IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 31 August 2022, accepted 19 September 2022, date of publication 22 September 2022, date of current version 3 October 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3208905

RESEARCH ARTICLE

Machine Learning-Based Pain Intensity Estimation: Where Pattern Recognition Meets Chaos Theory–An Example Based on the BioVid Heat Pain Database

PETER BELLMANN[1], **PATRICK THIAM**[1,2], **HANS A. KESTLER**[2], **(Senior Member, IEEE), AND FRIEDHELM SCHWENKER**[1], **(Member, IEEE)**
[1]Institute of Neural Information Processing, Ulm University, 89081 Ulm, Germany
[2]Institute of Medical Systems Biology, Ulm University, 89081 Ulm, Germany

Corresponding author: Peter Bellmann (peter.bellmann@uni-ulm.de)

The work of Peter Bellmann and Friedhelm Schwenker was supported by the Project Multimodal Recognition of Affect Over the Course of a Tutorial Learning Experiment through the German Research Foundation (DFG) under Grant SCHW623/7-1. The work of Hans A. Kestler was supported in part by the German Research Foundation (DFG) under Grant 217328187 (SFB 1074) and Grant 288342734 (GRK HEIST); and in part by the German Federal Ministery of Education and Research (BMBF) e:MED Confirm (id 01ZX1708C) and TRAN-SCAN VI—PMTR-pNET (id 01KT1901B).

ABSTRACT In general, classification tasks can differ significantly in their task complexity. For instance, image-based differentiation between vehicles and pedestrians is most likely expected to be less complex than CT-scan-based differentiation between several lung diseases. Intuitively, based on a human point of view, one can identify some classification tasks as more complex than other classification tasks. Moreover, based on expert knowledge and/or task-specific meta information, one could attempt to estimate the complexity ranks of specific classification tasks. In this work, based on the publicly available BioVid Heat Pain Database (BVDB), we experimentally confirm the intuitive assumption that the task of automated pain intensity recognition (PIR) is very challenging. Inspired by the field of chaos theory, we show that the BVDB-specific PIR task can not only be seen as highly complex, but is even identified as a classification task of chaotic nature. To this end, we apply Hao's working definition for chaotic systems and provide an experiment-based chaos check method. To validate our approach, as a non-complex counterpart, we include a task of handwritten numerals distinction. Our study provides two main contributions, i.e.: i) an enhanced understanding for the still present and – more importantly – substantial gap between the ground truth and the predictions reported by different research groups in combination with automated PIR tasks; and ii) an approach for a numerical complexity check based on chaos theory. Different research directions are discussed for future work. Note that improving PIR accuracy performance is not part of the study objective.

INDEX TERMS BioVid heat pain database, chaos theory, classification task complexity, decision trees, pain intensity recognition, physiological signals.

I. INTRODUCTION

Machine learning-specific pain assessment based on physiological signals constitutes a challenging task. Several studies indicate that it seems feasible to design robust and effective models which can reliably distinguish between a person's *no pain* and *severe pain* conditions. For instance, in [30], Werner *et al.* obtained an averaged accuracy value of 94.3% based on the X-ITE Pain Database [9], in combination with a leave-one-subject-out cross validation (LOSO-CV), with focus on the binary scenario of *no pain vs the highest electrical pain level*, using random forests. However, the distinction

The associate editor coordinating the review of this manuscript and approving it for publication was Turgay Celik.

between different levels of pain constitutes a highly complex classification (or regression) scenario that leads to unsatisfactory performances, i.e. recognition rates. By unsatisfactory, we mean that while it is possible to significantly outperform the chance level, there is still a huge gap between the ground truth and the obtained results in the literature. For instance, in [24], Thiam *et al.* obtained an averaged accuracy value of 43.89% based on the SenseEmotion Database [27], in combination with a LOSO-CV with focus on all of the four available classes (one baseline and three pain levels, i.e. a chance level accuracy of 25%), also using random forests.

Intuitively, the task of classifying different types of flowers, as for instance defined by the Iris data set [8], is much less complex than classifying different levels of pain based on physiological signals. Moreover, the implementation of well-advanced models, such as (deep) artificial neural networks (ANNs) [15], does not sufficiently close the gap to the ground truth (e.g. [21] and [23]), as for instance in comparison to several image-based classification tasks including one or even two hundred classes [1] (e.g. defined by the CIFAR-100 [13] or Caltech Birds [29] data sets). For readers that are interested in automated pain intensity recognition, we refer to the recently published survey studies, [18] and [32], which focus on ANN-based and hand-crafted feature extraction approaches, respectively.

There exist different approaches to measure the complexity of a given training algorithm (model). For instance, one can determine or estimate the number of multiplications, adaptation steps, learning epochs or similar operations that are applied during the training (and testing) phase. Alternatively, one can simply measure the operational time. Equivalently, one can think of different approaches to measure a classification *task complexity*, which is defined by the combination of the given data set and its labels. For instance, the labelled data set's meta information can be used as an initial estimation of the corresponding task complexity, e.g. the amount of data points, the feature space dimension, the type of the data (categorical, binary, numerical, time series, mixed, etc.), as well as the number of classes and their distribution.

