to the sensitivity of the required data since it requires human subjects to be exposed to different levels of pain under a physician's supervision. To overcome this problem, data augmentation technique can be used to generate artificial data similar to the real data and fed it to the predictive model to enhance the performance.

In this work, we aim to enhance the pain intensity recognition problem performance, by using a variant of GANs named the Least Square Generative Adversarial Networks (LSGANs), to generate augmented biopotential signal data, then make the classification of pain levels using the Support Vector Machine (SVM) algorithm. Moreover, we are going to experiment the effect of generating only selected features by boruta algorithm on the classification accuracy, at the end we will test the quality of the generated data after making adjustments on the least square loss function. The main objectives of this work can be summarized as the following:

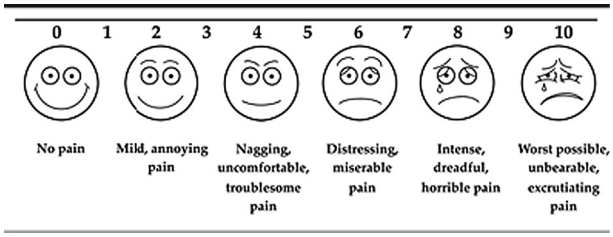

Fig. 1. Visual Analogue Scale.

1. Generate artificial data using LSGANs.
2. Improve the performance of the pain intensity level classification using SVM classifier.
3. Experiment the effect of generating only selected features using boruta algorithm on the classification.
4. Experiment the effect of making adjustments on the least square loss function on the classification accuracy.

This research is divided into five sections. Section 1 displays the introduction to the research problem and methodology. Section 2 presents some of the previous work done in the field. Section 3, explains the Generative Adversarial Networks (GANs) framework, Section 4, presents the methodology of this work. Section 5 provides an overview of the dataset used in this work and Section 6, presents the experiments and results.

2. Background and related work

Pain is defined by [5] as "a distressing experience associated with actual or potential tissue damage with sensory, emotional, cognitive and social components", it is an unpleasant and complex feeling that indicates something is damaged within the body. Knowing the exact level of pain experienced to have a critical impact for the caregivers to make a diagnosis and make the suitable treatment plan, unreliable description of pain can lead to mistakes like dosage error. Pain levels arrange between pain threshold which is when the patient starts feeling pain and pain tolerance threshold in which the patient cannot endure the pain anymore. Usually, pain intensity measurements depend on the patient self-report, the most common measure is the Visual Analogue Scale (VAS), in which the patient is asked to choose the level of pain visually from 11 levels (0–10).

The problem with this kind of measures is its dependence on patient awareness, communication, and experience of pain. Thus, it will not work on cases like traumas and infants. Therefore, more reliable ways of measuring the pain level have to be developed, such as automatic pain intensity recognition.

Automatic pain intensity recognition depends on the patient expressions and reflexes, the recognition can be done using recorded facial expressions, physiological signals, or both. The most famous datasets are UNBC-MacMaster Shoulder Pain Expression Archive Database [6] and BioVid Heat Pain Database [7].

In 2017, Lopez-Martinez and Picard R. [8] introduce an approach containing two-stage learning for the 10 levels VAS automatic estimation. The first stage is using Recurrent Neural Network (RNN) for Prkachin and Solomon Pain Intensity (PSPI) score estimation which is another pain intensity measurement from face images, which are fed to the next stage. The second stage takes the PSPI score for each person for VAS estimation using Hidden Conditional Random Fields (HCRFs); the model is personalized using a unique score for each person from his facial expressiveness. The dataset used contains 25 subjects suffering from shoulder pain; their faces were recorded while doing movements in both the affected and unaffected arms. For evaluation purposes, the dataset was split into train and test sets, after training the model on the training model, the Intra Class Correlation (ICC) and the Mean Absolute Error (MAE) were used as evaluation measures. Moreover, the results of the two algorithms were compared with the Support Vector Regression (SVR) as a baseline method.

Zhou J. [9] presented framework used Recurrent Convolutional Neural Network (RCNN) for automatic frame-level pain intensity estimation. First, the faces Images were wrapped to the same frontal pose from different poses using Active Appearance Model (AAM). Second, since extracting features from each frame in the video separately have its limitation in describing dynamic informa-

tion from previous frames, features were extracted from adjacent frames using a sliding-window strategy to allow using historical frames while keeping a fixed-length input. Finally, the recurrent convolutional neural network architecture was designed for continuous-valued pain intensity, by modifying the loss and activation function in the last layer of the network. At each iteration in the training process, a subset of the training set containing all PSPI levels with the same percentage was taken. As for the evaluation phase, following the leave-one-subject-out strategy, the average Mean Squared Error (MSE) and Product-moment Correlation Coefficient (PCC) were calculated, the experiments showed promising results compared to other approaches on the same dataset.

Thiam P., and Schwenker F. [10] presented a personalized pain recognition system depends on a hierarchical fusion architecture to take advantage of all the extracted features, where dimensionality reduction is applied on each set of features first, then each set of the remaining features are sent to the classification process, which consists of three layers, first, each subset is fed to the random forest classifier, second, the results are sent to a pseudo-inverse mapping and a multi-layer perceptron (MLP) mapping, finally, using both of the scores of the second layer, a pseudo-inverse mapping produce the final label. The model was personalized using the Hausdorff distance as a similarity metric, to select samples from participants from the training set that are similar to an unseen participant. The evaluation of this model follows the leave-one-participant-out strategy, the results outperform the baseline methods using 30 participants.

R. Lopez-Martinez [11] in 2017, implemented an approach based on multi-task learning using neural networks, that takes into account the individual differences in pain responses while still can learn from other subjects' data. The dataset originally contains the raw signals, therefore, features were extracted from the signals before being fed to the network. As for the classification, it was split into binary classification between each pain level with the no pain level experimented on each set of extracted features and was evaluated using the 10-fold cross-validation, the obtained accuracy results outperformed other baseline algorithms.

Lopez-Martinez, 2018, [12], presented an approach to handle the pain intensity recognition as a regression problem, where two types of Recurrent Neural Network (RNN) architectures are used, the first architecture is a traditional fully-connected RNN which is able to capture temporal dependencies since the output is fed back to the input. The second architecture is a Long Short-Term Memory Neural Network (LSTM-NN) which learns the long-term dependencies. The experiments showed that the results are obtained using the skin conductance features outperforms other baseline methods, using three evaluations metrics; Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and coefficient of determination (R2).

Chu Y. [13], in this research, the authors focused on the feature processing, since the dataset used is novel, after extracting features from the raw signals, then the extracted features are reduced using Genetic Algorithm (GA), to remove redundant and irrelevant features, then applied Principal Component Analysis (PCA), to transform the features into linearly uncorrelated space. Three types of classification including linear discriminant analysis (LDA), k-nearest neighbour (KNN) algorithm, and support vector machine (SVM) were applied and evaluated for single-signal datasets, multi-signal datasets, multi-subject datasets and multi-day datasets.

Werner P [14], in this paper, the proposed approach, features were extracted from both video and biomedical signals and were combined together to classify the pain intensity levels. The proposed classification process is an early fusion architecture using Random Forest classifier; many combinations of features were tested using the same classification architecture. Moreover, grid search was used to find the most optimal parameters. The experiments results evaluation followed the 10-fold stratified cross-validation for each subject. The best overall performance was obtained by using all the video and biomedical features.

Thiam P. [15], proposed an approach to find which modality from both video and signal features give the best result in the pain intensity recognition problem in person independent setting. Each modality was used to train a regression model based on random forest, and in the end, all of the modalities were used together in an early fusion model. The used evaluation metrics are Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE), experiments showed that Skin Conductance Level (SCL) give the best results on its own, while the Electrocardiogram (ECG) has the worst performance.

A. S. F. Thiam P. [16], in this work, a trimodal of audio, video, and signals were used for the pain intensity recognition problem, by doing experiments on several fusion architectures, to take the full advantage of the diversity of the extracted features. Both early and late fusion were applied, in the early fusion, all the extracted features is fed to a single Random Forest classifier. As for the late fusion, two architectures were presented, First, Late Fusion (A) in which the extracted features belonging to the same channel (Audio, Video, Signals) are concatenated, then fed to the random forest classifier. Second, Late Fusion (B), where each set of extracted features belonging to one modality is fed to a single Random Forest classifier, both architectures were mapped using the mean and Linear Discriminant Analysis (LDA) to decide the output. The evaluation of the fusion architectures showed that user-specific classification outperforms independent settings classification.

Kachele M. [17] proposed personalizing pain intensity recognition systems using different techniques with different information sources, the idea behind penalization is to estimate the pain level by identifying the most similar subjects in training set to the new test subject. The used techniques to find the similarities between subjects belongs to three main groups; First, Meta-information, which is general information about subjects like age and gender. Second, distance-based measures, the proposed techniques belonging to this group are K-Nearest Neighbor and Hausdorff distance. Finally, machine learning-based algorithm, this group is divided into measures based on supervised learning, measures based on unsupervised machine learning and proxy classification.

Table 1 shows some of the work done in this area; the methodology for each work is described after the table.

Table 1 Pain Intensity Recognition Related Work.

Data augmentation is an effective technique used to increase the amount and diversity of samples to enhance machine learning algorithms' performance; this kind of techniques is mostly used in computer vision applications. The simplest algorithms are making transformations on the existing images, such as flipping, cropping, and rotation [19]. The problem with such methods is that they do not generate a brand new sample, it just modifies the sample, that is why researchers start developing new algorithms for a more effective way for augmenting data.

Recently Cubuk E. [20], introduced an AutoAugmentation approach. AutoAugmentation is a procedure that searches for the most suitable data augmentation policy; each policy consists of many sub-policies, where each sub-policy contains two operations represents a function such as rotation. In this work, reinforcement learning is used to find the best combination of choices and orders of the functions that yield the neural network to give the best accuracy. Experiments showed that this procedure can be customised for a specific dataset, or can be transferred to other datasets.

Zhong [21], in this paper, a data augmentation method for training the Convolutional Neural Network (CNN) called Random Erasing was implemented, the main problem this method aims to solve

Table 1
Pain Intensity Recognition Related Work.

