Intelligent Environments 2018
I. Chatzigiannakis et al. (Eds.)

doi:10.3233/978-1-61499-874-7-417

What Am I Writing: Classification of On-Line Handwritten Sequences

Junaid YOUNAS [a,b,1], Stefan FRITSCH [b], Gerald PIRKL [a], Sheraz AHMED [a], Muhammad Imran MALIK [c], Faisal SHAFAIT [c] and Paul LUKOWICZ [a,b]

[a] *German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany; firstname.lastname@dfki.de*

[b] *Kaiserslautern University of Technology, Germany; lastname@rhrk.uni-kl.de*

[c] *National University of Sciences and Technology (NUST), Islamabad, Pakistan; firstname.lastname@seecs.edu.pk*

Abstract. This paper presents a novel approach for classification of online handwritten sequences into text, equations, and plots. This classification helps in identifying the progress of student/learner while attempting different problems in context of classroom equipped with tablets, iPads. Furthermore, it serves as a feedback (for both students and instructors) to analyze the writing behavior and understanding capabilities of the student. The presented approach is based on an ensemble of different machine learning classifiers, where not only the individual sequences are classified but also the contextual information is used to refine the classification results. To train and test the system, a real-world dataset consisting up of 11,601 sequences was collected from 20 participants. Evaluation results on the real dataset shows that the presented system, when tested in person independent settings, is capable of classifying handwritten on-line sequences with an overall accuracy of 92%.

Keywords. sequence classification, feature engineering, machine learning, base classifier, ensemble classifier

1. Introduction

With the evolution of digital world, new opportunities have emerged to interact with our environment and vice-versa. Displays and sensor technologies are the main driving forces behind it. Touch-screen displays have caught the attention of research community for on-line handwriting recognition. Handwriting can be broadly divided into two major categories: (i) Off-line and (ii) On-line. The major difference between the two is the way they are produced and analyzed. Writing produced on normal paper with normal pen is considered as off-line, whereas writing on the touch-screen devices either with finger or digital pen is termed as on-line handwriting. On-line handwriting also includes writing with sensor pens on normal or special paper.

History of handwriting classification goes back to early 20^{th} century, when German police used handwriting as biometric feature before the second world war [1]. After that,

[1] Corresponding Author: Junaid Younas, DFKI, Kaiserslautern, Germany; firstname.lastname@dfki.de.

Nottingham police came up with a handwriting classification system as a result of a decade worth of research. Their presented system was used to classify individuals based on their handwriting, to maintain the record of Nottingham residents.

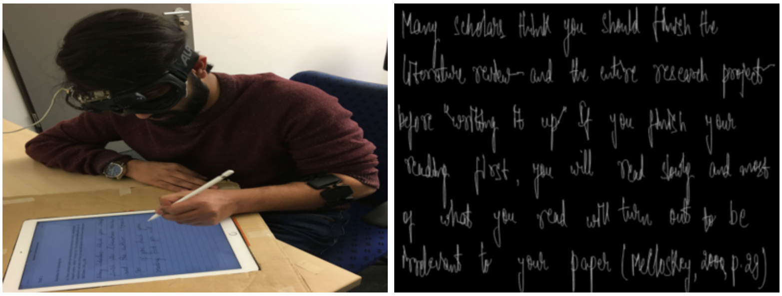
In 1954, Smith et al. [2] presented six factors for classification of large volumes of handwritings. These factors were speed, size, slant, spacing, pressure and form. Out of these factors, pressure and form were marked as unconscious behaviors and others as developed factors.

Writer and writing style classification approach for on-line handwriting recognition was presented by Bouletreau et al. [3]. Based on velocity, pen-tip displacement data was segmented into strokes. Every stroke was represented by a 1-d feature vector which was then used to train Kohonen network for writer classification. Writing styles were classified using clustering and discriminant analysis techniques. Schomaker et al. [4] presented synthetic parameters for classification of handwriting into different writing families. They termed their approach as preliminary step for handwriting classification.

Most of the work done in handwriting classification domain focuses on person classification and recognition [5,6], word recognition [7,8,9], mathematical expression recognition [10,11,12], and/or sentiment analysis [13,14]. Classification of written data into different classes hasn't been addressed so far, up to best of authors knowledge. Therefore, comparison with existing methods, if any, seems not to be fair. Classification of on-line handwritten data into text and non-text classes using global features, local features, Recurrent Neural Networks (RNNs), LSTM and BLSTM networks has been done in recent past [15,16,17].