In [16], the authors introduced three data complexity measures, which they identified as infeasible in practice. However, they showed that the complexity can be approximated by classification models. To this end, they used support vector machines (SVMs) [26] for their data complexity analysis. More precisely, they focused on the number of support vectors obtained during the training, with a higher amount of support vectors implying a higher complexity.

In this work, we focus on the complexity of a given feature space. We aim at showing that a classification task can be identified as *chaotic* (and hence as complex) based on Hao's working definition for chaotic systems [10]. Similar to the classification model-based approach in [16], to this end, we will use decision tree models to propose a chaos check method based on Hao's definition. Note that in contrast to [16], we use the term *task complexity* instead of *data complexity* to emphasise that a classification task [19] is defined by the combination of data samples and the corresponding labels. Note that improving pain assessment accuracy performance is not part of our current contribution.

The remainder of this study is organised as follows. In Section II, we motivate our work, present the goal of the study, provide Hao's working definition for chaotic systems and justify the choice of decision tree models. Subsequently, in Section III, we briefly describe the BioVid Heat Pain Database, which constitutes the main example of our numerical chaos check. The formalisation is presented and discussed in Section IV. Section V consists of the experimental evaluation, including a brief description of the Multiple Features data set [25] – which constitutes a low-complex classification task and which is used as the counterpart in our proposed chaos (complexity) check approach –, the experimental settings, as well as the illustration and discussion of the results. Finally, the paper is concluded in Section VI.

II. MOTIVATION: CHAOS AND COMPLEXITY

In this section, we will first discuss the versatile usability of decision tree models. Subsequently, we will provide a summary of Hao's working definition for chaotic systems and check its applicability to decision tree classifiers.

Note that our motivation is based on the following intuitive idea. Identifying a classification task as chaotic based on the decision tree model (i.e. system), implies that the corresponding task is (highly) complex.

A. DECISION TREES–MORE THAN CLASSIFICATION AND REGRESSION TOOLS

Classification and regression trees [7] are classic machine learning models. In this work, we focus on classification trees, which we will simply denote as decision trees. In general, in their main function, decision trees serve as base classifiers in classification ensembles [14], such as in the methods bagging [5], boosting [20], and random forests [6]. However, one can also count the number of decision nodes constructed during the training process to obtain an initial estimation of the corresponding task (labelled data) complexity. In addition, decision trees can be used to get feedback on the importance of individual features.

Note that decision trees are *instable* classification models [5]. This means that small changes of the training data can lead to large changes in the final model. Although *small* and *large* are relative terms, we will focus on the decision trees' instability. We will use this characteristic for the identification of some chaos-specific properties and hence classification task complexity. In the following section, i.e. in Section II-B, we will discuss the importance of stability and instability in chaotic systems.

B. HAO's WORKING DEFINITION FOR CHAOS

Hao provided a working definition for chaotic systems [10], [17], which can be summarised as in Definition 1.

Definition 1 (Working Definition for Chaos): A system is called chaotic, if it fulfils the following four properties.

1. The system's dynamics are deterministic.
2. No external noise is added to the system.
3. The apparently erratic behaviour of individual trajectories is sensitively influenced by infinitesimally small changes in the initial conditions.
4. In contrast to individual trajectories, there are global characteristics or quantities that are not sensitively influenced by the initial conditions.

In this work, we will define decision trees as our system. To this end, let us check the first two properties of Definition 1. Firstly, decision trees are deterministic models, in general. More precisely, repeating the training process with the same training set and parameters always leads to identical decision tree models. The same holds for the repetitive label output (excluding ties), based on some test set. Therefore, Property 1 (deterministic system) of Definition 1 is true for decision tree models. Secondly, once the training and test data are fixed, there is no external noise during the training or classification phases of a decision tree. Therefore, Property 2 (closed system) of Definition 1 is also true for decision tree models.

Note that we will check Properties 3 (instability condition) and 4 (stability condition) of Definition 1 experimentally, in Section V. To this end, we will define *infinitesimally small changes* in the initial conditions as the removal of one single data point from the training set. Moreover, as the individual characteristics (trajectories) we will focus on the analysis of the *number of nodes* constructed during the training process, the *test set accuracy* and the test set-specific *label outputs*. Since there is no universal task complexity measure [16], we will use Definition 1 in combination with these characteristics as an indicator for the complexity of classification tasks. For additional task (data) complexity definitions, we refer the reader to [2] and [16].

III. BioVid HEAT PAIN DATABASE PART A

In this work, we focus on Part A of the BioVid Heat Pain Database (BVDB) [28]. In total, 87 subjects (43 female, 44 male) participated in controlled heat pain elicitation experiments. A heat thermode, which was attached at the participant's forearm, was used to induce pain. The experiments consisted of individual calibration phases and the main procedure.

The calibration phase was introduced to define the ground truth labels. To this end, starting at $32°C$, the temperature was slowly increased, until the participant felt a change from warmth to low pain. The corresponding temperature was defined as the *pain threshold level* and is denoted by T_1. Subsequently, the temperature was further increased until the participant classified the pain as unbearable, which was defined as the *pain tolerance level* and is denoted by T_4. Note that it was not allowed to exceed $50.5°C$. Two intermediate pain levels, denoted by T_2 and T_3 were defined between T_1 and T_4 in equidistant manner. The *no pain level* was defined as $32°C$ and is denoted by T_0.