Num	Paper Name	Year	Type	Levels	Dataset	Results (The best result obtained)
1	Personalized Automatic Estimation of Self-Reported Pain Intensity from Facial Expressions [8].	2017	Video	VAS (11)	UNBC – McMaster Shoulder Pain Expression Archive Database	MAE: (mean) 2.47, (std) 0.18
2	Recurrent Convolutional Neural Network Regression for Continuous Pain Intensity Estimation in Video [9].	2016	Video	PSPI (16)	UNBC – McMaster Shoulder Pain Expression Archive Database	MSE: (mean) 1.54
3	Hierarchical Combination of Video Features for Personalized Pain Level Recognition [10].	2017	Video	4	40 participant dataset	Accuracy: 67.8%
4	Multi-task neural networks for personalized pain recognition from physiological signals [11].		Signals	5	BioVid Heat Pain Database	Accuracy: 79.98% [t]
5	Continuous Pain Intensity Estimation from Autonomic Signals with Recurrent Neural Network [12].	2018	Signals	5	BioVid Heat Pain Database	R2: (std) 0.22
6	Physiological signal-based method for measurement of pain intensity [13].	2017	Signals	4	6 subjects	Accuracy: 75%
7	Automatic Pain Recognition from Video and Biomedical Signals [14].	2014	Video, Signals	5	BioVid Heat Pain Database	Accuracy: (P0 vs P4) 80.6%
8	Multimodal Data Fusion for Person-Independent Continuous Estimation of Pain Intensity [15].	2015	Video, Signals	5	BioVid Heat Pain database	MAE: 0.84
9	Multi-modal Data Fusion for Pain Intensity Assessment and Classification [16].	2017	Audio, Video, Physiology	4	SenseEmotion Database [18]	Accuracy: (T0 vs T3) 85% [t]
10	Methods for Person-Centered Continuous Pain Intensity Assessment From Bio-Physiological Channels [17].	2016	Signals, Video	5	BioVid Heat Pain database	Accuracy: 40.48%, MAE: 0.892

is the occlusion problem, and improve the generalisation of CNNs. In training, the algorithm randomly chose a rectangle region of random images with an arbitrary size, and replace the original pixels with random values; therefore, more occlusion levels are generated.

Xie Q. [22], in this work, a data augmentation method is applied to unlabeled data in a semi-supervised learning setting; this method called Unsupervised Data Augmentation (UDA). UDA make the model more consistent while training the real unlabeled data, and the augmented unlabeled data, instead of using random noise, UDA use more realistic noise generated by previous data augmentation methods, and minimise the KL divergence between the predictions on the real data, and the predictions on the augmented data.

Tran T. [23], in this paper, the authors proposed using a novel Bayesian formulation for data augmentation, by treating the new data as missing data points that are sampled from the distribution of the given annotated data points. Moreover, both the process of the generator distribution and the classification model are trained jointly, and this method showed improvement in the result.

Lim S., 2019. [24] In this paper, an algorithm called Fast AutoAugment is proposed Bayesian Data Augmentation motivates that, Fast AutoAugment treats augmented data as missing data points, and are recovered by the exploitation-and-exploration of a family of inference-time augmentations, this search is optimized by Bayesian algorithm. Experiments using several datasets showed that the search time is speeding up than AutoAugment, and the error rate was improved.

Ho D., 2019. [25] introduced Population-Based Augmentation (PBA) replaced using fixed augmentation policy through the training epochs, by generating the best schedule of augmentation policies for each training epoch, the main advantage of using PBA algorithm is needless of intensive computing power, experiments showed that the time needed for training PBA is significantly less than other algorithms.

Table 2 shows some of the work done in this area, the methodology for each work is described after the table.

3. Generative adversarial networks

Generative models with adversarial networks have been considered recently as one of the most promising and interesting techniques in the field of data augmentation, as it led to a significant improvement for the image generation. GANs outperforms other techniques in terms of generated samples quality, and the usefulness in many different applications, like transforming text to images [27].

3.1. Generative models

Class of unsupervised learning problems related to generating fake samples from existed data, this class goes beyond extracting patterns from the training samples, to learn their underlying distribution, then design a model to generate fake samples following the same distribution as the real samples. The distribution of the real data is notated as P_{data}, while the distribution learned by the model is notated as P_{model}, the objective of the generative model is to make P_{model} as similar as possible to P_{data}, therefore, the optimum model will learn P_{model} that is equal to P_{data}. Generative models can be designed either by using density estimation; which aims to reconstruct the probability density function of the given samples or using sample generation; which generate new samples beside estimating the density of the training samples. Generative models can be separated into two main categories [28] based on the function used to evaluate the quality of the generated samples:

Table 2
Data Augmentation Related Work.

Paper Name	Year	Datasets	Results (best result obtained)
AutoAugment: Learning Augmentation Strategies from Data [20].	2019	SVHN, and ImageNet	Accuarcy: 83.5%, Error rate: 1.0%
Random Erasing Data Augmentation [21].	2017	CIFAR-10	Error rate: 3.08%
Unsupervised Data Augmentation for Consistency Training [22].	2019	CIFAR-10, and ImageNet	Accuracy: 79 %, Error rate: 2.7%
A Bayesian Data Augmentation Approach for Learning Deep Models [23].	2017	CIFAR-10	Accuracy: 93%
Fast AutoAugment [24].	2019	CIFAR-10	Error rate: 2.0%
Population Based Augmentation: Efficient Learning of Augmentation Policy Schedules [25].	2019	SVHN	Error rate: 1.1%

1. Cost function-based models: In this category, models use a cost function to estimate how good or bad the model is doing. The generative adversarial network follows this category.
2. Energy-based models: In this category, models define the probability density function using an energy function, that works as an indicator of the configuration of the variable, the lowest value the energy function produces, the more it indicates that the system is suitable.

Fig. 2 shows the main types of generative models.

Generative models have existed for a long time; even though the generative models have many benefits, they have not attracted much attention, due to the lack of the needed computational power, and the complication of implementing the structure of such models [29]. The importance of the generative models exceeds the limitation of generating fake samples. They can be useful in a variety of applications such as Handling missing data, like [30], which trained an image completion network to fill the missing parts of images. Another usage is enhancing data quality such as in [31], where blurry images have been enhanced to a clear high-resolution image using the generative model.

3.2. MiniMax algorithms

Decision-rule based algorithm originally used in game theory, where two players are playing a turn-based game, to maximize the current player gain, while minimizing the opposite player gain at the same time. In turn-based games, each player has no control on the other player moves, which arises the need of using adversary methods. Adversarial training is a discipline of machine learning used when another model or an optimization algorithm enters the worst case input for the model. In the minimax algorithm, each player aims to make the worst possible scenario for his opponent. An evaluation function determines the gain in each game, each player makes the best possible move, in which his evaluation score is maximized and the opposite player score is minimized. An optimal solution for two optimal players would be finding a point where the current player maximum gains are the same as the opposite player's minimum gains; such point is called a saddle point, also known as equilibrium.

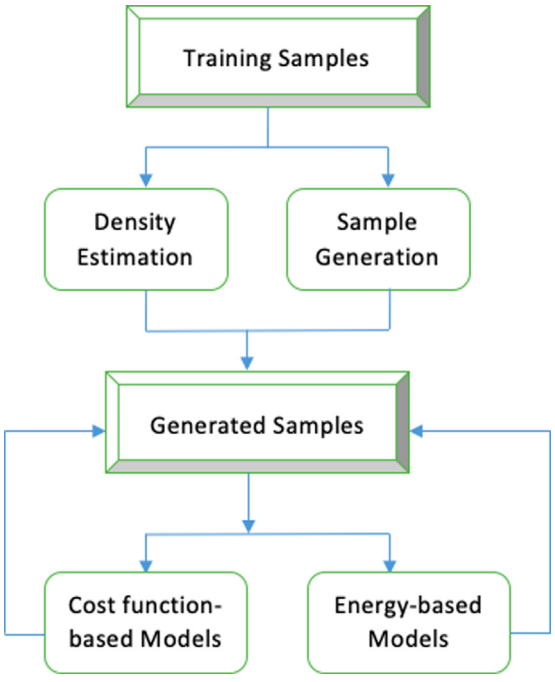

Fig. 2. Generative models main types.

3.3. Generative Adversarial Networks (GANs)

GANs was first introduced by Goodfellow et al. in 2014 [32], the basic idea is to simultaneously train two deep neural networks by allowing them to play a game together [33]. The first network is called the generator (G), responsible for generating fake sample by learning the distribution of the training samples. The second network is the discriminator (D), which is responsible for evaluating the quality of the generated samples by applying traditional supervised learning, to distinguish real from fake samples.

The generator takes random noise (z) as input to allow the variation of the generated samples, then apply the learned distribution from training samples on z to generate fake sample $G(z)$, each generated sample will be evaluated by the discriminator, which uses the real samples to learn the distinguishing features of the real samples, then take real and fake samples, and give each sample a probability of being real, probability of 1 means that the discriminator is sure the sample is real, while 0 means the sample is fake. The generator objective is to confuse the discriminator, and have all the samples given the probability of 0.5. Fig. 3 shows an illustration of the process.

At the beginning of the training, the generated samples will be easily distinguished by the discriminator, which allow the generator to learn from his previous mistakes and generate better samples, forcing the discriminator to learn deeper features of the real sample in order to be able to distinguish them, with the passage of time, the generator will produce more realistic samples, and the discriminator will be smarter at distinguishing real samples. In other words, both networks will be training each other by playing a game as opponents, each network tries to make the worst possible move for the other network. The objective function used to evaluate their moves is shown in Eq. 1:

$$Min_{(G)}Max_{(D)}V(D,G) = E_{x\sim P_{data}}[\log D(x)] + E_{Z\sim Pz(z)}[\log(1-D(x))] \quad (1)$$

The networks will keep training until they reach a saddle point, where both networks gains are equal, during the training, the discriminator aims to minimize the value of $D(G(z))$ and maximize the value of $D(x)$, while the generator aims to maximize the value of $D(G(z))$. Fig. 4 shows the evolution of the generator and discriminator through training.

