Identifying Users’ Emotional States through Keystroke Dynamics

Stefano Marrone[a] and Carlo Sansone[b]
Department of Information Technology and Electrical Engineering,
University of Naples Federico II, Via Claudio, 21, Napoli, Italy

Keywords: Cyberbullying, Keystroke Dynamics, Emotion Recognition, Deep Learning.

Abstract: Recognising users’ emotional states is among the most pursued tasks in the field of affective computing. Despite several works show promising results, they usually require expensive or intrusive hardware. Keystroke Dynamics (KD) is a behavioural biometric, whose typical aim is to identify or confirm the identity of an individual by analysing habitual rhythm patterns as they type on a keyboard. This work focuses on the use of KD as a way to continuously predict users’ emotional states during message writing sessions. In particular, we introduce a time-windowing approach that allows analysing users’ writing sessions in different batches, even when the considered writing window is relatively small. This is very relevant in the field of social media, where the exchanged messages are usually very small and the typing rhythm is very fast. The obtained results suggest that even very short writing windows (in the order of 30”) are sufficient to recognise the subject’s emotional state with the same level of accuracy of systems based on the analysis of larger writing sessions (i.e., up to a few minutes).

1 INTRODUCTION

Emotions play a fundamental role in human life, influencing the mental and physiological processes of our species. Emotions can be defined as complex response configurations, selected during the course of evolution to favour the adaptation of the organism to the environment, from which stimuli or representations are received that upset its equilibrium. As response mechanisms, emotions often involve similar neurophysiological and biochemical modifications, assuming a social and relational significance within the species. In other cases, however, they may manifest themselves differently, modulating according to the subjective experiences of each individual.

Affective computing, sometimes also referred to as Artificial Emotional Intelligence, is the branch of Artificial Intelligence (AI) that develops technologies able to recognise and express emotions (Tao and Tan, 2005). In virtue of this new perspective where classic AI is integrated with emotional intelligence, we now speak of emotional AI or the combination of emotional and artificial intelligence. Advances in affective computing technology have led to the growth of emotion recognition research in recent years. Systems able to perceive emotions bring multiple benefits to their users, as they are useful both to the user, who becomes more aware of the emotions he or she is showing, and to developers, who can make use of emotion recognition to make their projects adaptive to the user’s experience, as well as to support the detection of cognitive disorders, anxiety, or stress. The latter can be extremely useful in the detection of (cyber)bullying, a situation in which negative emotional situations can affect the mental health of (usually young) subjects (Sansone and Sperlí, 2021). As a consequence, several approaches have been developed for the automatic detection of emotions, for example by conducting voice intonation analysis, facial expression analysis or using physiological sensors. Yet, they usually required expensive, intrusive or hard to use hardware (Fragopanagos and Taylor, 2005).

Biometrics is a term referring to body measurements and statistical analyses intended to extract and quantify human characteristics. This technology, mostly used for users’ authentication or identification purposes (Jain et al., 2000), has increasingly been used for other aims, including entertainment and user-experience personalization (Mandryk and Nacke, 2016). Biometric approaches can be grouped into two distinct categories, based on the type of unique characteristic they try to leverage:

[a] https://orcid.org/0000-0001-6852-0377
[b] https://orcid.org/0000-0002-8176-6950

Marrone, S. and Sansone, C.
Identifying Users’ Emotional States through Keystroke Dynamics.
DOI: 10.5220/0011367300003277
In *Proceedings of the 3rd International Conference on Deep Learning Theory and Applications (DeLTA 2022)*, pages 207-214
ISBN: 978-989-758-584-5; ISSN: 2184-9277

- **Physiological**, referring to a direct physical measure of some human body parts, such as the face, fingerprint, iris, retina, voice, etc;
- **Behavioural**, referring to specific behaviours of a human while performing an action, such as handwriting, typing, speaking, and so on.

Among all, *keystroke dynamics* is considered one of the most effective and cheap (i.e., easy to implement using already available hardware) behavioural biometrics. In recent years, it has been more and more used to enforce user authentication by analysing habitual rhythm patterns as they types on a keyboard (both physical or virtual), so that a compromised password will not necessarily result in a compromised system (Karnan et al., 2011).