After defining the ground truth, each participant was stimulated 20 times with each of the pain levels in randomised order. To this end, the temperature was linearly increased to the corresponding value and held for four seconds. After decreasing the temperature back to T_0, i.e. $32°C$, the no pain level was held for a random duration of eight to twelve seconds.

During the main phase, the experimenters recorded videos from three different angles as well as three physiological signals. In this work, we focus on the recorded physiological signals, i.e. electrocardiogram (ECG), electrodermal activity (EDA) and electromyogram (EMG). ECG measures a person's heart activity, whereas EDA and EMG measure a person's skin conductance and muscle activity, respectively. The EMG sensors were attached in the shoulder area with focus on the trapezius muscle. The EDA sensors were attached to the ring finger and index finger, on one of the participant's hands.

To keep this study consistent with our previous works, we will use the exactly same hand-crafted features as in [11] and [12]. The features were extracted from windows of 5.5 seconds length from the temporal and frequency domains, including statistical descriptors, such as mean and extreme values, and signal-specific descriptors, such as the heart rate variability (defined by the ECG signal), amongst others. In total, 194 features were extracted, including 56, 68 and 70 features for the signals EMG, ECG and EDA, respectively. Each person-specific feature set was normalised, leading to zero mean and a standard deviation of value one. To focus on our current contribution, we refer the reader to [11] and [12] for a complete description of the preprocessing and feature extraction steps.

Moreover, we refer the readers interested in facial videos-specific pain intensity recognition based on the BVDB to [22] and [31].

IV. FORMALIZATION

By $X \subset \mathbb{R}^d$, $d \in \mathbb{N}$, we denote a d-dimensional, labelled data set. More precisely, the elements of X consist of pairs of data points and corresponding labels, i.e. $X = \{(x_1, y_1), \ldots, (x_N, y_N)\}$, with $N = |X|$, whereby y_i denotes the label of data point x_i, for $i = 1, \ldots, N$.

Our analysis is based on decision tree (DT) classifiers. By Θ, we denote the set of DT-specific training parameters and settings, for instance, including the *split criterion* or the *cost of misclassification*. Moreover, by DT_X^{Θ}, we denote the decision tree that is designed in combination with training set X and parameter set Θ. Note that in most cases, we will omit the superscript for the sake of readability, simply using the term DT_X. For any data point $z \in \mathbb{R}^d$, we denote the label output of model DT_X specific to z simply by $\mathrm{DT}_X(z)$.

By Q, we denote the set of model-specific measures, such as the *number of decision tree nodes*.

Let $X^1, X^2 \subset \mathbb{R}^d$ be two training sets. In the current study, we focus on measuring the differences between the resulting DT classifiers. To this end, we evaluate the *relative difference*,

Δ, between the corresponding classification models DT_{X^1} and DT_{X^2}, which we define as follows,

$$\Delta(\mathrm{DT}_{X^1}, \mathrm{DT}_{X^2}; q) := \left| \frac{q(\mathrm{DT}_{X^1}) - q(\mathrm{DT}_{X^2})}{q(\mathrm{DT}_{X^1})} \right|, \qquad (1)$$

whereby $q \in Q$ is a DT-specific measure as discussed above. Note that Δ is undefined if the corresponding denominator is equal to zero. However, this case never occurred in our experiments, which are presented in Section V. Moreover, note that Δ, as defined in Eq. (1), is not symmetric, i.e. in general, it holds $\Delta(\mathrm{DT}_{X^1}, \mathrm{DT}_{X^2}; q) \neq \Delta(\mathrm{DT}_{X^2}, \mathrm{DT}_{X^1}; q)$.

In addition, let $Z \neq \emptyset$ be a set of d-dimensional data points, i.e. $Z \in \mathbb{R}^d$. To measure the relative difference of label outputs between models DT_{X^1} and DT_{X^2} specific to the set Z, we define Δ_Z as follows,

$$\Delta_Z(\mathrm{DT}_{X^1}, \mathrm{DT}_{X^2}) := \frac{\left|\{z \in Z : \mathrm{DT}_{X^1}(z) \neq \mathrm{DT}_{X^2}(z)\}\right|}{|Z|}, \qquad (2)$$

whereby in contrast to Eq. (1), in Eq. (2), $|\cdot|$ denotes the number of elements of the corresponding sets, instead of the absolute value.

V. EXPERIMENTS

In this section, we will first briefly describe the Multiple Features data set, which will be used as an intuitive non-complex (and non-chaotic) counterpart to the BioVid Heat Pain Database. Subsequently, we will provide the evaluation protocol and finally present and discuss the outcomes. Note that, as already discussed, the focus of the experiments is not set on improving the pain assessment accuracy performance.

A. MULTIPLE FEATURES DATA SET

The Multiple Features (MFeat) data set [25] is publicly available at the UCI Machine Learning Repository.[1] The MFeat data set consists of 2,000 handwritten numerals, i.e. $0, \ldots, 9$, thus constituting a 10-class classification task. The provided feature dimension of the data is equal to 649. The features are organised in the following six feature sets: Fourier coefficients of the character shapes (76 features), profile correlations (216 features), Karhunen-Loeve coefficients (64 features), pixel averages in 2×3 windows, Zernike moments (47 features), and morphological features (6 features). The data set is balanced, including 200 samples per class.