Fig. 4 Goodfellow et al. [32], shows the progress of the generator and the discriminator through the training process. Where the dotted black line is the real sample distribution (P_{data}), the green line is the distribution estimated by the model(P_{model}), and the dotted blue line is the discriminative distribution. The first image at the left represents the beginning of the training, as illustrated the differences between P_{data} and P_{model} is obvious, and the discriminator values are varying greatly between *rea* and *fake* samples. At the end of the training as illustrated in the last image, P_{data} and P_{model} are identical, in which the discriminator values are stable, which means that the discriminator cannot distinguish real from fake samples anymore.

3.4. Generative adversarial networks training

GANs have many advantages and can be used in many different applications because of the good quality of the generated samples, but it comes with disadvantages as well, it can be very hard to train the GANs, since both the generator and the deep discriminator networks have to be trained simultaneously without allowing one network to crush the other, the unstable behaviour in GANs learning process limits the ability to experiment new variants [34].

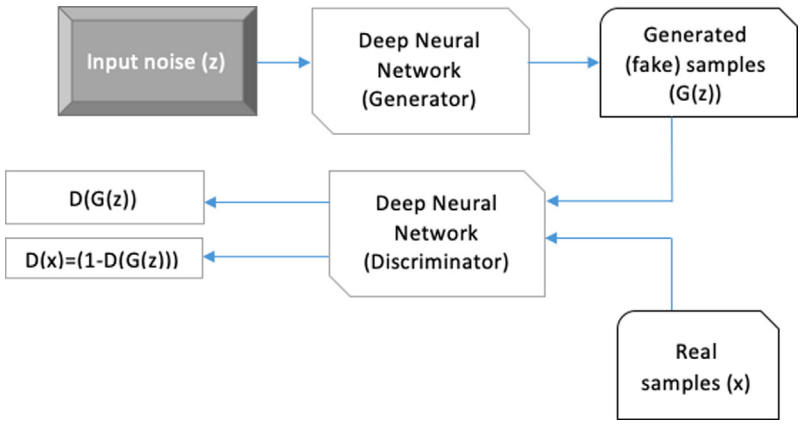

Fig. 3. Illustration of the generator and discriminator.

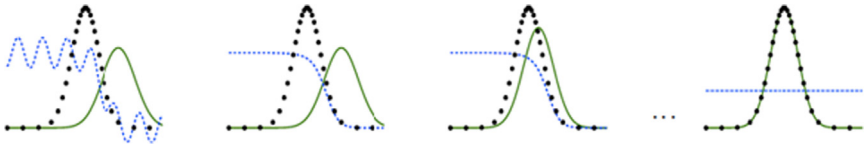

Fig. 4. Goodfellow et al. [32], progress of the generator and the discriminator through the training process.

Some of the complications that might arise during the training process are mentioned here:

1. Vanishing gradient: This problem usually arises at the beginning of the training, when the generator still weak, and the discriminator is sure about the real samples, which cause the generator learning to stop.
2. Mode collapse: Most of the distributions learned by the GANs are multimodal (have more than one peak), which limit the variation of the generated samples.
3. Nash equilibrium: The equilibrium point is not easy to find, especially in a two players non-cooperative game, where each player aims to minimize the opposite player gains. Moreover, in some cases, one of the players stops making any changes regardless of the opponent moves, which arise the Nash equilibrium problem.
4. Evaluation Measures: Several measures have been introduced for evaluating the GAN performance, each of these measures has its weaknesses and limitations [35], Most of the measures are developed for a specific kind of GANs. Thus none of them can be used for a fair comparison between all GANs variations.
5. To overcome the previous complications, researchers start to design new architectures, and procedures to make the training process more stable. [36] proposed some solutions such as feature matching, where the generator specified with a new objective, in which generate fake samples to match the real data statistics, then the discriminator decides if these statistics are worth matching or not.

3.5. Generative adversarial networks variants

- **Deep Convolutional Generative Adversarial Networks (DCGANs)**

A class of CNNs is proposed by [37] with specified architectural constraints, to allow more stable training for the GANs. The first constraint is replacing all the convolutions layers with stridden convolutions, which allows the network to learn its own subsampling. Second, eliminate all the fully connected layers that come after the convolutional layers, to increase the model stability. Third, using the batch normalization, which normalizes the input to each unit to have zero mean and unit variance in order to stabilise the learning. Batch normalization was applied to all the layers except for the discriminator input layer and the generator output layer, to avoid sample oscillation and model instability. As for the generator activation function, the *ReLU* activation is used for all layers except the output layer which uses the *Tanh* function, and for the discriminator, *LeakyReLU* activation is used. Fig. 5 shows an example of generator architecture.

In the [38], the authors introduce introduced the conditional version of the GANs, in which both the generator and the discriminator are conditioned based on an additional input y. y can be any data; usually, it is the class labels of the training data. In the generator, y is combined with the prior input noise $Pz(z)$ in joint hidden representation, while y in the discriminator, is presented as input with x. Eq. 2 shows the objective function of the conditional GANs. CGANs are useful for controlling the classes of the data being generated. Moreover, CGANs can be used to generate descriptive tags that are not in the training data labels. Fig. 6 show visualization of a simple CGAN.

$$Min_{(G)}Max_{(D)}V(D,G) = E_{x\sim P_{data}}[\log D(x|y)] + E_{Z\sim Pz^{(z)}}[\log(1 - D(x|y))] \quad (2)$$

- **Energy-Based Generative Adversarial Networks (EBGANs)**

A model that combines energy-based GANs with auto-encoders was proposed by [39], where the discriminator is viewed as an energy function. The energy function is a mapping function of each input point with a scalar value called the energy; the lower energy value is the better. The discriminator energy function can be viewed as the generator cost function, since the good generated sample is given low energy value, in contrast with the bad sample which is given high energy value. In energy-based GANs the discriminator aims to give the fake samples high energy values, while the generator aims to generate fake sample in the regions where

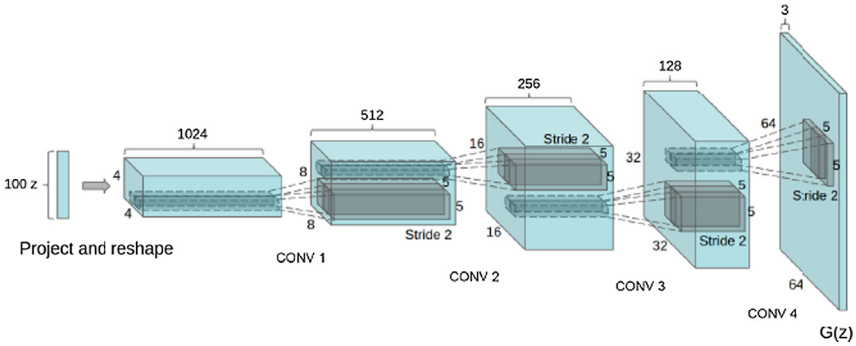

Fig. 5. DCGAN Generator Architecture [37].

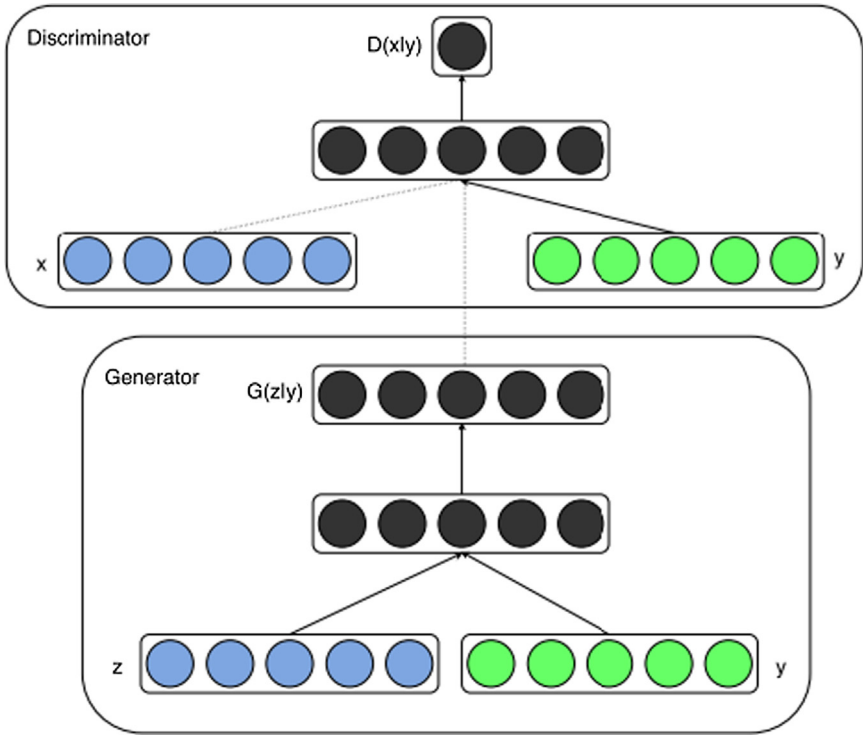

Fig. 6. Illustration of a simple CGAN.

the discriminator gives low energy values. The loss functions for the generator and the discriminator are defined in Eqs. (5) and (6), the model use different losses to get better quality gradients for the generator.

$$\iota_D(x,z) = D(x) + [m - D(G(z))] \tag{3}$$

$$\iota_G(z) = D(G(z)) \tag{4}$$

where m: margin loss Traditionally, auto-encoders have been used to represent energy-based models, since they allow the model to learn the energy manifold on its own, which means the discriminator can learn the data manifold without supervision. EBGANs showed better scalability and convergence pattern than the original GANs. A visualisation of EBGAN is showed in Fig. 7.

4. Our proposed approach

This section describes the methodology we aim to use in this research, Fig. 8 presents the general steps we will follow; each step is explained in details later in this section.

4.1. Data augmentation

In this section, the Least Square Generative Adversarial Networks (LSGANs) used for augmenting artificial data will be explained.

Least Square Generative Adversarial Networks

In this work, we decided to use the Least Square GANs (LSGANs) [40] as it have shown a very good results in terms of both generated samples quality, and learning stability. In LSGANs, the sigmoid cross-entropy loss function for the discriminator proposed in the original GAN paper was replaced with the least-squares loss function, the reason behind this replacement is to overcome the problem of the vanishing gradient during the training.