Ensemble classifiers are a popular choice for data stream classification [18,19]. The idea behind ensemble classifier is to learn a set of base classifiers and make a prediction based on output of the individual classifiers. This results into reduction of variance and bias, as predictions are now dependent on multiple classifiers trained on the same dataset, instead of a single one.

A wearable sensor-based adaptive system was presented by Pirkl et al. [20]. This system was used to monitor the progress of learner during exercises. It evaluates the learning analytics using multiple on-body sensors and sensor pen. They presented the cognitive analysis of novice and expert users by comparing time taken to perform the exercise to the total number of writing segments produced.



(a) Experimental setup (b) Output after pre-processing

Figure 1. A participant writing on iPad and its digital output

This paper presents a novel, generic approach for classification of handwritten sequences into text, calculation or plot/graph. Experimental setup for data collection is shown in Figure 1a. In addition to classification of individual sequences, contextual information is leveraged in order to further refine the classification results. The rest of paper is structured as follows: Section II elaborates system overview and its modules, followed by detailed analysis of the results and evaluation in section III. Section IV concludes the paper with hints towards future prospects of the presented work.

2. System Overview

We developed a data acquisition module for iPad which helped in collection of the dataset. Acquired data was preprocessed, and fed to the feature extraction module to extract the 26 dimensional feature vector. These feature vectors were then used to train individual classifiers along with an ensemble of them. We also utilized the contextual information for the given sequence while generating the final output. The proposed system is demonstrated in Figure 2.

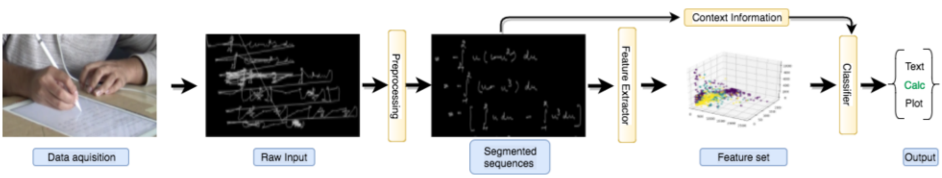


Figure 2. System overview

2.1. Data Acquisition Module

For the purposes of data collection, a dedicated iOS platform was developed. It provided functions for creating, managing and exporting documents. Documents were created on the basis of a template, which predefined the document's structure.

Apple pencil was used for writing on the iPad, which can deliver up to 240 data points per second. These points encapsulates information regarding the location of the pencil on the touchscreen, force of the touch, altitude and azimuth angle, and time[2]. An eraser option was also available in order to be able to make corrections to the written text. Handwriting of the user was rendered in view by linear interpolation between succeeding points. A SQLite database instance was used to store data on the device locally.

2.2. Preprocessing Module

Data acquisition module passed raw continuous data sequences to the preprocessing module. The data also included information regarding the erased sequences. Cleansing of the input data was performed followed by the segmentation into individual sequences. Every segmented sequence represented a single word or expression. The length of the segmented sequences was variable comprising of a single or multiple strokes based on the distance between them. After passing through preprocessing module, the data was visually identical to the one written on the iPad as shown in Figure 1b.

[2]https://developer.apple.com/documentation/uikit/uiview; documentation of the UIView class

2.3. Feature Extraction Module

In the proposed approach, we extracted features regarding the time information, precise and corrected (x,y) co-ordinates of hand-written sequences and the force information. The extracted features also included the commonly used features in on-line handwriting recognition literature [7,5] and signature verification [22,21] .

- Length of the segmented sequence (1): Distance in pixels between start and end point.
- Time of segmented sequence (2): Time in seconds for a single sequence.
- Variance and standard deviation $(3),(4)$ of the rate of change during the segmented sequence Δt.
- Displacement (5): The shortest possible distance in pixels of pencil movement for a given segmented sequence.
- Speed (6): The rate at which the given sequence is produced.
- Velocity (7): Time taken to cover the displacement for a given sequence.
- x, y-range $(8),(9)$: Range is defined by the difference between the maximum and minimum value present in given sequence values.