B. EXPERIMENTAL SETTINGS AND SOFTWARE

As discussed in Sections II and IV, we are using decision tree (DT) classifiers for our numerical analysis. More precisely, we use the Gini Index as the impurity measure in combination with the standard cost function, i.e. all types of classification errors are treated equally with cost value one. Moreover, we leave each decision tree unpruned, to measure the exact differences between the constructed DT models.

```
Input:
  • Training set:      $X \subset \mathbb{R}^d$, with $N_{\text{train}} = |X|$
  • Test set:          $Z \subset \mathbb{R}^d$
  • DT parameters:  $\Theta$
    e.g. $\Theta = \{\text{split criterion, cost of misclassification}\}$
  • Set of measures: $Q$
    e.g. $Q = \{\text{test set accuracy, number of tree nodes}\}$
Initialisation:
  • Set $\delta^q = \mathbf{0}^{N_{\text{train}}}$, for each $q \in Q$
  • Set $\delta^{\text{out}} = \mathbf{0}^{N_{\text{train}}}$
  • Design a DT in combination with $X$ and $\Theta$,
    i.e. $\mathrm{DT}_X^\Theta$ (← this model defines the ground truth)
FOR $i = 1, \ldots, N_{\text{train}}$ and each element $q \in Q$:
  • $X^i = X \setminus \{x_i\}$
  • Design a DT in combination with $X^i$ and $\Theta$,
    i.e. $\mathrm{DT}_{X^i}^\Theta$
  • $\delta_i^q = \Delta(\mathrm{DT}_X^\Theta, \mathrm{DT}_{X^i}^\Theta; q)$, as defined in Eq. (1)
  • $\delta_i^{\text{out}} = \Delta_Z(\mathrm{DT}_X^\Theta, \mathrm{DT}_{X^i}^\Theta)$, as defined in Eq. (2)
```

FIGURE 1. Evaluation protocol per test fold (epoch). For the given training set, *X*, a reference decision tree model is trained in combination with parameter set Θ (DT_X^Θ). In each iteration, *i*, one single data point is removed from the initial training set *X*, leading to decision tree $\mathrm{DT}_{X^i}^\Theta$. Model $\mathrm{DT}_{X^i}^\Theta$ is used to compute the relative difference to the ground truth defined by DT_X^Θ.

For each test set, we will focus on the percentage difference in the number of nodes, accuracy, as well as the output diversity. The first two measures are computed by using Eq. (1), whereas the output diversity is calculated by applying Eq. (2). Note that will analyse whether Properties 3 (instability condition) and 4 (stability condition) of Definition 1 are fulfilled. More precisely, we will check whether all of the three measures are sensitively or not sensitively influenced by small changes in the training data.

For the BVDB, we will apply a nested 87-fold cross validation as follows. Note that the BVDB consists of 87 participants, with 100 data points each, i.e. 20 per class (5 classes). For each test fold (i.e. test subject), we will apply 8,601 iterations. In each iteration, we will remove one data point from the initial training set, which consists of 8,600 data points. Thus, the change of the initial conditions is equal to $1/8600 \approx 0.012\%$, for the BVDB.

For the MFeat data set, we will apply a nested 20-fold cross validation. Note that the MFeat data set consists of 2,000 data points in total, with 200 points per class (10 classes). For each test fold, we will apply 1,901 iterations. In each iteration, we will remove one data point from the initial training set, which consists of 1,900 data points. Thus, the change of the initial conditions is equal to $1/1900 \approx 0.053\%$, for the MFeat data set.

Note that for both data sets, MFeat and the BVDB, each test fold consists of 100 data points, equally distributed among the classes, i.e. 20 per class for the BVDB and 10 per class for

[1] MFeat data set: https://archive.ics.uci.edu/ml/datasets/Multiple+Features

TABLE 1. Cross validation evaluation parameters. The term +1 indicates that one model is used to train on all available training data to provide the reference values. The MFeat and BVDB data sets consist of 10 and 5 equally distributed classes, respectively. Each test fold is equally distributed as well.

Data set	MFeat	BVDB
Total number of data points	2,000	8,700
Number of epochs	20	87
Test data size per epoch	100	100
Training iterations per epoch	1,900+1	8,600+1
Change in training size (initial condition)	0.053%	0.012%

TABLE 2. Percentage change averaged over all test epochs. For each test epoch, one single data point is removed from the training set per iteration. ΔNds: Difference in number of decision tree nodes. ΔAcc: Difference in accuracy. ΔOut: Difference in label outputs. To compute ΔNds and ΔAcc, Eq. (1) is used, whereas Eq. (2) is used for ΔOut. The number of test folds (epochs) is denoted in brackets. The change of the initial conditions is equal to 0.012% and 0.053% for the BVDB and the MFeat data set, respectively.

	Averaged Mean		Averaged Max	
	MFeat (20)	BVDB (87)	MFeat (20)	BVDB (87)
ΔNds	0.19 ± 0.15	0.25 ± 0.11	9.48 ± 2.40	1.66 ± 0.45
ΔAcc	0.14 ± 0.27	2.50 ± 1.55	2.89 ± 1.32	29.0 ± 14.8
ΔOut	0.27 ± 0.47	3.67 ± 1.57	4.90 ± 1.97	41.1 ± 17.1

the MFeat data set. The test fold-specific evaluation protocol is summarised in Figure 1. The evaluation protocol leads to 87×8601 and 20×1901 nested cross validations for the BVDB and the MFeat data sets, respectively. The first iteration is used to train the reference model in combination with the whole training set, without removing any data points. The data sets-specific nested cross validation parameters are summarised in Table 1.