Using the sigmoid cross-entropy loss function can cause the vanishing gradient problem, when updating the generator using the augmented data on the right side of the decision boundary but still far from the real data. while using the least squares loss function have the ability to move the augmented samples toward the decision boundary, since its penalty the samples on the right side based on the distance from the boundary, see Fig. 9 for more explanation.

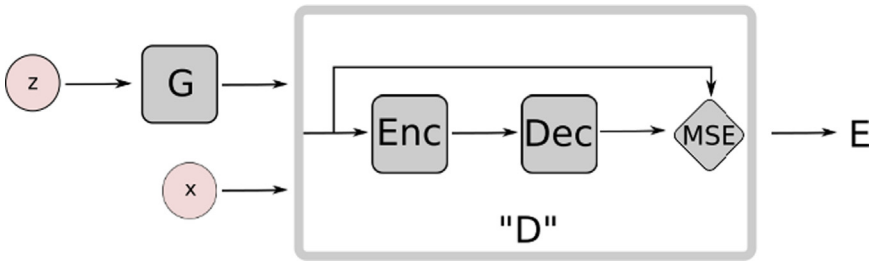

Fig. 7. EBGAN with auto–encoder architecture [30].

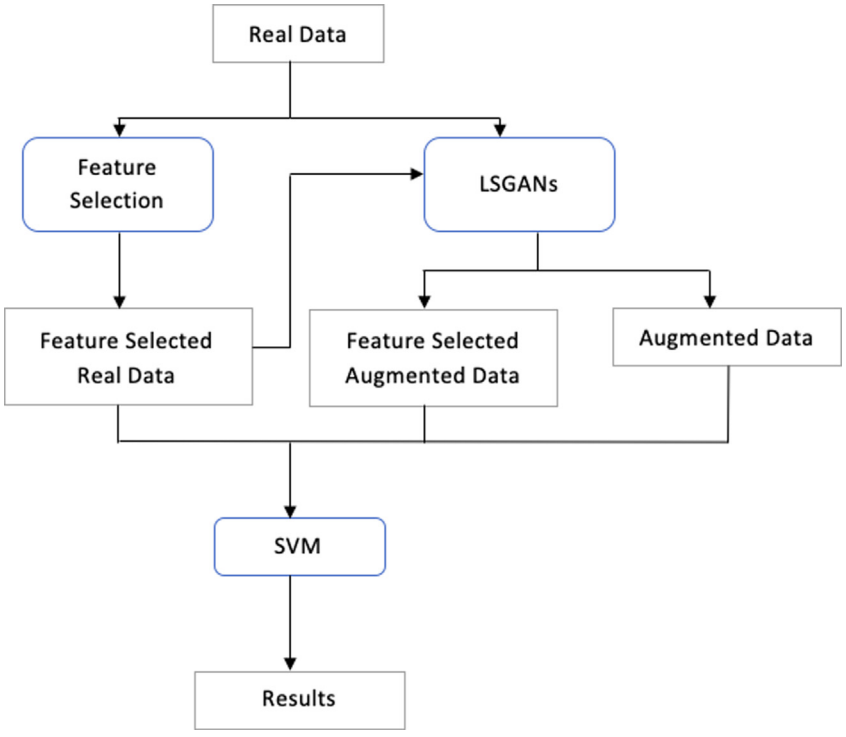

Fig. 8. Our proposed approach.

Fig. 9 Illustration of different behaviours of sigmoid and least square loss functions. (a) shows the decision boundaries of the two-loss functions, as shown the decision boundary should go across the real data. (b) shows the decision boundary for the sigmoid loss function, in which a very small error is given to the samples with high distance from the real data. (c) shows the decision boundary of the least square function, where the far augmented samples are penalized which force the generator to generate samples closer to the real data points.

The LSGANs objective function is showed in Eqs. (5) and (6).

$$Min_{(D)}V_{LSGAN}(D) = \frac{1}{2}E_{x\sim P_{data}}[D(x)-b^2] + E_{Z\sim P_{Z^{(z)}}}[(D(G(z))-a)^2] \quad (5)$$

$$Min_{(G)}V_{LSGAN}(G) = \frac{1}{2}E_{z\sim P_{pz(z)}}[(D(G(z))-c^2)] \quad (6)$$

where: a: Augmented data. b: Real data. c: The value that G wants D to believe for fake data.

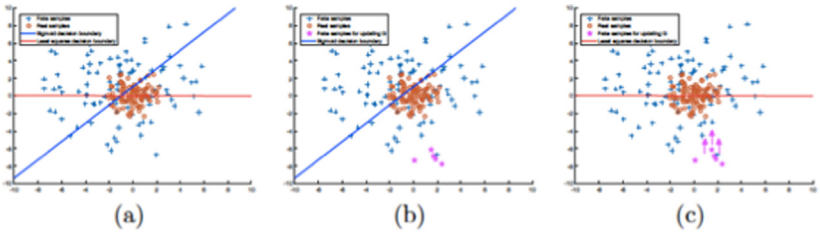

Fig. 9. Illustration of different behaviors of sigmoid and least square loss functions [40].

4.2. Feature selection

Feature selection is the process of selecting the most relevant subset of features from the complete set of features in the dataset without losing important information, by discarding the irrelevant or less helpful features in solving the problem at hand [41]. Relevant features hold the most useful information for the predictive model while the irrelevant features are redundant, noisy or completely useless features. The feature selection process is considered very important since it reduces the memory storage, training time, computational cost and increases the performance of the predictive model [42].

Feature selection is categorized into three main classes: First, the wrapper methods, that considers feature selection as a search problem, where it tries different combinations of features and evaluate the quality of each combination using a predictive algorithm. Second, the filter methods, which deals with feature selection as a pre-processing step, where it uses statistical methods to associate each feature with a score, rank them based on this score, and then use the features with the highest scores in the predictive algorithm. Finally, the embedded methods, where the feature selection process is integrated with the prediction algorithm as a part of the learning phase, in each iteration the model learn which combination of features improves the prediction algorithm performance.

In this research, we will use one of the wrapper methods work by repeating two steps [43], the creation of a subset of features, and then evaluates the combination of features in the subset. These two steps keep repeating until a stopping criterion is satisfied; either by obtaining a required performance, or completing a defined number of iterations.

Exhaustive search (brute-force) is an example of wrapper methods that guarantee to find the best subset of features by trying every possible combination of features, but it is not commonly used because of the expensive computational power it requires, which makes it not feasible for datasets with a large number of features. Therefore, methods that are more intelligent are have been developed for better performance in terms of computational power.

In our research, we started the feature selection phase by removing the features that depend linearly on other features, then apply a feature selection algorithm called the boruta algorithm.

First, the correlation measure was used to compute the strength of the linear relationship between features. If the correlation is too high, then they are dependent, and one of the features had to be eliminated, therefore, we eliminate one of the features that have correlation value more than 0.75 or less than −0.75. Second, we applied the boruta algorithm, which was first introduced in 2010 by Jankowski et al. [44], it is an improvement of using random forest classification for the feature selection purpose. Random forest classification is constructed by combining multiple decision trees, where each decision tree represents an individual classifier. Each tree uses different subsets of the original objects and features in the dataset. Based on the classification results of each tree, a score is computed for each feature [45]; the most useless features are eliminated in each iteration of the learning phase.

The main idea behind boruta algorithm is that each feature is duplicated, and the values are randomly shuffled, these new features are called shadow features. The original features and the shadow features are combined and fed to the random forest algorithm. Table 2 shows the original features and their duplicated shadow features.

In each iteration, the shadow features values are shuffled and a score is computed for each feature. If an original feature score is higher than the maximum score of the shadow features, the feature will be selected, but if the original feature score is lower or equal to its shadow feature score is considered unimportant. Thus the feature will be eliminated with its duplicated shuffled version. The algorithm stops when all the features are classified as selected or not selected, or when a pre-defined number of iteration is completed, Fig. 10 present the boruta algorithm (see Table 3).

The boruta algorithm parameters where specified as follow the random forest classifier depth equals to 5, run the algorithm for 100 iterations and the 'perc' function which define a quaintly instead of maximum to compare between original and shadow features as we can see in Table 3, was specified with 85. Fig. 11 shows the feature selection steps used, and the number of the selected features.

4.3. Classification

Classification is a class of supervised machine learning, in which the implemented model learns from labeled data points to approximate a mapping function between the dataset features to the discrete labels, then predict the label or class to a new unseen data points. Classification learners are divided into two types:

1. Lazy Learners: this type of classification store the training data, and compare the entered data points to classify it based on the most similar training data point.
2. Eager Learners: this type of classification construct a predictive model based on the training data, and use this model to predict the labels for new data points.

Support Vector Machine (SVM) is an eager learner classification algorithm first introduced in 1995 by [46], which develop a predictive model based on existing labelled dataset. Each sample in the dataset is represented by its features in an n-dimensional space, where n is the number of features in the dataset.

The SVM learning process is done by finding the optimal hyperplane with maximum possible margin the distance between the nearest data point and the hyperplane – [47], that separates the feature vectors based on their classes. The chosen hyperplane serves as the decision boundary for the classification problem. Both linear and non-linear separation rely on finding the optimal hyperplane; only the non-linear datasets will have to use the help of a kernel function [48].

A kernel function is used for mapping the non-linear datasets into a higher dimensional space to ease the mission of finding the optimal hyperplane. The most suited kernel function must be chosen based on the existing training dataset distribution.

Another important value for the SVM is the Complexity parameter (C), which is used to control the number of allowed misclassifications to increase the margin value which leads to finding a better hyperplane. For testing the model performance, we used

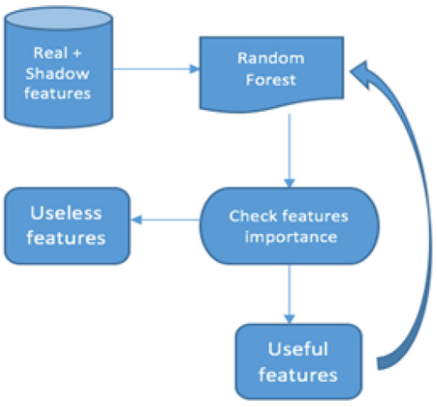

Fig. 10. Boruta Algorithm.

Table 3
Shadow Features.