$$x - range = max[x(t)] - min[x(t)] \tag{1}$$

- x,y-skew $(10),(11)$: Skewness is a measure of the amount and direction of departure from horizontal symmetry for a given sequence.
- x,y-kurtosis $(12),(13)$: Kurtosis is a measure of the height and sharpness of the central peak for a given sequence.
- Variance of Δx, Δy $(14),(15)$: The rate of change of pixels in both horizontal and vertical direction.
- Standard deviation of Δx, Δy $(16),(17)$.
- Variance and standard deviation of direction angles of a segmented sequence (18), (19): Measure of angles between consecutive pixel for a given sequence.
- Variance and standard deviation of gradient of a segmented sequence $(20),(21)$: Gradient or slope of consecutive pixels in a given sequence. Gradient is measure of steepness and direction of line.
- Vicinity aspect (22): The aspect of the trajectory of a given sequence [7]

$$\frac{(y - range) - (x - range)}{(y - range) + (x - range)} \tag{2}$$

- Vicinity curliness (23): The length of given sequence divided by $max(\Delta x, \Delta y)$ [7].
- Range of force (24): The difference between the maximum and minimum force value for a given sequence.
- Mean force (25): Average force applied for a given sequence.
- Variance of force (26).

2.4. Classification Module

Classification module comprised of a combination of the baseline machine learning algorithms, which were fused together as an ensemble classifier. In the presented approach, we utilized K-nearest neighbors, Random Forests and Decision Trees as the base classifiers. Classification module is demonstrated in Figure 3.

2.4.1. K-Nearest Neighbors (KNN)

K-nearest neighbors [23] is one of the simplest machine learning algorithms for pattern recognition. KNN classification decision is made on the basis of majority votes of the nearby data points. The target object is assigned to most common class present in the K-nearest neighboring data points.

2.4.2. Decision Tree (DT)

Decision tree classifiers [24] have been successfully used in a wide range of classification problems because of their flexibility, simplicity and computational efficiency. Decision trees recursively partitions the dataset into smaller subsets and defines a decision framework consisting of a set of tests defined at each branch in the tree.

2.4.3. Random Forest (RF)

Random forest classifier [25] is a combination of the tree predictors, and is considered as a very effective machine learning algorithm for prediction. Each individual tree within the forest optimizes over a randomly selected set of features which significantly reduces correlation between the different trees contributing towards the diversity of the ensemble classifier.

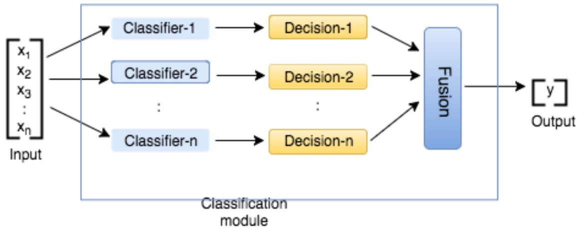


Figure 3. Classification module

2.4.4. Ensemble Classifier

The performance of the base classifier is mainly dependent on the successful learning of patterns present in training data. Learning patterns of every base classifier may vary from each other despite of being trained on the same dataset. To improve the performance, an ensemble of classifiers was used, which fused outputs of the individual base classifiers to generate an improved and stable single output. Ensemble classifiers have been successfully leveraged in the past to improve the classification results.

3. Experiments and Evaluation

3.1. Data Collection

A group of 20 participants (14 males, 6 females) were selected for the experiments. All participants were students from different disciplines belonging to different regions, i.e Pakistan, India, Cuba, Venezuela, USA, etc. 18 participants were right handed and 17 participants (12 males, 5 females) had their first writing experience on digital devices (iPad in our case).

During the experiment, participants were asked to solve different exercises based on the instruction manual provided to them. Exercises comprised of text reproduction, creative writing, copying equations, solutions to basic calculus problems and drawing some easy graphs.

3.2. Dataset and Evaluation Protocol

The dataset consisted of $11,601$ segmented writing sequences. 54% sequences belonged to text, 24% to calculation and rest of the 22% belonged to the plot/graph class. We used 70% of data for training and the rest of the 30% for testing. Small subsets of consecutive sequences belonging to a single class were used to provide context information. Taking the contextual information into account, a single final decision was produced for the given subset.

We used scikit library [26] to train the machine learning models. We also evaluated our system when contextual information with individual sequence was provided. We evaluated K-nearest neighbors, random forest and decision tree classifier. We carried out 10-fold cross-validation during the training. The obtained results are reported using the average accuracy and confusion matrices.

3.3. Results and Discussion

Detailed analysis of the presented approach considering performance of the base classifiers and their ensemble combination with and without information of the neighboring sequences is elaborated below.

We start our discussion with the selection of optimal parameters for individual base classifiers. To extract the optimal parameters for every base classifier, a grid-search was performed along with a 10-fold cross validation on the train set. The best number of nearest neighbors $k = 11$ was estimated, which is the key parameter to train a KNN classifier. Parameters for the random forest classifier are the number of estimators, $est = 99$ and splitting criteria is $entropy$.