In this work, we instead focus on the use of keystroke dynamics for user emotions recognition. We believe that it could become the cheapest and most available method for emotions recognition, as the only hardware it requires is a common keyboard. Additionally, a keystroke recorder can be either hardware or software, with the latter approach being very unobtrusive, so that a person using the keyboard is unaware that their actions are being monitored resulting in an unbiased typing rhythm. In particular, we introduce a time-windowing approach that allows analysing users' writing sessions in different batches, even when the considered writing window is relatively small. This is very relevant in the field of social media, where the exchanged messages are usually very small and the typing rhythm is very fast.

The rest of the paper is organised as follows: Section 2 presents the EmoSurv dataset as well as the approach used for recognising emotions on short writing windows; Section 3 shows the experimental setup, while the obtained results are reported in Section 4. Finally, Section 5 draws some conclusions and reports future works.

2 THE EmoSurv DATASET

Neutral	’Once there was a cat and a mouse. Usually, cats eat mice, and mice run away from cats. But this cat and this mouse liked each other very much. They liked each other so much that they lived together.’
Happy	’We can not help falling in love with cute and funny babies. Their beautiful and joyful laughter makes us happy.’
Calm	’Beautiful nature and calm music are always relaxing. The soft sounds are so pleasant to listen to. Watching the superb nature calms our body and soul.’
Sad	’The boy yells at his father to wake him up. But the father is dead. The poor boy is very sad. He realized that his father is gone forever.’
Angry	’Jake has a horrible temper, especially when he drinks alcohol. He gets angry and aggressive when he is drunk. He savagely beats and violates his wife.’

Figure 1: The EmoSurv fixed texts, by emotion.

EmoSurv(Maalej and Kallel, 2020) is a recent dataset containing keystroke data for 124 subjects along with the associated emotion labels, grouped into five classes: Anger, Happiness, Calmness, Sadness, and Neutral State. Timing and frequency data were recorded while participants were typing free and fixed texts before and after a specific emotion was induced through the visualisation of a video on an interactive web application[1]. To perform the data collection process, the application guides each participant to go through the following tasks:

1. The subject has to answer a list of questions about some demographic characteristics such as age range, sex and number of fingers he uses to type. The answers are stored in a table;
2. The subject has to type a free and a fixed text before the emotion the induction process, assuming that they are in their neutral state;
3. A specific emotion-eliciting video is shown;
4. When the subject is done visualising the whole video, they are asked to answer some focus-checking questions to make sure they watched the entire video and were not distracted. If the answers are wrong, the relative data are discarded;
5. The participant is asked to type a free and a fixed text just after finishing the video;
6. Finally the subject can choose to leave the application, or continue the data acquisition process and watch another video to experience a different emotional state.

The dataset also comes with some pre-extracted features, based on digraphs and trigraphs, namely combination of two or three consecutive keystroke events. The data is organised into four .csv files:

- The **Fixed Text Typing Dataset** was collected while the participants were typing a fixed text (e.g., copying a prompted message) and it includes features such as the user id, the emotion index (e.g., ‘H’ for Happy, ‘S’ for Sad, etc.), the specific pressed key, the answer to the focus-related question and seven features associated with keys press-release combinations and timing;

[1] www.emosurv.tech

Table 1: Number of sentences (#Sen), of characters (#Char) and average writing time (AvgT, in seconds, over all the recorded sessions) for the fixed-text sentences.

Emotion	#Sen	#Char	AvgT
Angry	24	4158	59
Calm	31	5156	68
Happy	36	4514	52
Neutral	116	26762	110
Sad	32	4793	60

- The **Free Text Typing Dataset** was collected while the participants were typing a free text and includes the same features as the Fixed Text Dataset;
- The **Frequency Dataset** includes frequency related features, such as the relative frequency of the delete and backspace key, and the time required to write the sentence;
- The **Participants Information Dataset** includes demographic information such as gender, age range, status, country, etc. It also contains information about the writing style, such as whether the participant types with one hand or two hands, using one or more fingers.