Original Features		Shadow Features		Label
F1	F2	F1	F2	
59.58	78.96	74.67	82.52	2
74.67	142.60	64.15	78.96	0
59.19	82.52	59.58	80.38	1
64.15	80.38	59.19	142.60	3

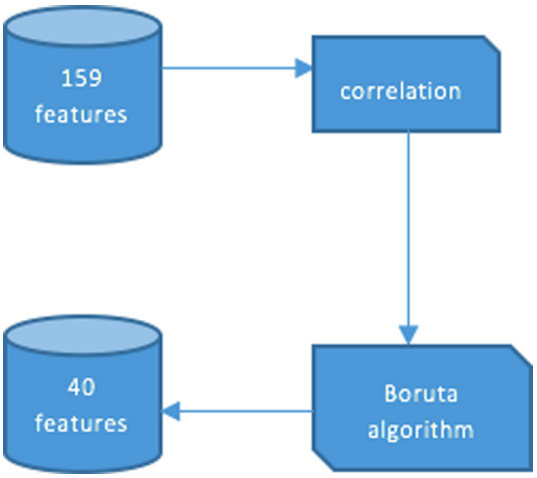

Fig. 11. Features Selection steps.

the cross-validation technique which works by dividing the dataset into two sub dataset, training dataset, and testing dataset. The model is trained on the training dataset; then, the unseen testing dataset is used to evaluate the performance of the model.

4.4. LSGANs limitation

LSGANs uses the least square loss function in which mainly focus on generating data within the decision boundaries, meaning that it will not penalise the data as long as it is on the right side of the boundary, even if it is far from real data points.

The problem with this approach, in this case, is the nature of our dataset distribution, as shown in Figs. 13–17, the data variance is very large and outliers are strongly present, which is likely to happen in medical datasets because human bodies do not act the same, which means that each body has his unique reactions to pain. The expected distribution of the augmented data is to be much closer to a normal distribution than the real data when using the least square loss function, which means that the variance of the augmented data will be smaller than the real data variance (see Fig. 18 and 19).

Even though the augmented data boundaries separate data points with different labels more efficiently than the real data points, it arises the problem of obliteration the uniqueness of each data point values. The least-square function shown in Eq. (7), square the difference of the error between the real value and the predicted value which leads to a much larger error especially with outliers, so we need to modify the loss function to consider outliers, we cannot replace the square with the absolute value because it cannot be differentiated. Also we cannot remove the outliers from the dataset as they represent more than half of it

$$LS - Lossfunction : \sum_{i=1}^{n} (Y_{true} - Y_{predicted})^2 \tag{7}$$

In this research, we are aiming to make a change on the least square loss function, such that it takes into consideration the distance from the real data points beside the distance from the boundaries. The logarithmic loss function in contrast to the least square function, penalises the generated data points based on the distance between generated and real data points, without taking into consideration the boundary of the class. Another way of enhancing the performance of the least square function is to take the log of each point error, as the log loss function considers the distance from the real data points, we are aiming to make a balance between the distance between the boundary and the real data points, Eq. 8 shows the loss function.

$$LSLossfunction : \sum_{i=1}^{n} \log((Y_{true} - Y_{predicted})^2) \tag{8}$$

We also propose taking the square root of the error for each data point as in Eq. 9, so that the error will be smaller, this approach will help with improving the performance of the loss function with the outliers' data points, which are considered the majority of our dataset.

$$LSLossfunction : \sum_{i=1}^{n} \sqrt{(Y_{true} - Y_{predicted})^2} \tag{9}$$

In the end, we propose taking the square root of each error, to enhance the stability of the training by minimising the errors as much as possible, then take the log of the resulted value, to

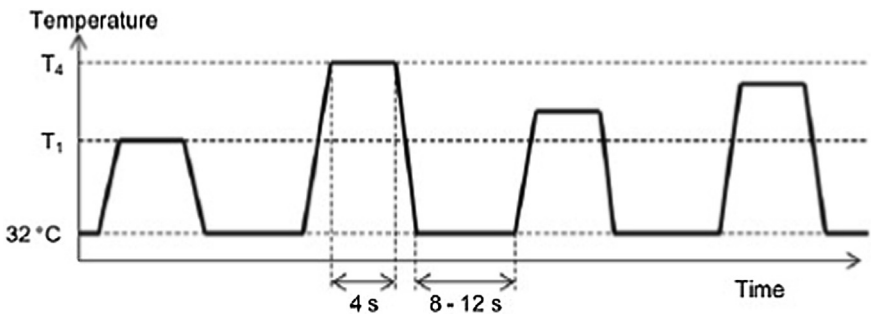

Fig. 12. Accuracy obtained from classification using only the real data with all features.

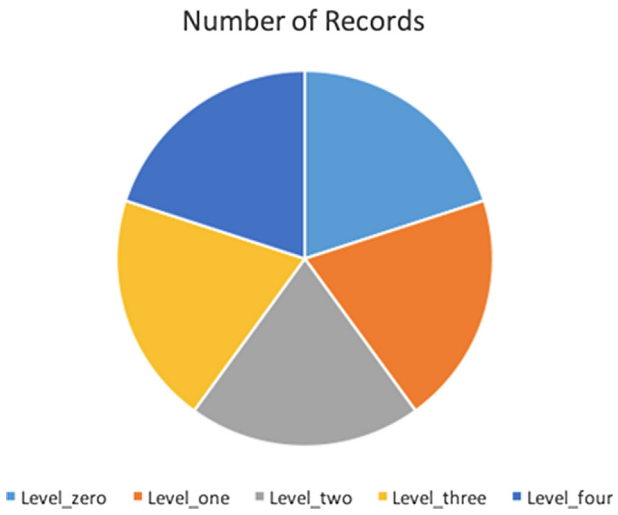

Fig. 13. Number of records for each unique value in the 'Label' column.

enhance the quality of the generated samples, this way we take advantage of both Eq. (8) and (9), the proposed loss function is showed in Eq. (10):

$$LSLossfunction : \sum_{i=1}^{n} \log(\sqrt{(Y_{true} - Y_{predicted})^2}) \quad (10)$$

5. Experiments and results

5.1. Dataset description

The dataset used in this research was obtained from an experiment consist of 85 healthy participants of different ages and genders exposed to painful heat stimuli by attaching the ATS thermode to each subject right forearm [26]. The participants were asked to report when they reach both the pain threshold – where subject start to feel pain instead of just the sensation of heat – and pain tolerance threshold – where subject cannot tolerate the pain anymore – both records were saved as T1 and T4 then two other levels were calculated in between T2 and T3. See Fig. 12 below.

After saving the temperatures that cause the four pain levels, each subject was exposed to these temperatures randomly 20 times with pauses between each stimulus, in this way, each subject has a total of 80 stimuli.

The biopotential signals saved from participants in this experiment are: Electromyography (EMG): electrical signals produced by the body muscles cells, the signals of zygomaticus, corrugator, and trapezius muscles were recorded. The sensation of pain causes an involuntary muscle contraction, EMG signals changes due to this contraction. Skin Conductance Level (SCL): this signal is measured by attaching two electrodes to the skin to measure the electrical flow between two points of the skin when the body feels pain the skin becomes a better conductor of electricity and based on that the SCL signals are changed when the subject feels pain. Electrocardiogram (ECG): This signal indicates how the heart is functioning by measuring the electrical activity of the heart muscle. Healthy hearts signals have specific characteristics, causing any abnormalities in the heart to produce different shape of ECG signals, when the body feels pain, the heartbeats rhythm change causing changes in the ECG signal. After recording these signals for the four levels of pain besides the no pain level signals, mathematical equation belongs to six groups: amplitude, frequency, stationarity, entropy, linearity, variability, and similarity were applied on these signals extracting 159 features.

5.2. Exploratory Data Analysis (EDA)

EDA is the process of exploring our dataset, investigating its characteristics, and patterns, and discover missing values and outliers. EDA can be supported with visualization to improve the understandability of the dataset.

First, our dataset contains 8500 rows and 161 columns. All the columns are of type float except for the subject ID column which was dropped, and the class column which is called 'Label' with type object, the unique value of this column is: 'level_zero', 'level_one', 'level_two', 'level_three', 'level_four'.

The records are divided equally between all the unique values, as shown in Fig. 13. In order to make the predictive model be able to understand the labels values, we replace the text values with numerical values, as shown in Table 4.

Second, we had to check if the dataset contains missing values, which occur when no data found for a variable in observation; this usually happen due to human error or unavailability of the data. In our dataset, we found two columns containing missing values, the first with 48 records, and the second with 3 records, the rest of the variables were complete with 0 missing values, Fig. 14 show the number of missing values in each variable. The missing values were handled by filling them with the mean of each column.

Third, observations that have values which differ significantly from other observations in the same dataset. Outliers can be detected by multiplying the Interquartile Range (IQR) by 1.5, the values higher than the summation of the multiplication result and the third quartile, or less than the summation of the multiplication result and the first quartile is considered an outlier. In this dataset, we decided to keep the outlier, because as shown in Fig. 15, almost every variable contain outliers, and some of them have more than 2000 outlier, if these values are dropped or changed, a major content of the data will be ignored.

5.3. Results

In this section , the results obtained from the experiments done in this research will be presented. The performance of the pain intensity recognition problem is usually measured by calculating the accuracy of the classification between no-pain level and each pain level as binary classification, but we are going to measure the accuracy for the classification of all the pain levels together

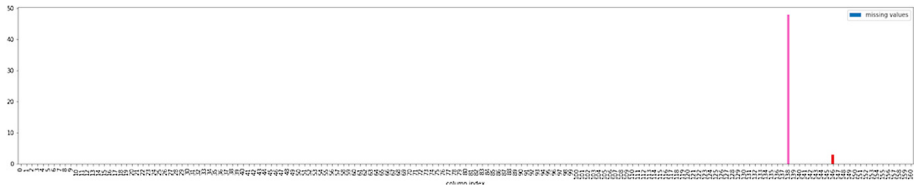

Fig. 14. The number of missing values in each column in the dataset.

