”The smell of fear” - In Search of Patterns Between Emotions and Breath Composition - Literature Survey

Jason Tam[1], Musa Abdel-Rahman[2], Catherine Liu[3]
Supervised by Jörg Wicker

I. INTRODUCTION

An important process of many biological organisms is gas excretion. The gas excretions of humans compose of Carbon Dioxide and various Volatile Organic Compounds (VOCs) such as isoprene and acetone. These VOCs have been found to play an important role in the communication between plants, and even between plants and animals, however we do not know if these VOCs have any impact on human communication. Research within the field to understand how these VOCs interact or are correlated with human emotions have been attempted by [1].

The intention of this study is to continue on from the accomplishments mentioned in [1]. The current plan is to further explore the potential patterns/associations that are intrinsically present in the collected dataset, starting with general pattern mining strategies with unsupervised learning techniques, and may apply supervised learning methods for finer details of the resulting patterns if feasible. This paper explores techniques that are potentially useful for various analysis possibilities of the data, such as using *Dynamic time warping* to align different sessions of the same movie, or clustering techniques that may reveal high volume detection in certain molecular mass channels are trigged by certain scene types.

II. PREVIOUS WORK ON THE SUBJECT

A. Breath Composition and Emotional States

Previous research has already been conducted in this area to try and identify and understand VOCs and the impact they have on a person’s emotional state [1] [2]. Within these studies, Wicker et al utilized data collected from a cinema in Mainz, Germany which included concentrations of VOCs in the cinema, and also information on the content of the movie at the time of each measurement (this was given the variable ’scene label’). From this, they tried to identify the relationship between VOC concentrations and the ’scene labels’. Four approaches were used to analyze the data sets:

- Using regression to predict future VOC concentrations given the past VOC concentrations and scene labels (forward prediction)
- Using abductive reasoning to help predict past scene labels from future VOC concentrates (backward prediction)
- Validation of predictions using forward learners as input to the backward learners
- Using a modified Granger causality methodology where comparisons are made between performance with and without a particular time series

Ten minute time frames were examined as various factors were taken into consideration such as the delay in time for the breath gases to reach the mass spectrometer, external factors such as audience influences, or the misalignment of data. Within the first method of forward prediction, trees were utilized to help explore and understand the data, as this helped to show dependencies that occurred within the data especially in understanding what scenes made particular VOC concentrations go up. Abductive reasoning was used in the second approach to explore relationships and to validate the prediction of future events in time series. Again, trees were utilized to help visualize any dependencies or any relationships. Random Forest algorithms were used to construct models for each mass, CO_2 and scene labels. Using these methods it was found that good models could be trained on those variables that evoked strong emotional responses, such as blood, and death scenes. This was proven to be the case both in repeat screenings of the same movie or within different films with the same scene label.

It may be beneficial for us to utilize regression models and trees to find relationships within our data sets as this is a simple and easy way to understand any relationships that may occur, which can also be easily explainable. However, it is important to remember that when using random forests to construct our models, that thorough data preparation has been undertaken as random forests have been observed to over-fit for some datasets with noisy classification or regression tasks.

B. Gas Sensing Systems and Emotion Recognition

Takahashi and Sugimoto [3] also conducted similar experiments in which they used a smart gas sensing system to achieve emotion recognition using breath composition. The gas sensing system was comprised of a sensor that was made using a quartz crystal resonator with plasma-polymer film. This sensor was able to pick up VOCs under dry air conditions. They collected samples from 20 subjects, who partook in questionnaires that were based on the multiple mood scale. The questions evoked emotions, which in turn were captured using the smart gas sensing system. A dental rinse was used to excite the emotion, and readings were taken before and after using this rinse. From this, two

[1] jtam013@aucklanduni.ac.nz
[2] mabd292@aucklanduni.ac.nz
[3] cliu129@aucklanduni.ac.nz

emotions, comfortableness (having a positive emotion and feeling refreshed) and no emotion were considered. Using this data, machine learning approaches were utilized to explore and validate if breath composition was feasible for emotion recognition.