3.3.1. Person dependent results

Person dependent classification refers to organization of train and test set in a way that both contains a percentage of every user's data. We trained a KNN classifier with $k = 11$, number of nearest neighbor configuration. After 10-fold cross-validation, we achieved the overall accuracy of 73% as summarized in Table 1. The KNN model was particularly effective for classification of the text class, predicting 92% of the total sequences correctly . Results for calculation class and plot/graph class were 64% and 65% respectively.

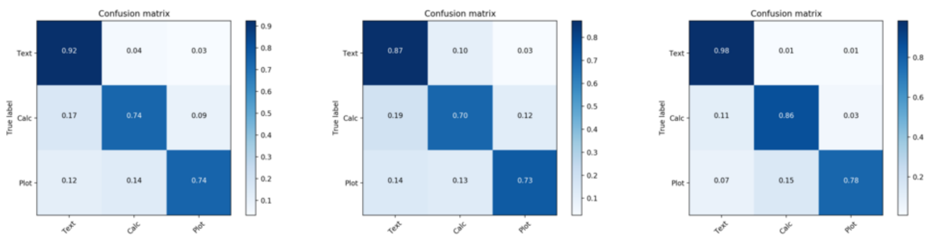
Overall results accomplished by the decision tree classifier were 72.3%. Individual class results were found to be 81%, 67% and 69% for text, calculation and graph/plot class, respectively.

The employed random forest classifier was able to outperform other base classifiers. Individual results achieved by training random forest classifier were found to be 90%, 79%, and 77% for text, calculation and plot/graph class, respectively.

Table 1. Person dependent performance analysis of base classifiers

Classifier	Overall Accuracy(%)	Text (%)	Calculation (%)	Graph (%)
KNN	73	92	64	65
Random Forest	80	90	79	77
Decision Tree	72.3	81	67	69
Ensemble Classifier	83.2	92	74	74

We tried different combinations for ensemble classifier. Best results were observed by an ensemble of random forest, decision tree and k-nearest neighbor classifier, when 83% of the input sequences were correctly classified. Success rate for the text class prediction was found to be 92% and 74% for the calculation and plot/graph class.



(a) Person dependent results (b) Person independent results without context information (c) Person independent results with context information

Figure 4. Normalized confusion matrices for ensemble classifier

3.3.2. *Person independent results*

In the person independent evaluation, train and test sets were split based on the users. It means that for a particular user, his data can either be used in training or testing phase, but not in both. We used the same base and ensemble classifiers with the same configuration to establish a strong comparison with the person dependent setting.

KNN achieved an overall accuracy of 75.1% for person independent configuration with the class-base accuracy of 95%, 52% and 59% for text, calculation and plot/graph, respectively, as shown in Table 2. Decision tree classifier produced an overall prediction accuracy of 69.3%, by predicting 83%, 62% and 68% of the sequences from text, calculation and plot/graph class correctly.

Random forest classifier again surpassed the performance of the other base classifiers when evaluated in a person independent setting. Overall correct prediction rate for random forest classifier was found to be 80.8%. Random forest classified 83% of the text sequences, 75% of the calculation sequences and 77% of the plots/graphs correctly as highlighted in Table 2.

By evaluating an ensemble classifier, we achieved an overall accuracy of 80%. For person independent setting, our presented approach correctly classified 87% of the text, 70% of the calculation and 73% of the plots/graphs sequences, as shown in Table 2.

Table 2. Person independent performance analysis of base classifiers

Classifier	Overall Accuracy(%)	Text (%)	Calculation (%)	Graph (%)
KNN	75.1	95	52	59
Random Forest	80.8	83	75	77
Decision Tree	69.3	83	62	68
Ensemble Classifier	80	87	70	73
Ensemble+Context	92	98	86	78

We also tested the presented approach, to produce single result for consecutive sequences by retaining the contextual information. Multiple sequences belonging to the same class were fed to the classifier. These sequences were first classified individually. Based on the individual results, a single final prediction was made by the system utilizing the contextual information. Confusion matrix in Figure 4c highlights the classification results for the ensemble classifier when contextual information is used. By using the contextual information, the overall accuracy improved to 92%, where about 98% of the text sequences were classified successfully. Classification rate of calculation and graph/plot class was found to be 86% and 78%, respectively as shown in Table 2.