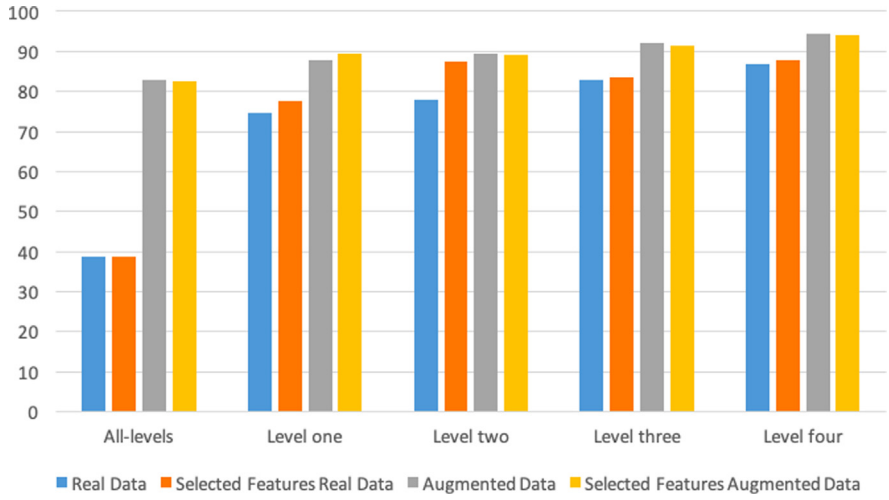

Fig. 15. The number of outliers in each column in the dataset.

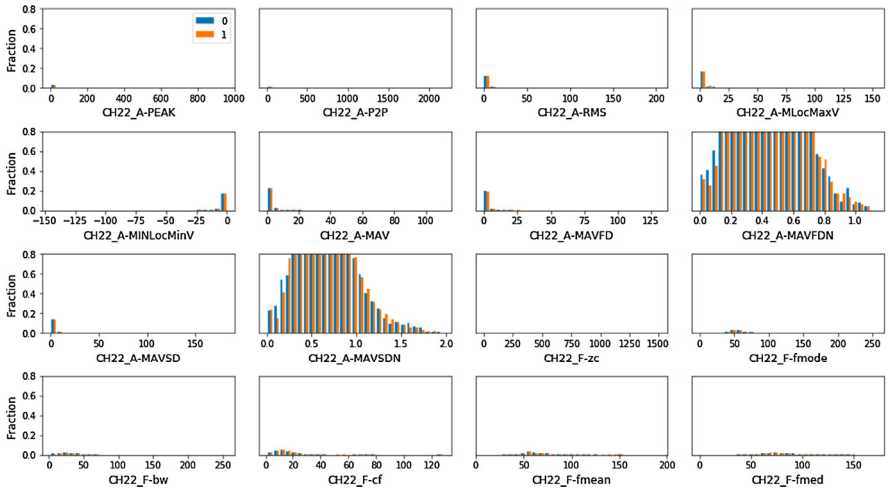

Fig. 16. Real data distribution by features between no pain and level 1.

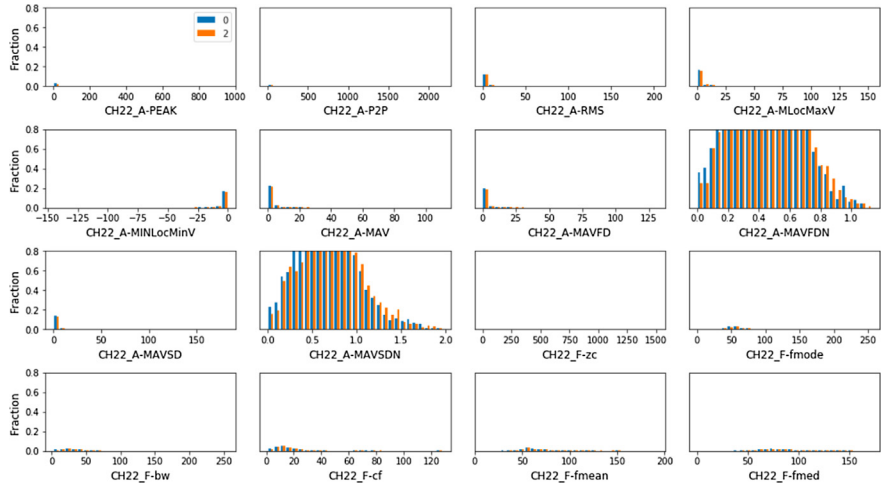

Fig. 17. Real data distribution by features between no pain and level 2.

as well. In this work, we build the SVM model using scikit-learn library in python, in order to obtain the optimal hyperplane, we used the Radial Basis Function (RBF) as a kernel function, and the C value was assigned with 1.0. As for the model evaluation, the model was trained on 75% of the data, and the accuracy was calculated on the remaining 25% on the data. First, we start our experi-

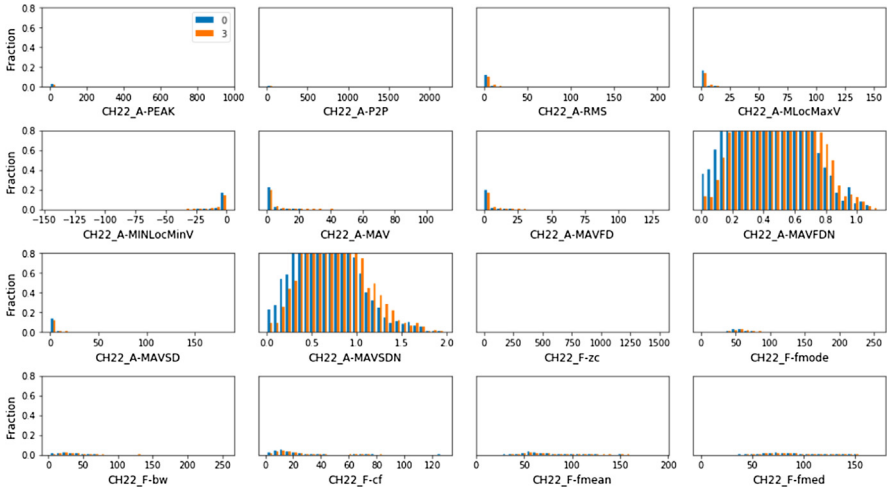

Fig. 18. Real data distribution by features between no pain and level 3.

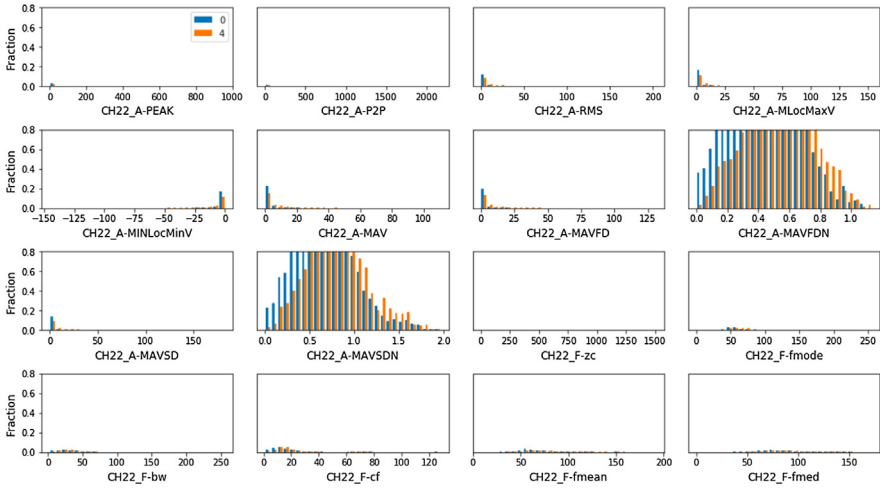

Fig. 19. Real data distribution by features between no pain and level 4.

Table 4
The original unique values of the 'Label' column, and their replacements.

Old Value.	New Value.
level_zero	0
level_one	1
level_two	2
level_three	3
level_four	4

ments by making the classification using only the real data, with all the existed features. The accuracy for all the levels was 38.6, as for the accuracy for each level of pain are from level one to level four are 74.5, 78, 82.9, and 86.8. The results are shown in Fig. 20, the results showed that the accuracy improved when the pain level is higher. Second, we made the classification using the 40 features obtained after applying the feature selection phase. The accuracy for all the levels was 38.6, as for the accuracy for each level of pain are from level one to level four are 77.7, 87.4, 83.4, and 87.7. The results are shown in Fig. 21. Comparing to the classification using all the features, the accuracy of classifying al the level of pain did not change after the feature selection phase, even though all the results of binary classifications was improved when was done using only the selected features (see Fig 22–25).

Third, after doing the experiments using only the real data, we start the data augmentation process using LSGANs. The implementation of the LSGANs was done using the rivalgan library, which is written using TensorFlow library in python. Rivalgan library was implemented to generate numerical data using different types of GANs. The library was customized to suits the pain intensity dataset.

As GANs does not distinguish the class variable and consider them all dependent variables, we partitioned the dataset into five smaller datasets based on the class, then we fed each dataset alone to the LSGANs, and generate 5000 sample of each class to have 25000 augmented samples, and 8500 real samples making 33500 samples in total, the distribution of the augmented samples are displayed in the figures below.

Using both real and augmented data, we made the classification to get an accuracy of value 82.8 for all levels classification, 87.9, 89.4, 92, and 94.5 for classifying pain from level one to level four respectively as shown in Fig. 26.

Fourth, we generated data with only the remained features after applying the feature selection phase, the same number of samples generated with all the features were generated using the selected features only. Fig. 27 shows the accuracy obtained from making

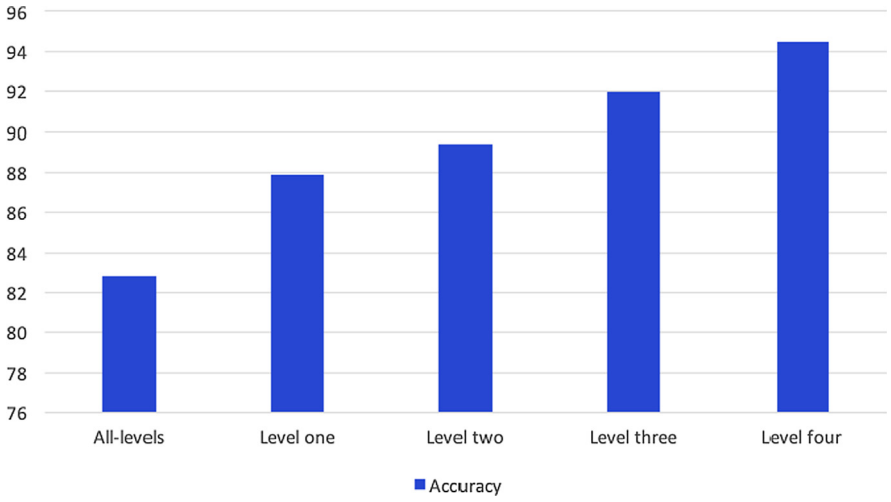

Fig. 20. Accuracy obtained from classification using only the real data with all features.