Takahashi and Sugimoto first explored using Neural Networks to explain their data. A (16-12-8-6-2) network was used, where the activation function of the neurons was a linear function in both the input and output layers, and a sigmoid function applied to the hidden layers. Trial and error was used to find the number of neurons they would use in the hidden layers. A 'leave-one-out' cross-validation was employed to evaluate the validity of the model. From this the average recognition rate was 47.5%, with the detection of comfortableness being 55%, and having no emotion resulted in an even lower recognition rate and was a lot harder for the Neural Network to detect. Support Vector Machines (SVM) were also utilized to test emotion recognition from breath gas information. During the emotion recognition process, the feature vector was fed into the SVM and the output from the SVM investigated in the decision logic that selected the best emotion, and the class of the SVM indicated the recognition result. The kernel function used a gaussian function and the margin parameter and function parameter were defined by trial and error. Again, the 'leave-one-out' cross-validation method was used to evaluate the validity of the model. From this, a higher average recognition rate of 67.5% was achieved.

Following this, Takahashi and Sugimoto extended the aforementioned research by introducing another emotion into the experiment - displeasure [4]. In terms of experiment methodology, all equipment and testing were kept constant to the previous study. Again, Neural Networks were employed in trying to achieve emotion recognition using breath gas information. In alignment with Takahashi and Sugimoto's previous study, it was found no emotion was hard to recognize, and after the introduction of displeasure into this experiment, this too was also hard for the neural network to recognize. In the neural networks explored, the output layer contained 3 neurons with varying results for average emotion recognition based on the neural network structure. It was found a (7-11-11-3) network had less than 50% success rate for the average recognition rate. This average recognition rate increases to 75% once a (14-14-6-3) network is introduced, however this comes at the expense of a reduced rate of pleasure recognition. Again, the number of neurons in the hidden layers and the learning factors were deemed through trial and error in order to converge the learning of the neural network.

Within the SVM model, the 'one-vs-all' method was utilized due to the multiclass classification of the dataset. Moreover, the feature vector was fed into the SVMs during the emotion recognition process, and the output from each SVM was investigated in the decision logic that selects the best emotion. A 3rd order polynomial was used as the kernel (and c as 1 and 10 for the two models they fit) and trial and errors used to find other parameters in order to give the training data a complete classification rate. The best performing SVM model that Takahasi and Sugimoto created had an average recognition rate of 83% -higher than the neural network model and also higher than the previous SVM models they made when only two emotional states were explored.

Through both these experiments [3] [4], the results suggest that machine learning models could be employed to understand and predict emotion through breath gas information. In both the Neural Network and SVM models, we see a lot of trial and error is involved in computing the parameters involved to obtain the best average recognition rates. This can be a tedious process and would need to be recalculated whenever new data was uploaded and require much more expertise to tune. Within the Neural Network, we see advantages as the architectures can be easily adapted to any situation and their hidden layers reduce the need for feature engineering. However, they are computationally intensive to train, and they require much more expertise to tune. SVM models are also hard to tune due to the importance of picking the right kernel and may not scale well to larger or other data sets, so this will need to be taken into consideration. Once all these factors are taken into consideration, these supervised learning techniques may still be beneficial for us to utilize with our data set if we already know what we wish to classify or predict.

C. *Classification of Emotional Biosignals*

Similar work was done by Frantzidis et al [5] where data mining was utilized to explore the classification of emotional biosignals that were evoked through pictures selected from the International Affective Picture System (IAPS). The IAPS pictures were rated in terms of valence (pleasure) and arousal. Using data mining techniques in the way of a C4.5 decision tree algorithm, the biosignals were classified according to their valence dimensions. These, along with the gender of the subject, became inputs that were fed into a Mahalanobis classifier which categorized the data into high and low arousal. Using this combined data mining and pattern recognition technique saw an average recognition rate of 77.68% for the four emotional states (these states differed both in their arousal and valence dimensions).