For each of the five classifiable emotions there is a corresponding emotion-inducing video and a fixed sentence the participant is asked to type after watching the video. These fixed texts are inherent to the emotion inducted and are of different lengths (Fig. 1). Each subject has at least typed the sentence relating to the neutral emotion, but not all subjects have typed the sentences relating to the remaining four emotions. This depends on which video was shown to a specific subject, and on how many times he decided to repeat the data collection process. It can be easily noted how the sentence related to the Neutral emotion is the longest one among all of the sentences. Table 1 reports the number of sentences and of samples (characters), as well as the average duration, in seconds, of the typing sessions registered for each emotion.

2.1 Data Pre-processing

By analysing the dataset, it was found that the number of unique userId was equal to 83, contrary to the 124 declared in the documentation. However, it was noted that some ids were repeated multiple times in the dataset, meaning that they were assigned to more than one unique data acquisition session. For example, the user whose id is 93 was assigned to four different typing sessions. This was interpreted as a mistake in the data registration. Thus, the "UserId" column has been modified to make sure that subsequent sessions with repeated userId were assigned a different and fresh id. The same change was carefully applied in the Participant Information Dataset as well. As a result of this operation, the number of userIds increased from 83 to 116 (still less than the 124 declared in the documentation). Also, we removed all the instances (rows) presenting an erroneous value ($-1.58 * 10^{12}$) in any of the available columns or with a NaN for the "D1U1" feature (for the other features, NaN is allowed). After these operations, the number of characters of the free text dataset is reduced from 46871 to 45358.

2.2 Feature Extraction

The features already made available with the dataset are related to a single keystroke (D1U1), digraphs (D1D2, U1D2, D2, D3) and trigraphs (D1D3, D1U3). These features may not be suited for emotion recognition as they are extremely local. Instead, we believe that studying the typing rhythm of the user over a certain interval of time could result in a better performance. Thus, in this work we leverage 20 high-level features based on the dwell time (i.e., the time elapsed between a key press and the same key release), on the flight time (i.e., the time elapsed between a key release and the next key press) and on the D2D-time (down to down, i.e., the time elapsed between a key press and the next key press):

- CPMilli: number of characters pressed in the selected time window;
- Mode-dwell: mode of the dwell time of keys pressed in the selected time window;
- stdDev-dwell: standard deviation of the dwell time of keys pressed in the selected time window;
- stdVar-dwell: standard variation of the dwell time of keys pressed in the selected time window;
- range-dwell: range of the dwell time of keys pressed in the selected time window;
- min-dwell: minimum dwell time of keys pressed in the selected time window;
- ax-dwell: maximum dwell time of keys pressed in the selected time window;
- mode-flight: mode of the flight time of keys pressed in the selected time window;
- stdDev-flight: standard deviation of the flight time of keys pressed in the selected time window;
- stdVar-flight: standard variation of the flight time of keys pressed in the selected time window;
- range-flight: range of the flight time of keys pressed in the selected time window;

- min-flight: minimum of the flight time of keys pressed in the selected time window;
- max-flight: maximum of the flight time of keys pressed in the selected time window;
- mode-d2d: mode of the down to down time of keys pressed in the selected time window;
- stdDev-d2d: standard deviation of the down to down time of keys pressed in the selected time window;
- stdVar-d2d: standard variation of the down to down time of keys pressed in the selected time window;
- range-d2d: range of the down to down time of keys pressed in the selected time window;
- min-d2d: minimum of the down to down time of keys pressed in the selected time window;
- max-d2d: maximum of the down to down time of keys pressed in the selected time window;
- num-deletes: number of times the backspace key was pressed in the selected time window.

In previous works on this topic, similar high-level features were extracted while taking into consideration the entire typing session, i.e., the time it took the user to type the fixed sentence. As already pointed out before, in this work we want to build a model able to identify users' emotions even when the available typing session is not very long. Therefore, a sliding window mechanism was applied by considering only the keys the user pressed over a fixed time window (for example, 10 seconds). This means that, given the registered data for a session and a fixed time window, the value of every high-level feature was calculated only for the keys pressed in every time window identified. As a consequence, from each sentence we extract a matrix having:

- Exactly 20 columns (one for each feature);
- A number of rows different for each sentence, based on *i*) the total time it took the participant to type the requested, *ii*) the chosen time window and *iii*) the considered stride value (i.e., how much the windows are distant each other during the sliding operation).

We also leverage the demographic features, from the Participants Information Dataset. It is worth noting that these features are unique for each subject (and, of course, for each emotion registered for that subject) and they thus assume the same value regardless of the considered window.