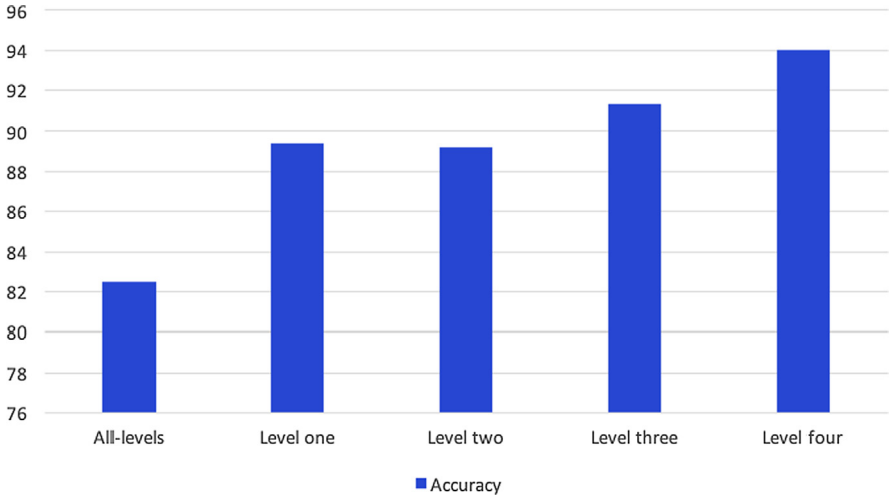

Fig. 21. Accuracy obtained from classification using only the real data with the selected features.

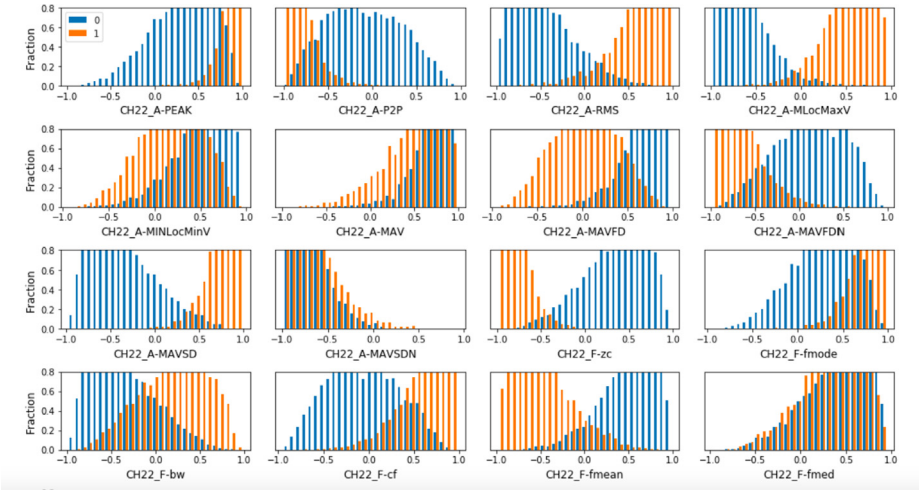

Fig. 22. Augmented data distribution by features between no pain and level 1.

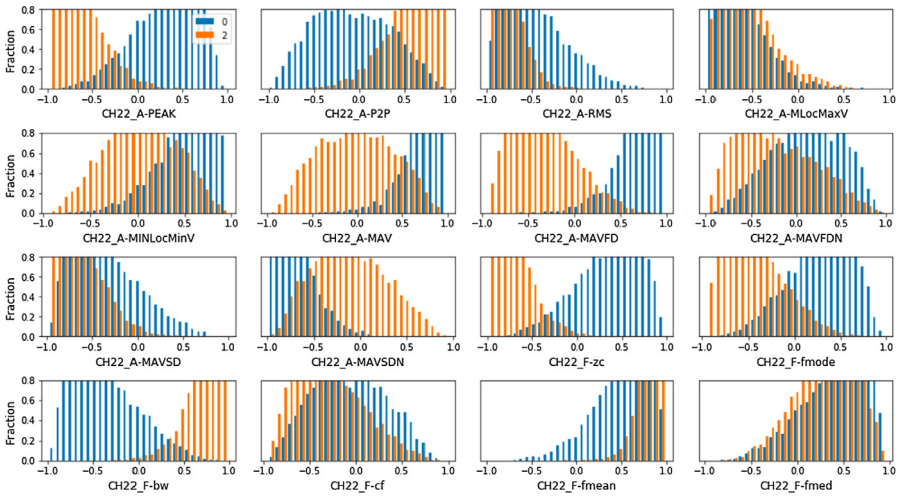

Fig. 23. Augmented data distribution by features between no pain and level 1.

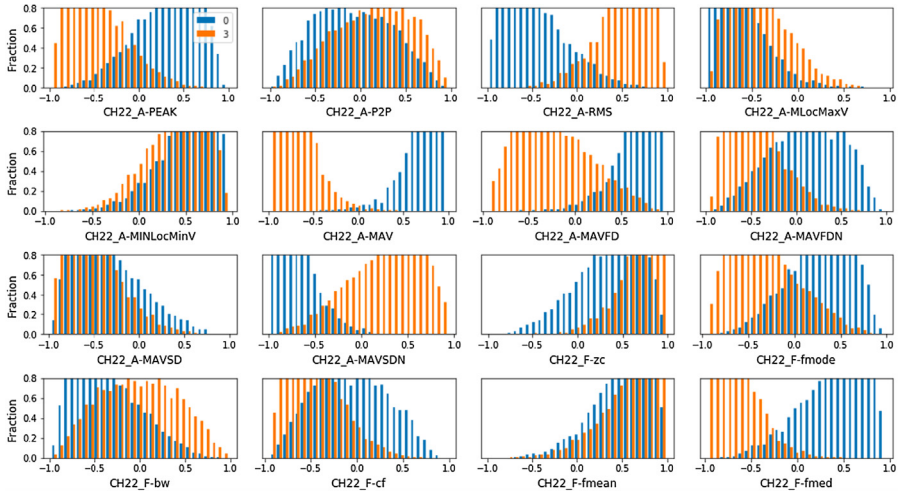

Fig. 24. Augmented data distribution by features between no pain and level 1.

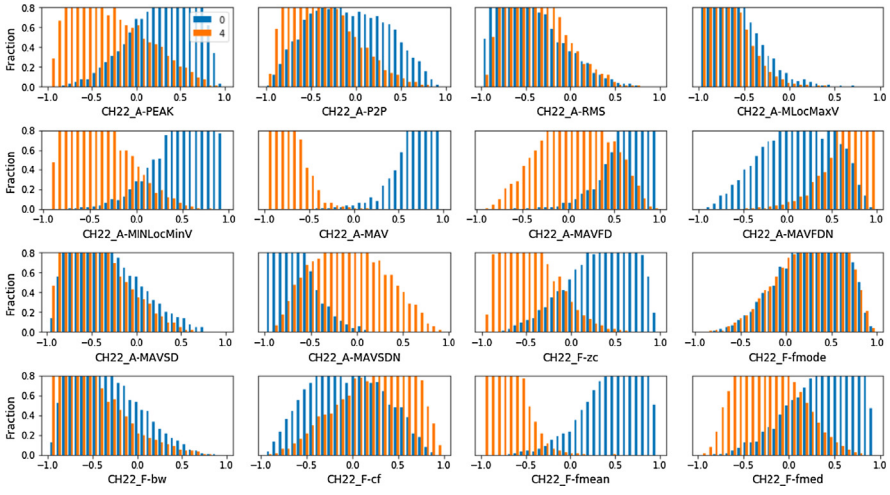

Fig. 25. Augmented data distribution by features between no pain and level 1.

the classification on the selected features data. Classifying all levels of pain got an accuracy of 82.5, as for classifying pain from level one to level four are 89.4, 89.2, 91.4, 94.

Augmented data with the selected features showed fewer accuracy values than the augmented data with all the features, except for the classification of level one, which lead us to the conclusion

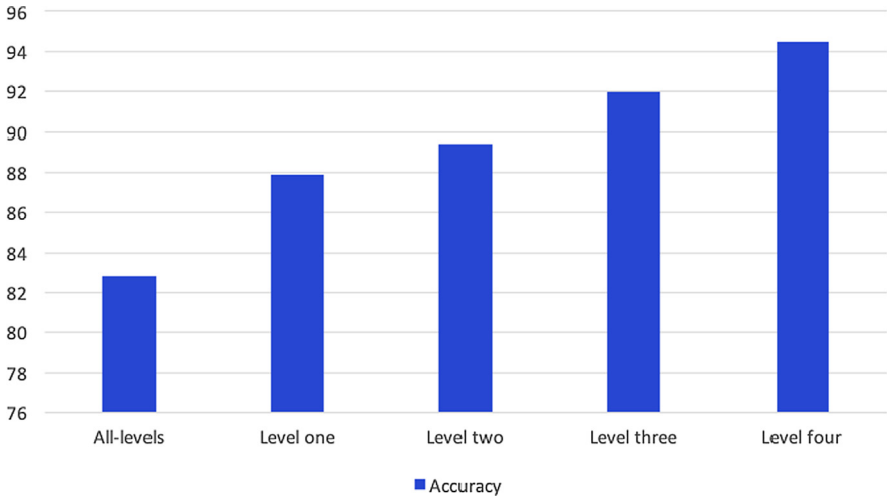

Fig. 26. Accuracy obtained from classification using augmented and real data with all features.