We used Matlab[2] (version R2019b) for the experiments with the build-in function *fitctree* for the construction of the decision trees.

C. RESULTS

Table 2 depicts the averaged percentage changes for the number of decision tree nodes (ΔNds), the accuracy (ΔAcc) and label output difference (ΔOut), including the standard deviation values. From Table 2, we can make the following observations. Firstly, the highest averaged mean values are observed for the difference in outputs, for both data sets, MFeat and the BVDB. Secondly, for the MFeat data set, all of the relative differences (ΔNds, ΔAcc, ΔOut) are smaller than 1% – even smaller than 0.3% – on average. Thirdly, for the BVDB, only the relative difference for the number of nodes is less than 1% and also even less than 0.3%. For ΔAcc and ΔOut, the averaged relative difference is equal to 2.5% and 3.67%, respectively. Note that the change of the training data is equal to 0.012% for the BVDB. Fourthly, the averaged maximum changes are the highest for the number of nodes (9.48%), in combination with the MFeat data set. Based on the BVDB, the highest averaged maximum change is noted for ΔOut, with 41.1%.

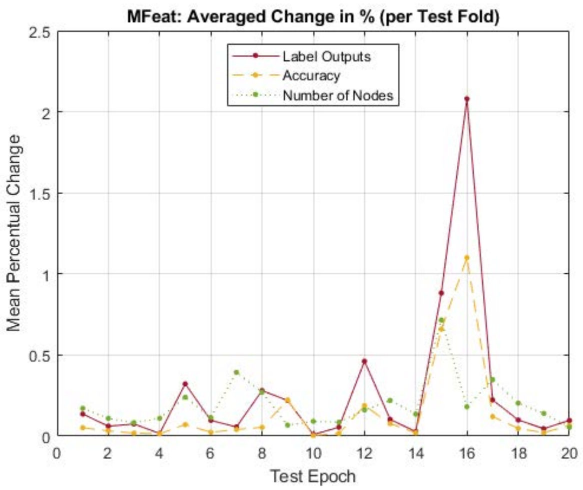


FIGURE 2. Averaged percentage change per test epoch. Each dot represents the mean value of 1,900 iterations. In each iteration, one data point is removed from the training set (0.053% training data change).

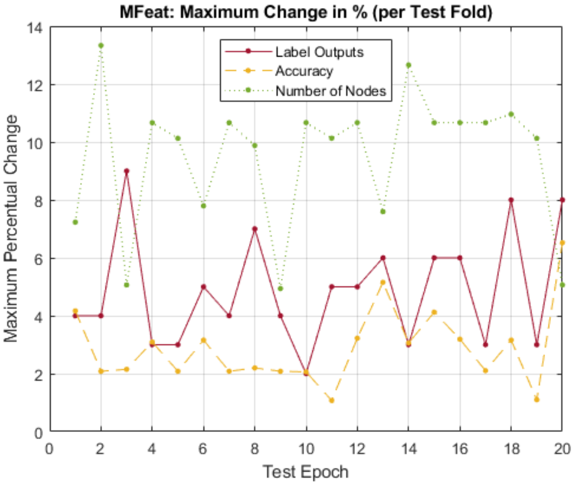


FIGURE 3. Maximum percentage change per test epoch. Each dot represents the maximum value within 1,900 iterations. In each iteration, one data point is removed from the training set (0.053% training data change).

Note that the averaged mean values from Table 2 represent the averaged values over $20 \times 1,900 = 38,000$ and $87 \times 8,600 = 748,200$ iterations for the MFeat data set and the BVDB, respectively. Figures 2–5 depict the epoch-specific values, which led to the results presented in Table 2.

From Figure 2 (MFeat data set), we can observe that the averages of ΔNds, ΔAcc and ΔOut exceed the value of 0.5% only once, i.e. in epoch 15. Moreover, the value of 1% is exceeded only in epoch 16 and solely for ΔAcc and ΔOut, whereby the averaged ΔOut also exceeds the value of 2% (in epoch 16).

Figure 3 depicts the maximum percentage changes for the MFeat data set per epoch, i.e. the maximum of 1,900 training iterations, with a change of 0.053% of the initial training data. From Figure 3, we can observe that the maximum relative

[2] Matlab website: https://www.mathworks.com/

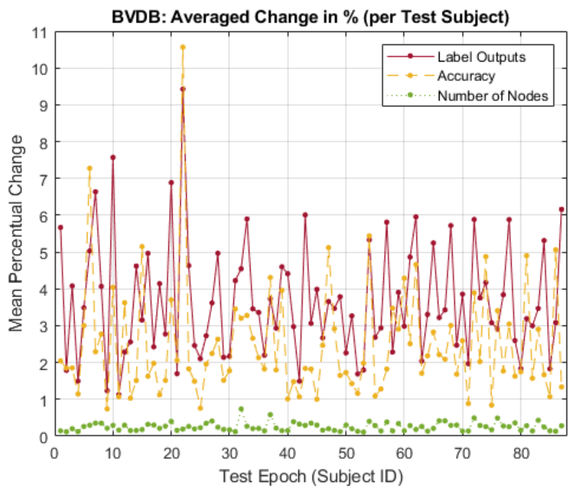


FIGURE 4. Averaged percentage change per test epoch (subject). Each dot represents the mean value of 8,600 iterations. In each iteration, one data point is removed from the training set (0.012% training data change).