This experiment looks at supervised learning once classifications for the data set have been made, and uses combined methodologies to arrive at the average recognition rates. This is helpful to note and utilize in further work as most data preparation, transformation and modeling may go through many iterations and many different models before coming to a result we are happy with. Using a C4.5 decision tree is any easy way to come to an easily interpretable classification however it is important to know and understand the data you are working with, as small variations in the data can lead to different decision trees. This would mean that very different results could be concluded each time a different data set was run. It can be seen in both this project and the aforementioned project by Wicker et al [1] [2] that employing trees in interpreting and understanding relationships between the data could be a great first step in explanatory analysis for our project.

D. Data Analysis on VOCs

It was found that Electric nose systems, (an instrument consisting of an array of reversible but only semi-selective gas sensors coupled to a pattern recognition algorithm) (also known as e-noses) have been widely used for the analysis of VOCs [6]. Scott, James and Ali offer various ways in which the data analysis from these systems can be analyzed, which will be explored further in this literature review. It is essential to combine many analytical techniques together to help classify data as more often than none relevant data may be lost when trying to reduce the dataset on a whole. It is interesting to note that the classification algorithm chosen to model the data is very important - choosing a poor algorithm results in a model that may lack the ability to model non linear data, or it may result in an over trained and over-fit model.

III. MACHINE LEARNING

Machine learning [7] is a field of data science that focuses on 'machines' extracting relationships/patterns that are intrinsically present in data. 'Machines' refers to all systems that can make use of algorithms to perform computations, as well as the ability to store and recall the intermediate results of such computations in order for final solutions to converge. It is an extension on the traditional methods from statistics with concepts from artificial intelligence, and the techniques used to process data can generally be categorized into *supervised* and *unsupervised* learning. Supervised learning represents scenarios where a target response variable is present within the data, and chosen algorithms are to determine how the target response is affected by the other input variables in the data. The result would provide a logic that determines a likely output when a new input value is given, which could be labels with classification algorithms or numerical values with regression algorithms. Unsupervised learning represents scenarios where a target response is not specified, and the algorithm is to determine if any underlying patterns/relationships are present in the data, which is treated as unstructured noise.

The objective of this study is centred around determining potential relationships between breath composition and emotion, via a collected dataset that contains gas composition detected in cinema sessions of different movie types. While there is no clear target response variable for the analysis to predict, one can assume it might be possible to predict the traces of some molecular masses that would be captured by the VOC device once enough patterns are extracted the data, given the attendance data and scene categorizations of the movie.

A. Supervised Learning

Techniques that are categorized as supervised learning [7] are useful when the target response variable is included in the input to the algorithm. This approach aims to determine the relationship between the target response and the other variables in the data, and use it for prediction on new input variables. Common techniques include ones that are traditionally from the statistics domain, such as *linear regression* and *logistic regression*, as well as ones that are later developed under the data science domain such as *k-nearest neighbour*, *random forests*, *decision trees*, *support vector machines* and *neural networks*.

This approach is most commonly used in analysis where obtaining accurate predictions have the highest priority, while the explanation of the results is often secondary. For this reason, such analysis often take the approach of partitioning the data into training and testing sets. Building independent prediction models using different methods with the training data, and choose the one with the best prediction performance using the testing data. As most of the algorithms that were developed outside of the traditional statistics domain favours prediction in the expense of explanation (which are often referred to as 'black box methods'), hence such techniques are often avoided when causal inference is required as part of the analysis result.

B. Unsupervised Learning

Unsupervised learning [8] covers a broad range of algorithm concepts, with each of them emphasized in certain strengths. Almost all of them share a common philosophy, which is essentially determining a probabilistic model for new input variables based on the existing data. For simpler cases where the data is *unstructured*, which means the order of the input's arrival is irrelevant, one can assume that they are all drawn independently from the same underlying distribution $P(x)$ in an identical manner. This setting is particularly useful for analysis objectives such as *outlier detection* or general *monitoring*. *Classification* problems can also make use of this by determining a probability distribution for each of the labels in existing data, which are then used to classify the new inputs based on their attributes. While such a framework can be applied to a wide range of models, the objective in the practice of machine learning remains to be finding/developing models that are most appropriate for the data set to be analyzed while being aligned with certain predetermined goals.