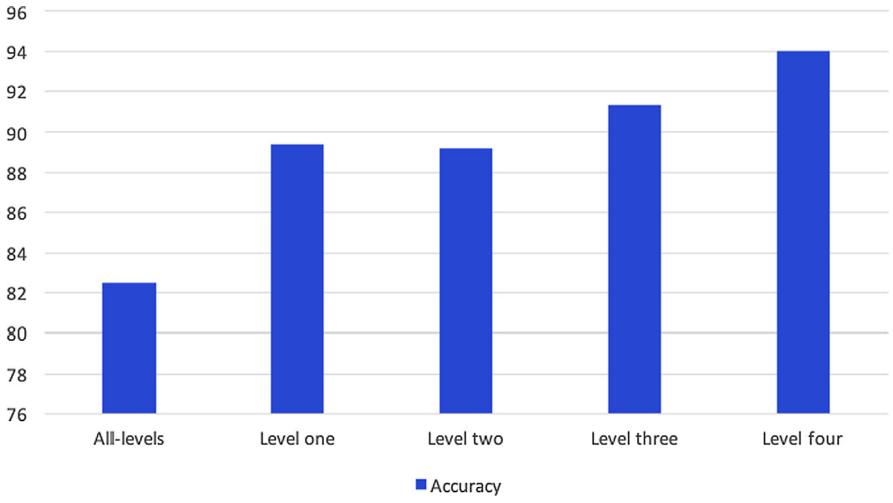

Fig. 27. Accuracy obtained from classification using augmented and real data with the selected features.

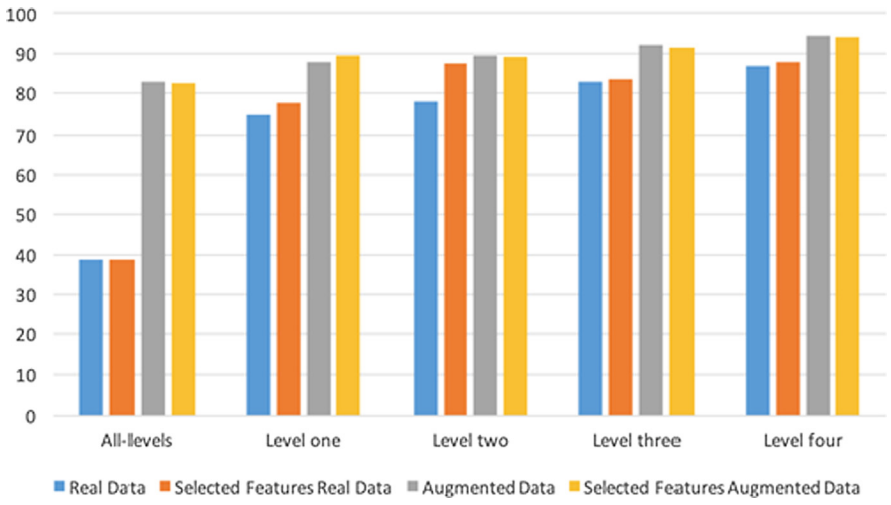

Fig. 28. Comparison between the results obtained from the experiments.

that LSGANs performs better when trained on the whole dataset instead of only part of it.

In the end, we made a comparison between all the experiments shown in Fig. 28 from the comparison it was noticed that using the augmented data in the classification have significantly improved the performance of the model, the best accuracy values were obtained from making the classification using the augmented data from all the features existed in the dataset.

6. Conclusion

Recognizing the pain intensity level is crucial for identifying the best treatment plan suitable for the patients by the caregiver, but the current methods used for pain assessment are not capable of identifying the exact level of pain accurately. Thus, it became important to automate this process and predict the pain level more accurately. Training a machine learning model to be able to predict the pain level requires a huge amount of data which is can be a problem, especially in medical data. Therefore, we generated an artificial data similar to the existed data using Least Square Generative Adversarial Networks. We have implemented predictive model using the Support Vector Machine algorithm, and fed it with the augmented data beside the real data, and got 82.8 accuracy for classifying all the levels of pain together which is 44.2 higher than the accuracy using only the real data. the feature selection phase on the data reduce the time need to build the model. Moreover, the accuracy is improved with the real data, and degraded when using augmented data only.

References

[1] B.A., Pros and cons of gan evaluation measures., Computer Vision Image Understanding 179 (2019) 41–65.
[2] P.A., Support vector machine – a survey, Int J Emerging Technol Adv Eng (2012) 82–85.
[3] A.M., B.L., Towards principled methods for training generative adversarial networks, ICLR (2017) 1–17.
[4] H.S.E.J. Bodian C., Freedman G., B.Y., The visual analog scale for pain: Clinical significance in postoperative patients, Anesthesiology 95 (2001) 1356–1361.
[5] H.J. Chu Y., Zhao X., S.Y., Physiological signal-based method for measurement of pain intensity., Frontiers in Neuroscience 11 (2017) 1–13.
[6] C.C., V.V., Support-vector networks. kluwer academic publishers (1995) 273–297.
[7] M.D. e. a. Cubuk E., Zoph B., Autoaugment: Learning augmentation strategies from data, CVPR (2019) 113–123.
[8] M.M. e. a. Goodfellow I., Pouget-Abadie J., Generative adversarial nets, ArXiv (2014) 1–9.
[9] W.P.T.H.C.S. e. a. Gruss S., Treister R., Pain intensity recognition rates via biopotential feature patterns with support vector machines, PlosOne 10 (2015) 1–14.
[10] G.I., E.A., An introduction to variable and feature selection, J Mach Learn Res 3 (2003) 1157–1182.
[11] H.G., S.J., The psychology of pain, Elsevier (2005).
[12] S.I.A.P. Ho D., Liang E., C.X., Population based augmentation: Efficient learning of augmentation policy schedules, ICML (2019) 1–14.
[13] C.C. Hsu C., L.C., A practical guide to support vector classification, Theory, Culture and Society (2008) 1–16.
[14] G.I., Nips 2016 tutorial: Generative adversarial networks, NIPS (2016) 1–57.
[15] Craig WAK. Updating the definition of pain. Pain 2016:2420–4.
[16] A.M.S.F. Kachele M., Thiam P., P.G., Methods for person-centered continuous pain intensity assessment from bio-physiological channels, IEEE J Selected Topics Signal Process 10 (2016) 854–864.
[17] J.A. Kursa M., R.W., Boruta - a system for feature selection, Fundamenta Informaticae (2010) 271–285.
[18] K.K., E.P., Definition of pain and classification of pain disorders, J Adv Clinical (2016) 87–90.
[19] W.S.M.F.T.R. e. a. Li J., Cheng K., Feature selection: A data perspective, ACM Comput Surveys 50 (2017).
[20] K.T.K.C. Lim S., Kim I., K.S., Fast autoaugment, ArXiv (2019) 1–10.
[21] S.-S.E. Lizuka S., I.H., Globally and locally consistent image completion, ACM Trans Graphics 36 (2017) 1–14.
[22] L.-M.D., P. R, Multi-task neural networks for personalized pain recognition from physiological signals (2017).
[23] L.-M.D., P.R., Continuous pain intensity estimation from automatic signals with recurrent neural network (2018).
[24] R.O. Lopez-Martinez D., P.R., Personalized automatic estimation of self-reported pain intensity from facial expressions (2017)..
[25] P.K.S.P. Lucey P., Cohn J., M.I., Painful data: The unbc-mcmaster shoulder pain expression achieve database. (2011).
[26] M.M., O.S., Conditional generative adversarial nets, ArXiv (2014) 1–7.
[27] M.K. Mostert W., E.A., Filter versus wrapper feature selection based on problem landscape features, GECCO (2018) 1489–1496.
[28] O.A., E.A., Deep generative models: Survey (2018).
[29] M.A. Radford A., C.S., Unsupervised representation learning with deep convolutional generative adversarial networks, ICLR (2016) 1–16.
[30] Y.X. e. a. Reed S., Akata Z., Generative adversarial text to image synthesis, ArXiv (2016) 1–10.
[31] K.J. Rudnicki W., Kierczak M., K.J., A statistical method for determining importance of variables is an information system, RSCTC (2006) 557–566.
[32] Z.W. e. a. Salimans T., Goodfellow I., Improved techniques for training gans, NIPS (2016) 1–10.
[33] S.C., K.T., A survey on image data augmentation for deep learning, Journal of Big Data 6 (2019) 1–48.
[34] W.P. e. a. Thiam P., Amirian M., Multimodal data fusion for person-independent continuous estimation of pain intensity, Commun Comput Inform Sci 517 (2015) 1–10.
[35] T.P., S.F., Multi-modal data fusion for pain intensity assessment and classification (2017).
[36] K.V. Thiam P., S. F, Hierarchical combination of video features for personalized pain level recognition, European Symposium on Artificial Neural Networks, Computational Intelligence, and Machine Learning (2017) 465–470.
[37] C.G.P.L. Tran T., Pham T., R.I., A bayesian data augmentation approach for learning deep models, NIPS (2017) 2798–2807.
[38] T.C., B.H., Recent trends in deep generative models: A review (2018).
[39] L.G.T.P.Z.Y. e. a. Velana M., Gruss S., The senseemotion database: A multimodal database for the development and systematic validation of an automatic pain and emotion recognition system., Multimodal Pattern Recognition of Social Signals in Human Computer Interaction (2016) 127–139.
[40] E.H. e. a. Walter S., Gruss S., The biovid heat pain database data for the advancement and systematic validation of an automated pain recognition system. (2013).
[41] W.J., P.L., The effectiveness of data augmentation in image classification using deep learning., arXiv (2017) 1–8.
[42] G.S. e. a. Werner P., Walter S., Automatic pain recognition from video and biomedical signals (2014).
[43] H.E.L.M. Xie Q., Dai Z., L.Q., Unsupervised data augmentation for consistency training, ArXiv (2019) 1–20.
[44] P.J. e. a. Xu X., Sun D., Learning to super-resolve blurry face and text images, in: Proceedings of the IEEE International Conference on Computer Vision, pp. 251–260.
[45] M.M. Zhao J., L.Y., Energy-based generative adversarial networks, ICLR (2017) 1–17.
[46] K.G.L.S. Zhong Z., Zheng L., Y.Y., Random erasing data augmentation., arXiv (2017) 1–10.
[47] S.F. Zhou J., Hong X., Z.G., Recurrent convolutional neural network regression for continuous pain intensity estimation in video. (2016).
[48] X.H. e. a. Mao X., Li Q., Least square generative adversarial networks (2017).