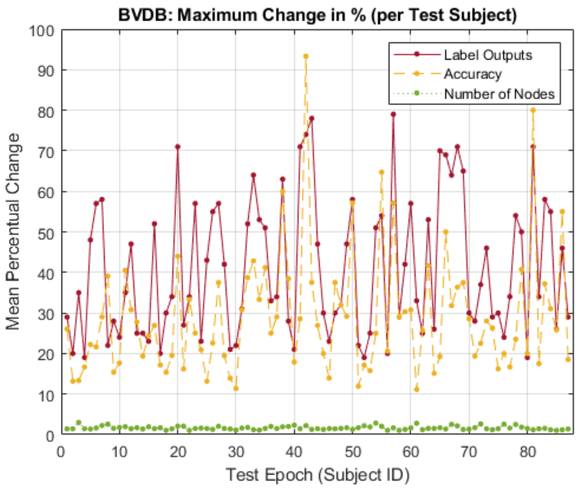


FIGURE 5. Maximum percentage change per test epoch (subject). Each dot represents the maximum value within 8,600 iterations. In each iteration, one data point is removed from the training set (0.012% training data change).

changes are observed for the number of decision tree nodes, exceeding 13% in epoch 2. The maximum change of accuracy is observed in epoch 20, exceeding 6%, whereas the maximum value for ΔOut is observed for epoch 3, leading to 9%. Note that the size of the test sets is always equal to 100 data points. Therefore, a change of 9% implies that by removing one data point from the 1,900 training points leads to a model that provides 9 different label outputs for the given test set, in comparison to the ground truth model that is trained in combination with the whole training data.

From Figure 4 (BVDB), we can make the following observations. The averaged relative change in the number of nodes never exceeds 1%. The corresponding difference in accuracy falls below 1% in only four epochs, i.e. epochs 9, 25, 71 and 75. The averaged ΔOut values exceed always 1%. In epoch 22, the averaged ΔOut and ΔAcc values even exceed 9% and 10%, respectively.

Figure 5 depicts the maximum percentage changes for the BVDB per epoch, i.e. the maximum of 8,600 training iterations, with a change of 0.012% of the initial training data. From Figure 5, we can observe that the relative change in the number of decision tree nodes never exceeds 10%. The maximum value is observed for epoch 3 and is approximately equal to 3% (3.0004%). The maximum ΔAcc and ΔOut values always exceed 10%. The maximum change in accuracy is observed in epoch 42, exceeding 90%. The maximum ΔOut value is noted in epoch 57 and is equal to 79%. That means that by removing one single data point from the 8,600 training data points led, at least once, to 79 differences in the label outputs on the corresponding 100 data points-specific test set, in comparison to the decision tree model that was trained in combination with the whole data set.

D. DISCUSSION

The reason why we included the difference in the label outputs (ΔOut) is that it is more precise than the difference in accuracy (ΔAcc). Note that two classifiers can have an accuracy of 50%, respectively, while disagreeing on all of their label outputs. Therefore, in the following, we set the focus on the measures ΔNds and ΔOut. While in a chaotic system, i.e. complex classification task, we expect the difference in the number of nodes, ΔNds, to be low on average, we assume the difference in the label outputs, ΔOut, to be relatively high in comparison. Moreover, in non-complex tasks, we expect both measures to be low on average and thus violating Property 3 (instability condition) of Definition 1.

Since our current work presents the initial outcomes in combination with our proposed complexity check, we do not have any empirical data to compare with. We observed that both measures, ΔNds and ΔOut, stayed below 0.3% on average, based on a $20 \times 1,900$ cross validation evaluation, for the MFeat data set. On the other hand, while ΔNds stayed below 0.3%, the averaged ΔOut values exceeded 3.5% based on a $87 \times 8,600$ cross validation evaluation in combination with the BVBD, with a change of 0.012% in the initial conditions.

If we focus on the relation between the mean ΔOut and ΔNds values, we obtain the following outcomes. For the MFeat data set, it holds $\Delta\text{Out} : \Delta\text{Nds} \approx 1.42$. In contrast, for the BVDB, it holds $\Delta\text{Out} : \Delta\text{Nds} \approx 14.68$. While it could be difficult to define task-independent absolute thresholds for ΔNds and ΔOut, the relation $\Delta\text{Nds} : \Delta\text{Out}$ might allow for a complexity comparison across different classification tasks.

VI. CONCLUSION

From the current work, we can draw the following conclusions.

Firstly, we made a general observation based on the experimental outcomes with regard to the instability of decision tree (DT) classifiers. It is well known that DTs are instable models. However, in this work, we explicitly measured their instability in combination with the BioVid Heat Pain Database (BVDB). More precisely, note that our reference DT classifiers were always trained on 8,600 data points, whereas the corresponding comparison models were always trained on 8,599 data points of the same training data. We observed a maximum disagreement on 79 label outputs between the reference model and the comparison model based on a test set consisting of 100 data points.