Latent Variable Models

There are some techniques that are used to construct probabilistic models based on some latent or hidden variables, which are quite commonly used when performing *dimensional reduction* and *clustering*, considered corner stones of unsupervised learning.

Factor Analysis is a popular method used to deal with high-dimensional datasets. Datasets that have high number of dimensions may sometimes be a real challenge for the machines, because there can be too many parameters to reliably estimate even with the simplest algorithms. Factor analysis makes use of multivariate gaussian vectors to estimate a gaussian density for high dimensional data, and provide a low-dimensional representation of the data and associated results. *Independent Components Analysis (ICA)* is an extension to this technique that deals with data with factors that are not gaussian-distributed, which is suitable for many real world datasets that often have a certain degree of sparseness.

Principal Component Analysis (PCA) is another common method used to simplify high-dimensional data. It transforms a set of observations that are possibly correlated into a set of linearly independent ones, which are termed **principal components**. This resulting set is often used as input to other algorithms, and their linear independence often assist greatly in reducing calculation complexity.

Mixtures of such gaussian distribution are made use in clustering algorithms such as *K-means*. Datasets containing n observations are partitioned in to k clusters, where each observation would belong to the cluster with the nearest mean. This is achieved by estimating the maximum likelihood parameters using the iterative algorithms such as the *Expectation-maximization* algorithm. This is commonly used in preprocessing steps to determine initial configurations for other algorithms.

Time Series Models

While there are possible approaches to partition the dataset in a time-independent manner for this study, the observations are nevertheless collected sequentially, which potential underlying patterns may exist. Traditional methods in statistics such as the **A**uto**R**egessive **I**ntegrated **M**oving **A**verage (ARIMA) methods [9] construct probabilistic models for present and future observations with past observations. *State Space Models* (SSM) and *Hidden Markov Models* (HMM) are also common methods used to process time-series data, where they assume the observed is a consequence of data in hidden states.

IV. DYNAMIC TIME WARPING

Dynamic Time Warping [10] is a technique developed to analyze time series data, with initial inspirations from concepts in studies of speech recognition. Word recognition is mostly performed by matching existing templates in a library to the time series speech data recorded. The time axis of either sequence is often adjusted by either elongation or compression to minimize some distance measure for the two data sets to be as aligned as possible.

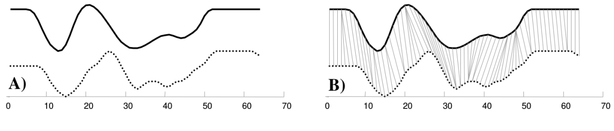

Fig. 1. A visualization of the main mechanism in the Dynamic Time Warping technique. [11]

Figure 1 shows an example from [11] of the utility of dynamic time warping. Figure 1 A) presents the vertical position of an individual's hand while expressing the word "pen" in Sign language in two separate scenarios. While an observable level of similarity is present between the two sequences, a simple approach of assuming the ith point on one sequence is aligned with ith point on the other will produce a pessimistic dissimilarity. Figure 1 B) presents an alignment produced by a more sophisticated distance measure.