3.4. Discussions

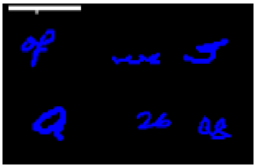
Written text is easy to distinguish since it follows clear pattern. Therefore, sequences belonging to text class are classified correctly with highest accuracy, both in base classifiers and ensemble classifiers as shown in Figure 4. Small sequences comprising of a single or couple of letters were often confused with the calculations. Therefore, the presented approach faces difficulty while distinguishing between text and calculation, when input sequence is a single stroke number or letter.

Similarly, some punctuation marks were misclassified as plot/graph. Misclassification from the text class are shown in Figure 5a and 5b.

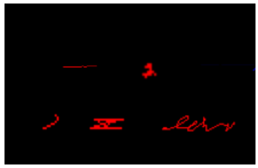
As a derivation or a calculation has a close resemblance to text production, our presented approach struggles in differentiating between calculation and text, shown as confusion matrices in figs. 4a to 4c. Individual strokes for calculation carries high resemblance with a single or two letter stroke of the text class. Improvements can be achieved by providing contextual information, as discussed later in this section. Long horizontal lines drawn to format the fractions into nominator and denominators along with square bracket signs are often confused with plot/graph class. Few misclassified sequences from calculation class are shown in Figure 5c and 5d.

Graph/plot sequences indicates clear patterns, features and structure. These patterns are significantly different from that of textual writing and calculations in visualization. In the given scenario, we consider all strokes present in the graph as plot/graph sequences including axis markers, plot legends, and/or axis labels. Therefore, some sequences were confused with text and calculation class, as shown in Figure 5e and 5f.

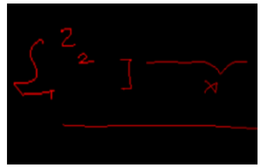
Considering that the segmented sequences composed of a single or double characters, it was quite difficult to identify whether they belonged to text or the calculation class, unless carrying specific symbols associated to a particular class. Most of the sequences in calculation were composed of characters which were very similar to text, but when combined with mathematical symbols, demonstrated significant difference be-



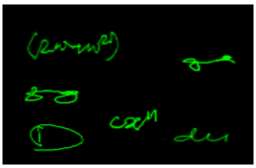
(a) Text misclassified as calculation



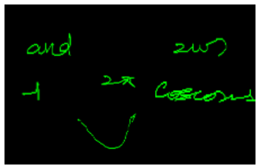
(b) Text misclassified as plot/graph



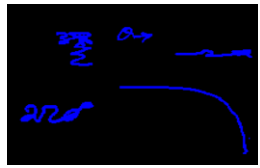
(c) Calculations misclassified as plot/graph



(d) Calculations misclassified as text



(e) Plot/graph misclassified as text



(f) Plot/graph misclassified as calculation

Figure 5. Misclassified sequences using ensemble classifier

tween text and calculation. The classifier could be improved by incorporating information regarding the neighboring individuals, which means if preceding and following sequences of input sequence belong to same class, then its more probable input sequence belongs to the same class. Therefore, contextual information was leveraged to improve the overall results by 12%. Results for text class were further improved to 98%, while for the calculation class, an increment of 16% was noticed, which was quite significant. The presented approach requires only about $1/3^{rd}$ of a second to classify a given sequence using contextual information, which makes it suitable for real-time settings. Results with and without contextual information are reported in Table 2.

4. Conclusion and future work

In this paper, we presented an approach capable of classifying on-line handwritten sequences without any prior conditions. The presented approach is generic, as it does not put any constraints on end user while attempting exercises, thus can be used in any setup.

The proposed system was evaluated on the handwriting of 20 participants and achieved a successful classification rate of 92%. Every data class was correctly predicted in a fraction of a second 78% of the time, as shown in Figure 4c. We believe that the presented approach will help instructors to oversee the performance of the students during writing, solving exercises, and/or plotting graphs.

We aim to tackle the problem of differentiating between copying text, creative writing and attempting solutions in the future work. We also plan to release the dataset publicly with multi-labeled sequences, i.e. current classification labels and copy or creative writing labels. We also propose a cognitive analysis system by incorporating gaze information in our presented approach which quantifies the stress level and difficulty of the exercises for an individual learner.

5. Acknowledgment

This work is partially funded by Higher Education Commission of Pakistan (HEC), German Research Institute of Artificial Intelligence (DFKI), and German Ministry of Education and Research in the project Be-Greifen.