Secondly, based on Hao's working definition for chaotic systems, we introduced a simple method for a chaos check. Note that our evaluation is based on the simple idea that a chaotic classification task is of high complexity. Our chaos check is based on the observation of relative changes in the number of nodes (ΔNds), accuracy (ΔAcc) and label outputs (ΔOut), in combination with small changes of the initial conditions as proposed by Hao. To this end, we focused on the smallest possible changes of the training data, by iteratively removing one single data point from the initial training set. Based on the resulting averaged values of ΔNds, ΔAcc and ΔOut, we can decide whether Properties 3 and 4 of Definition 1 are fulfilled.

Note that we would like to emphasise that our proposed chaos and hence complexity check is a first attempt to combine supervised learning and chaos theory, based on DT models. There are different possibilities to adjust the introduced method, e.g. by including different model-specific measures or changing the evaluation protocol. Similar to our recently proposed detection of ordinal class structures [3], [4], the chaos/complexity check depends on the provided feature space. Therefore, a different application of our proposed approach could be the detection of features with chaotic behaviour, leading to a possible feature selection approach.

As discussed in Section V-D, it might be sufficient to focus on the relation between the measures ΔNds and ΔOut. This might be used to find complex subtasks, which then could be addressed more specifically, e.g. by dividing the initial task into binary or trinary subtasks. Additionally, one might include the number of classes, features or samples into the relation calculation between ΔNds and ΔOut, or even in Equations (1) and (2). For instance, intuitively speaking, in a classification task with ten classes, two models are more likely to disagree, with respect to the label outputs, than in a task with only five classes. Finally, one could try to adapt the proposed chaos/complexity check to time series data. This might lead to a feature space-independent task complexity analysis based on (filtered) raw data.

ACKNOWLEDGMENT

The authors gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research.

REFERENCES

[1] M. Amirian and F. Schwenker, "Radial basis function networks for convolutional neural networks to learn similarity distance metric and improve interpretability," *IEEE Access*, vol. 8, pp. 123087–123097, 2020.

[2] M. Basu and T. K. Ho, *Data Complexity in Pattern Recognition*. London, U.K.: Springer, 2006.

[3] P. Bellmann, L. Lausser, H. A. Kestler, and F. Schwenker, "A theoretical approach to ordinal classification: Feature space-based definition and classifier-independent detection of ordinal class structures," *Appl. Sci.*, vol. 12, no. 4, p. 1815, Feb. 2022.

[4] P. Bellmann and F. Schwenker, "Ordinal classification: Working definition and detection of ordinal structures," *IEEE Access*, vol. 8, pp. 164380–164391, 2020.

[5] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.

[6] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[7] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Evanston, IL, USA: Routledge, 2017.

[8] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Ann. Eugenics*, vol. 7, no. 2, pp. 179–188, Sep. 1936.

[9] S. Gruss, M. Geiger, P. Werner, O. Wilhelm, H. C. Traue, A. Al-Hamadi, and S. Walter, "Multi-modal signals for analyzing pain responses to thermal and electrical stimuli," *J. Vis. Exp.*, no. 146, Apr. 2019, Art. no. e59057.

[10] B.-L. Hao, *Chaos II*. Singapore: World Scientific, 1990.

[11] M. Kächele, M. Amirian, P. Thiam, P. Werner, S. Walter, G. Palm, and F. Schwenker, "Adaptive confidence learning for the personalization of pain intensity estimation systems," *Evolving Syst.*, vol. 8, no. 1, pp. 71–83, 2017.

[12] M. Kächele, P. Thiam, M. Amirian, F. Schwenker, and G. Palm, "Methods for person-centered continuous pain intensity assessment from biophysiological channels," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 5, pp. 854–864, Aug. 2016.

[13] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Dept. Comput. Sci., Univ. Toronto, Tech. Rep., 2009.

[14] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. Hoboken, NJ, USA: Wiley, 2014.

[15] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[16] L. Li and Y. S. Abu-Mostafa, "Data complexity in machine learning," California Inst. Technol., Tech. Rep. CaltechCSTR:2006.004, 2006.

[17] O. Loistl and I. Betz, *Chaostheorie: Zur Theorie Nichtlinearer Dynamischer Systeme*. R. Oldenbourg Verlag: Munich, Germany, 1996.

[18] R. M. Al-Eidan, H. Al-Khalifa, and A. Al-Salman, "Deep-learning-based models for pain recognition: A systematic review," *Appl. Sci.*, vol. 10, no. 17, p. 5984, Aug. 2020.

[19] T. M. Mitchell, *Machine Learning* (McGraw-Hill Series in Computer Science). New York, NY, USA: McGraw-Hill, 1997.

[20] R. E. Schapire, "The strength of weak learnability," *Mach. Learn.*, vol. 5, no. 2, pp. 197–227, 1990.

[21] S. D. Subramaniam and B. Dass, "Automated nociceptive pain assessment using physiological signals and a hybrid deep learning network," *IEEE Sensors J.*, vol. 21, no. 3, pp. 3335–3343, Feb. 2021.

[22] M. Tavakolian, M. B. López, and L. Liu, "Self-supervised pain intensity estimation from facial videos via statistical spatiotemporal distillation," *Pattern Recognit. Lett.*, vol. 140, pp. 26–33, Dec. 2020.