Table 2: Number of rows (i.e., number of extracted windows) for each emotion of subject ID 41, as the windows size and stride (in seconds) vary.

Emotion	WS/Stride			
	15/3	15/4	15/7,5	10/5
Neutral	18	14	8	12
Calm	13	10	5	9
Sad	15	12	6	10

2.3 Windowing and Sample Size

As mentioned in Section 1, in this work we propose to perform text-based emotion recognition by introducing a time-windowing approach that allows analysing users' writing sessions in different batches, even when the considered writing window is relatively small. As a consequence, the feature extraction process (Sec. 2.2) can be strongly impacted by the values chosen for the window size and the stride, with the number of rows extracted from each sentence being inversely linked to those parameters (Tab. 2). Given the characteristics of the considered dataset, in this work we will use 15 seconds and 7.5 seconds for the window size and the stride respectively.

It is worth noting that the number of rows extracted from each sentence still depends on the length (number of words) of the sentence itself. However, in some situations it would be preferable to work with samples all having the same size. In our windowing scenario, a possible solution to achieve this is to fix the desired number of **R**ows in each **S**ample (RS from now on) and extract several samples from the same sentence by considering (possibly overlapping) subportions of the original features matrix, all having the same RS value. Figure 2 illustrates this procedure for a feature matrix of 7 rows, considering RS set to 3. As a consequence, for each user and sentence (i.e., emotion registered for that user), there will no longer be a single sample but multiple sub-samples, all sharing the same class. An interesting side-effect of this approach is an increase in the available training samples. It must be noted that once a particular RS value is chosen, all samples consisting of a number of rows smaller than RS have to be removed. Therefore, the total number of subjects and of unique sentences may decrease. Table 3 reports a brief overview of this aspect, in terms of users, sentences and samples.

It is also worth noting that considering the original samples having different number of rows or extracting sub-samples all having the same number of rows are both viable approaches to the problem. In both cases, as a sample consists of several rows (variable in the former, fixed in the latter) we are facing a

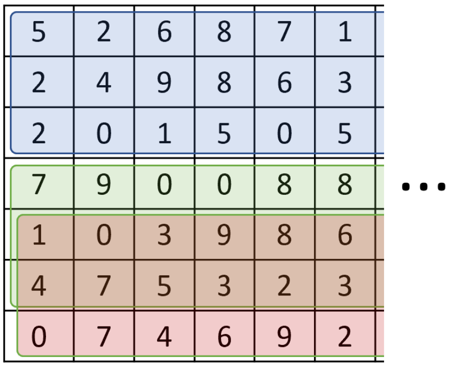
Table 3: Number of usable subjects and corresponding number of sentences and sub-samples as the considered RS value vary.

RS	#Subjects	#Sentences	#Sub-Samples
3	115	231	705
4	113	218	547
5	111	202	444
6	108	174	350

multi-instance problem. In the following sections, we will compare both approaches, comparing them under different configurations.

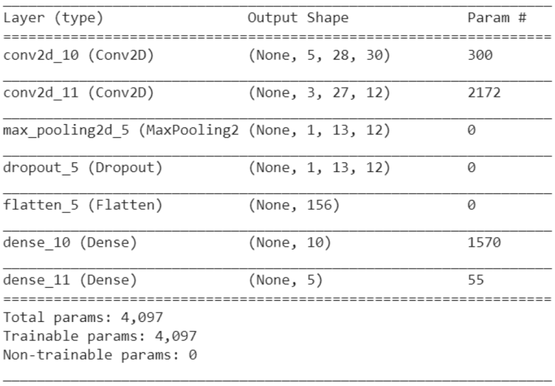
3 EXPERIMENTAL SETUP

As described in the previous section, we are facing a multi-instance problem with samples consisting of matrices having *i*) the same number of columns (features), *ii*) different or fixed number of rows (based on the considered approach) and *iii*) assigned to one of the five possible classes available in the EmoSurv dataset. Concerning the latter aspect, despite the problem could be addressed as a multi-class classification task, given the reduced number of available samples in this work we face the problem by using a 1-vs-all binarization approach. This means that we will consider five models (one per class), each trained on the binary task of determining whether the considered sample belongs to the model's class or not. During the inference stage, all the new samples will be fed to all the five models, using a voting strategy to determine the class. In this work, we will explore *majority voting* and *highest-probability* strategies.