References

[1] J. Moore: Handwriting classification, *The Police Journal* **vol. 18** (1945), 39–61.

[2] T. L. Smith: Six basic factors in handwriting classification, *J. Criminal Law and Criminology*, (1954).

[3] V. Bouletreau, N. Vincent, R. Sabourin, and H. Emptoz: Synthetic parameters for handwriting classification, *ICDAR'97* (1997), 102–106.

[4] L. Schomaker, G. Abbink, and S. Selen: Writer andwriting-style classification in the recognition of onlinehandwriting, *IEE European Workshop on Handwriting Analysis and Recognition* (1994), 1–4.

[5] M. Liwicki, A. Schlapbach, H. Bunke, S. Bengio, J. Marithoz, and J. Richiardi: Writer Identification for Smart Meeting Room Systems, *Seventh IAPR Workshop, DAS'06* (2005).

[6] A. Kholmatov and B. Yanikoglu: Identity authentication using improved online signature verification method, *J. Pattern recognition letters* **vol. 26** (2005), 2400–2408.

[7] M. Liwicki and H. Bunke: HMM-Based On-Line Recognition of Handwritten Whiteboard Notes, *Tenth International Workshop: ICFHR'06*, La Baule, France, (2006).

[8] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel: Backpropagation applied to handwritten zip code recognition, *J. Neural computation* **vol. 1** (1993), 856–890.

[9] C. C. Tappert, C. Y. Suen, and T. Wakahara: The state of the art in online handwriting recognition, *PAMI'90* **vol. 12** (1990), 787–808.

[10] N. E. Matsakis: Recognition of handwritten mathematical expressions, *Ph.D. dissertation*, Massachusetts Institute of Technology, (1999).

[11] R. Zanibbi and D. Blostein: Recognition and retrieval of mathematical expressions, *IJDAR* (2012), 331–357.

[12] H. Mouch'ere, C. Viard-Gaudin, R. Zanibbi, and U. Garain: Icfhr2016 crohme: Competition on recognition of online handwritten mathematical expressions, *ICFHR'16* (2016), 607–612.

[13] R. Xia, F. Xu, C. Zong, Q. Li, Y. Qi, and T. Li: Dual sentiment analysis: Considering two sides of one review, *IEEE transactions on knowledge and data engineering* **vol. 27** (2015), 2120–2133.

[14] J. W. Huppertz and R. Smith: The value of patients handwritten comments on hcahps surveys,*J. Healthcare Management* **vol. 59** (2014), 31–48.

[15] D. Willems, S. Rossignol, and L. Vuurpijl: Mode detection in on-line pen drawing and handwriting recognition, *ICDAR'05*, Washington DC, USA, (2005), 31–35.

[16] T. V. Phan and M. Nakagawa: Text/non-text classification in online handwritten documents with recurrent neural networks, *ICFHR'14* (2014), 23–28.

[17] E. Indermhle, V. Frinken, and H. Bunke: Mode detection in online handwritten documents using blstm neural networks, *ICFHR'12* (2012), 302–307.

[18] T. G. Dietterich et al.: Ensemble methods in machine learning, *MCS* **vol. 1857** (2000), 1–15.

[19] A. Rahman and S. Tasnim: Ensemble classifiers and their applications: A review, *arXiv preprint* (2014).

[20] G. Pirkl, P. Hevesi, P. Lukowicz, P. Klein, C. Heisel, S. Grober, J. Kuhn, and B. Sick: Any problems? a wearable sensor-based platform for representational learning-analytics. *UbiComp'16* (2016), 353-356.

[21] M. I. Malik, M. Liwicki, A. Dengel, S. Uchida, and V. Frinken: Automatic signature stability analysis and verification using local features, *ICFHR'14* (2014), 621-626.

[22] M. I. Malik, M. Liwicki, A. Dengel, and B. Found: Man vs. machine: A comparative analysis for signature verification, *J. Forensic Document Examination* **vol. 24** (2014), 21-35.

[23] C. M. Bishop: Pattern Recognition and Machine Learning, *Information Science and Statistics* New York, USA, (2006).

[24] P. H. Swain and H. Hauska: The decision tree classifier: Design and potential, *IEEE Transactions on Geoscience Electronics* **vol. 15** (1977), 142–147.

[25] L. Breiman: Random forests, *J. Machine Learning* **vol. 45** (2001).

[26] F. Pedregosa, G. Varoquaux, et. al: Scikit-learn: Machine learning in Python, *J. Machine Learning Research* **vol. 12** (2011), 2825–2830.