[23] P. Thiam, H. Hihn, D. A. Braun, H. A. Kestler, and F. Schwenker, "Multi-modal pain intensity assessment based on physiological signals: A deep learning perspective," *Frontiers Physiol.*, vol. 12, Sep. 2021, Art. no. 720464.

[24] P. Thiam, V. Kessler, M. Amirian, P. Bellmann, G. Layher, Y. Zhang, M. Velana, S. Gruss, S. Walter, H. C. Traue, D. Schork, J. Kim, E. André, H. Neumann, and F. Schwenker, "Multi-modal pain intensity recognition based on the SenseEmotion database," *IEEE Trans. Affect. Comput.*, vol. 12, no. 3, pp. 743–760, Jul. 2021.

[25] M. van Breukelen, R. P. W. Duin, D. M. J. Tax, and J. D. Hartog, "Handwritten digit recognition by combined classifiers," *Kybernetika*, vol. 34, no. 4, pp. 381–386, 1998.

[26] V. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer, 2013.

[27] M. Velana, S. Gruss, G. Layher, P. Thiam, Y. Zhang, D. Schork, V. Kessler, S. Meudt, H. Neumann, J. Kim, F. Schwenker, E. André, H. C. Traue, and S. Walter, "The SenseEmotion database: A multimodal database for the development and systematic validation of an automatic pain- and emotion-recognition system," in *Proc. MPRSS*, in Lecture Notes in Computer Science, vol. 10183. Cham, Switzerland: Springer, 2016.

[28] S. Walter, S. Gruss, H. Ehleiter, J. Tan, H. C. Traue, S. Crawcour, P. Werner, A. Al-Hamadi, and A. O. Andrade, "The biovid heat pain database data for the advancement and systematic validation of an automated pain recognition system," in *Proc. CYBCONF*, Jun. 2013, pp. 128–131.

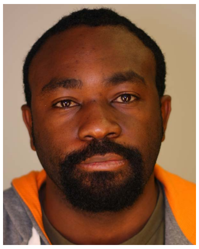
[29] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona, "Caltech-UCSD Birds 200," California Inst. Technol., Tech. Rep. CNS-TR-2010-001, 2010.

[30] P. Werner, A. Al-Hamadi, S. Gruss, and S. Walter, "Twofold-multimodal pain recognition with the X-ITE pain database," in *Proc. ACII Workshops*, Sep. 2019, pp. 290–296.

[31] P. Werner, A. Al-Hamadi, K. Limbrecht-Ecklundt, S. Walter, S. Gruss, and H. C. Traue, "Automatic pain assessment with facial activity descriptors," *IEEE Trans. Affect. Comput.*, vol. 8, no. 3, pp. 286–299, Jul./Sep. 2017.

[32] P. Werner, D. Lopez-Martinez, S. Walter, A. Al-Hamadi, S. Gruss, and R. W. Picard, "Automatic recognition methods supporting pain assessment: A survey," *IEEE Trans. Affect. Comput.*, vol. 13, no. 1, pp. 530–552, Jan. 2022.

PETER BELLMANN received the degree in mathematics from Ulm University, Ulm, Germany, in 2016, where he is currently pursuing the Ph.D. degree in computer science with the Neural Information Processing Department. His research interests include ordinal classification, multiple classifier systems, multi-modal fusion architectures, and machine learning techniques for the recognition of affective states in human-centered signals.



PATRICK THIAM received the Licence degree in mathematics from the University of Yaoundé I, Yaoundé, Cameroon, in 2007, and the M.Sc. and Ph.D. degrees in computer science from Ulm University, Ulm, Germany, in 2014 and 2021, respectively. His research interests include among others multi-modal supervised and semi-supervised learning, active learning, and deep learning approaches for the recognition of affective states in human–computer interaction.

HANS A. KESTLER (Senior Member, IEEE) received the degree in electrical engineering from the Technical University of Munich, and the Ph.D. and Habilitation degrees in computer science from Ulm University, in 2002 and 2011, respectively. From 2014 to 2015, he was a Professor of systems biology of aging at the Fritz Lipmann Institute, Jena. Since 2016, he has been the Head of the Institute of Medical Systems Biology and the Core Unit Bioinformatics within the Faculties of Computer Science and Medicine, Ulm University. He is currently an Associated Group Leader with the Leibniz Institute on Aging. He has published more than 330 articles in journals, books, and conferences. His research interests include methodological foundations of pattern recognition, bioinformatics, molecular systems biology, and digital health. Since 2013, he has also been the Vice-President of the German Classification Society and the Data Science Society.

FRIEDHELM SCHWENKER (Member, IEEE) received the Diploma and Ph.D. degrees in mathematics and computer science from the University of Osnabrück. He is currently a Professor with the Institute of Neural Information Processing, Ulm University. He was a Co-Editer of 20 special issues and workshop proceedings published in international journals and publishing companies. He has published more than 200 papers at international conferences and journals. His research interests include artificial neural networks, machine learning, statistical learning theory, data mining, pattern recognition, information fusion, and affective computing. He has served as the Co-Chair of the IAPR TC3 on Neural Networks and Computational Intelligence. He has also been the Chair of the IAPR TC9 on Pattern Recognition in human–computer interaction.