5	2	6	8	7	1
2	4	9	8	6	3
2	0	1	5	0	5
7	9	0	0	8	8
1	0	3	9	8	6
4	7	5	3	2	3
0	7	4	6	9	2

Figure 2: Illustration of the sub-sample extraction strategy with an RS value of 3 for a feature matrix consisting of 7 rows and some columns (represented by the dots). The three colours highlight the obtained three sub-samples. It is worth noting that as the number of rows is not a multiple of RS, the last two sub-samples are partially overlapped.



```
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_10 (Conv2D)           (None, 5, 28, 30)         300
_________________________________________________________________
conv2d_11 (Conv2D)           (None, 3, 27, 12)         2172
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 1, 13, 12)         0
_________________________________________________________________
dropout_5 (Dropout)          (None, 1, 13, 12)         0
_________________________________________________________________
flatten_5 (Flatten)          (None, 156)               0
_________________________________________________________________
dense_10 (Dense)             (None, 10)                1570
_________________________________________________________________
dense_11 (Dense)             (None, 5)                 55
=================================================================
Total params: 4,097
Trainable params: 4,097
Non-trainable params: 0
_________________________________________________________________
```

Figure 3: Structure of the proposed CNN.

Focusing on the multi-instance side of the problem, each sample is thus a bag consisting of all the rows composing the matrix. The size of these bags is variable (see Section 2), but can be fixed by using the windowing approach described in section 2.3. Despite Multi-Instance Learning (MIL) does not necessarily require bags consisting of the same number of elements (Foulds and Frank, 2010), this opportunity opens up different experimental scenarios. In particular, in this work we will:

- train a MIL Support Vector Machine (SVM) model (Andrews et al., 2002; Doran and Ray, 2014) on the original bags. This implies that each sentence corresponds to a single bag containing all the features extracted in all the time windows considered for that typing session;
- train a MIL-SVM model on the dataset consisting of the fixed-size bags. This implies that each sentence corresponds to one or more bags, each containing all the features extracted in the time windows associated with a sub-portion of that typing session;
- train a Convolutional Neural Networks (cNN) on the fixed-size bags.

Focusing on the MIL-SVM setups, since all the items in each bag are associated with the same class, in this work we will leverage the Normalized Set Kernel (NSK) approach (Gärtner et al., 2002) to use a MIL-aware kernel to map entire bags into features, before using the standard SVM formulation to find bag classifiers. Moving to the CNN setup, it is worth noting that the fixed bags dataset make it possible to consider each sample as a sort of "image-like feature-map", able to keep track of both semantic (on the columns) and temporal (on the rows) features. This said, in this work we designed a simple CNN from scratch, consisting of two convolutional layers, a max pooling, drop-out (set to 0.3 to reduce overfitting) and two dense layers (Figure 3).

3.1 Class Balancing

As described in section 2, the dataset is widely unbalanced towards the "neutral" class. While SVM can deal with this problem, the considered CNN may easily tend to diverge, especially as a consequence of the reduced size of the considered dataset. Thus, in this work we experiment with the use of the proposed CNN together with four different balancing techniques:

- **Class Weights:** different weights are assigned to both the majority and minority classes to prevent the considered models from predicting the more frequent class more often than the others;
- **Random Undersampling:** consists in reducing the number of samples of the majority class through a random selection of samples to be dropped;
- **Oversampling:** consists in increasing the number of samples of the minority classes through the use of SMOTE (Chawla et al., 2002), an approach performing synthetic data augmentation based on the original training data;
- **Under-oversampling:** consists of deleting the samples from the majority class (as in random undersampling) before duplicating the samples from the minority classes (as in oversampling).

An interesting side effect of the oversampling approach is in the further increase of the training data.

4 RESULTS

This section reports the results of the proposed analysis. All the tests were run by using a 60/20/20 stratified subject-based hold-out random splitting to generate the training, the validation and the test set respectively from the considered dataset. The number of rows for samples (RS) has been set to 5, to consider not too small samples while also not excluding too many subjects. Also, to avoid unfair results, all the balancing techniques (sec. 3.1) were applied only to the training set. The used MIL-SVM classifier is based on a Python implementation publicly available on git-hub[2], while the considered CNN has been crafted from scratches in PyTorch. All the other preprocessing and elaborations (including data cleaning, balancing, etc.) were performed in Python. The experiments were run on a physical server equipped with 2x Intel(R) Xeon(R) CPUs (4 cores each, running at 2.13GHz), 32GB of DDR4 RAM and an Nvidia Titan XP GPU (Pascal family) having 12GB DDR5 RAM, hosted in our HPC center.

Table 4: Comparison of the analysed setups, in terms of classification accuracy (Acc), precision (Pre), recall (Rec) and F1-score (F1), varying the bag type (Fixed Bags - FB, Variable Bags - VB), the balancing technique (Class weights - CW, Undersampling - US, Oversampling - OS, Under-oversampling - UOS) and the voting approach (Highest probability voting - HPV, Most-frequent voting - MV). Best results are reported in bold.

Approach	**Acc**	**Pre**	**Rec**	**F1**
CNN CW-HPV	0.48	0.58	0.48	0.50
CNN CW-MV	0.44	0.56	0.43	0.43
CNN US-HPV	0.57	0.43	0.57	0.48
CNN US-MV	0.57	0.43	0.57	0.48
CNN OS-HPV	0.46	0.45	0.46	0.43
CNN OS-MV	0.41	0.43	0.41	0.40
CNN UOS-HPV	0.52	0.48	0.52	0.49
CNN UOS-MV	0.54	0.5	0.54	0.5
MIL-SVM VB	**0.76**	**0.80**	**0.69**	**0.74**
MIL-SVM FB-HPV	0.52	0.6	0.52	0.53
MIL-SVM FB-MV	0.48	0.52	0.48	0.47

Table 4 reports the results obtained by using the analysed setups, in terms of classification accuracy, precision, recall and F1-score. In particular, we report the results obtained by varying the bag type (between fixed or variable size, as in Section 2.3), the balancing technique (Setion 3.1) and the voting approach (Section 3), highlighting in bold the top-performing in each column. It is interesting to note that the best approach results to be the MIL-SVM trained on variable-size bags, with all the other approaches achieving significantly lower results. This result was somehow expected, as the problem is multi-instance and with variable bags by nature.

To better frame the results achieved by using the proposed approach, we compared it against the only available competitor[3] on the considered dataset. The project, publicly available on GitHub[4], introduces a new set of features based on edit distances to capture the number of typos typed by the subject, assessing their effectiveness for emotion recognition. It is worth noting that the competitor approach also extracts some high-level features very similar to the ones defined in this work (such as D1U1_mean, D1U1_std, D1U2_mean D1U2_std, and so on). The main difference with the competitor approach is that it extracts these features by considering the whole typed sentence (opposite to what we propose, by using a sliding time window). As classification algorithm, the competitor approach uses XGBoost (Chen et al., 2015).

[2] garydoranjr/misvm

[3] At the time of writing this article

[4] alodieboissonnet/EmotionRecognitionKeystrokeDynamics

Table 5: Comparison of the best performing proposed approaches versus the considered competitor, in terms of classification accuracy (Acc), precision (Pre), recall (Rec) and F1-score (F1).

Approach	**Acc**	**Pre**	**Rec**	**F1**
Proposed approach	0.76	0.80	0.69	0.74
Competitor	0.80	0.81	0.80	0.79

Table 6: Per-class comparison of the best performing proposed approaches versus the considered competitor.

	Angry	Calm	Happy	Neutral	Sad	Competitor
Accuracy	0.56	0.70	0.80	0.93	0.78	0.80
Precision	0.76	0.62	0.89	0.93	0.79	0.81
Recall	0.58	0.58	0.70	0.94	0.64	0.80
F1-Score	0.58	0.60	0.78	0.94	0.71	0.79

To ensure a fair comparison, we took care performing the same subject selection and using the very same test set.

Table 5 reports the comparison of the best performing proposed approach versus the considered competitor. Results show that the competitor performs slightly better than the proposed approach. However, it is worth noting that while we analyse a 15 seconds long time windows, the competitor operates on the whole typing session that has, for the considered dataset, an average duration of 84 seconds. This makes the competitor approach unsuited for the short messages scenario (e.g., Twitter, instant messaging app, etc.). To further analyse the proposed approach, in table 6 we report the same information but organised per class. Interestingly, these results show that our approach performs comparably or better w.r.t. the competitor for three classes, with "Angry" and "Calm" pushing down our average performance.

5 CONCLUSIONS

In this work we analysed the detection of users' emotional states based on the analysis of written text focusing on the case of short writing sessions (i.e., up to a few seconds), typical of modern instant messaging applications. To this aim, we leverage keystroke dynamics, a behavioural biometric analysing habitual typing patterns on a keyboard. In particular, we introduced a time-windowing approach that allows analysing users' writing sessions in different batches, re-shaping the emotion recognition task into a multi-instance problem. The obtained results suggest that even very short writing windows (in the order of 30") are sufficient to recognise the subject's emotional state with the same level of accuracy as systems based on the analysis of larger writing sessions (up to a few minutes). Despite promising, it is worth noting that the use of keystroke dynamics also presents some challenges that need to be addressed, including possibly low generalisation (as the values of keystroke parameters taken from a specific user may depend on the type of software used) and inconsistencies in the users' typing rhythm due to external factors (e.g., injury, fatigue, or distraction) instead of emotions.

Future works will focus on increasing the dataset through some new data augmentation techniques, to also balance the number of instances per class. Also, we will investigate whether keystroke dynamics can be combined with other biometrics or with other text-based analyses (e.g., sentiment analysis) to further improve the recognition performance. Finally, as in this work we only used the Fixed Text Dataset (sec. 2), a further step will be testing the effectiveness of the proposed approaches on the Free Text Dataset, which is related to the subjects typing rhythm as they wrote spontaneous sentences after watching videos. The hope is that this type of analysis would better integrate with users' daily activities and, of course, with chat messages analysis, possibly providing more reliable, stable and precise predictions.

ACKNOWLEDGEMENTS

This work is supported by the Italian Ministry of University and Research (MUR) within the PRIN2017 - BullyBuster - A framework for bullying and cyberbullying action detection by computer vision and artificial intelligence methods and algorithms (CUP: F74I19000370001). The authors are also really grateful to SCoPE group from the University of Naples Federico II for the given support.

REFERENCES

Andrews, S., Tsochantaridis, I., and Hofmann, T. (2002). Support vector machines for multiple-instance learning. *Advances in neural information processing systems*, 15.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.

Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., Chen, K., et al. (2015). Xgboost: extreme gradient boosting. *R package version 0.4-2*, 1(4):1–4.

Doran, G. and Ray, S. (2014). A theoretical and empirical analysis of support vector machine methods for multiple-instance classification. *Machine learning*, 97(1):79–102.

Foulds, J. and Frank, E. (2010). A review of multi-instance learning assumptions. *The knowledge engineering review*, 25(1):1–25.

Fragopanagos, N. and Taylor, J. G. (2005). Emotion recognition in human–computer interaction. *Neural Networks*, 18(4):389–405.

Gärtner, T., Flach, P. A., Kowalczyk, A., and Smola, A. J. (2002). Multi-instance kernels. In *ICML*, volume 2, page 7.

Jain, A., Hong, L., and Pankanti, S. (2000). Biometric identification. *Communications of the ACM*, 43(2):90–98.

Karnan, M., Akila, M., and Krishnaraj, N. (2011). Biometric personal authentication using keystroke dynamics: A review. *Applied soft computing*, 11(2):1565–1573.

Maalej, A. and Kallel, I. (2020). Emosurv: A typing biometric (keystroke dynamics) dataset with emotion labels created using computer keyboards.

Mandryk, R. L. and Nacke, L. E. (2016). Biometrics in gaming and entertainment technologies. In *Biometrics in a Data Driven World*, pages 215–248. Chapman and Hall/CRC.

Sansone, C. and Sperlí, G. (2021). A survey about the cyberbullying problem on social media by using machine learning approaches. In *International Conference on Pattern Recognition*, pages 672–682. Springer.

Tao, J. and Tan, T. (2005). Affective computing: A review. In *International Conference on Affective computing and intelligent interaction*, pages 981–995. Springer.