While DTW has shown to be successful in many disciplines, it should be noted that the algorithm can sometime produce undesired results. Based on the principle that the algorithm attempts to explain variability on one axis (Y) by warping the other (X), it can often lead to unintuitive alignments where single points on one sequence would map to multiple points or a large section on the other sequence. Such points are commonly termed "singularities", and while compensative approaches have been proposed as alternatives, they in general restrict the possibilities of allowed warping configurations. It is possible that the "correct" warping might be prevented from discovery with such restrictions. Common definition for the "correct" warping is usually defined by the features being matched, such as peaks, troughs and plateaus on the sequence. This approach is efficient in producing a good alignment if the sequences are similar except for some local accelerations and decelerations in the horizontal (time) axis. Slight differences in these features between the two sequences (such as a peak being higher) may cause the algorithm to fail. This is demonstrated in Figure 2

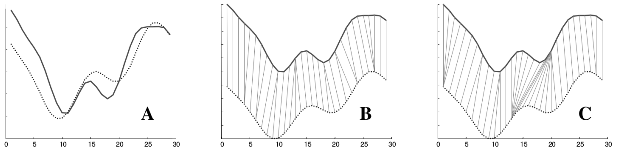

Fig. 2. A visualization of the DTW mechanism failing due to difference in features between the two sequences. [11]

Derivative Dynamic Time Warping

While this traditional approach determines the warping factor essentially by minimizing the (typically Euclidean) distances between the two sequences, it is particularly sensitive to differences in the vertical (Y) axis. While differences at the global scale such as difference means, scalings or linear trends can be removed efficiently, its biggest weakness remains to be considering only the Y values during the optimization process. This can often cause points on a rising trend to be mapped to a declining trend, which we would not intuitively agree. *Derivative Dynamic Time Warping* is an alternative technique developed by Keogh and Pazzani [11] that considers the first derivative of the sequences instead of the actual Y values. They have shown in their paper that this technique is effective against scenarios describe in Figures 1 and 2. Such techniques can potentially be very useful in comparison of time segments from datasets of different movie sessions, and determining if a correlation is present between such trends.

V. Clustering and Pattern Classification

Clustering data into groups based on similarity is often one of the most important steps in any data-analysis application. Currently, most clustering algorithms partition the data into groups that are disjoint, while other algorithms extend this approach to probabilistic or overlapping clustering [12]. The goal of data clustering is to identify homogeneous groups or clusters from a set of objects. In other words, data clustering aims to divide a set of objects into groups or clusters such that objects in the same cluster are more similar to each other than to objects from other clusters [13]. Some researchers present cluster analysis as part of broader features analysis of data and break it into clusters, it may include some of them of or all [12]:

- Factor analysis: Where we first reduce the dimensionality of data
- Clustering: Where we create clusters within data
- Discriminant analysis: Measure how well the data properties are captured

Many important applications, like e-nose systems, generate data that has a set of dependent variables and semi-independent variables, and the data vectors [6]. The goal is to uncover disparate and hidden relations between them, by using different types of exploratory and confirmatory techniques to recognize patterns. It is also useful when we have data vectors with no descriptors, therefore similarities between classes to define groups, classes, and patterns will be used, as there would not be any feedback about classes before [13]. Similarity is usually measured by a distance function defined on pairs of patterns. As there are a wide variety of feature types and scales, the distance measure should be chosen carefully. It is most common to calculate the dissimilarity between two patterns using a distance measure defined on the feature space [6]. A good clustering algorithm can be evaluated based on two primary objectives [12]:

- High intra-class similarity
- Low inter-class similarity

Many Pattern recognition Techniques have been developed in the past, and typically classified as below, based on what we assume for our underlying data [12]

TABLE I

Pattern recognition

Technique	Definition	Example
Connectivity Model	Distance Connectivity between Observations	Hierarchical Cluster Analysis
Centroid Models	Distance from mean value observation of each observation is the measure	K-means
classification	classify using majority vote	K-nearest neighbour
linear data compression	Significance of statistical distribution of variables in the dataset is the measure	Principal component analysis

A. Hierarchical Cluster Analysis (HCA)

HCA techniques attempt to separate data into specific groups, based on a similarity measure. Initially each data point represents its own cluster, then the threshold for the decision when to declare two or more objects to be members of the same cluster is lowered incrementally [6] [12]. This strategy will produce hierarchical representations in which the clusters at each level of the hierarchy are created by merging clusters at the next lower level. At the lowest level, each cluster contains a single observation. At the highest level there is only one cluster containing all of the data [13], and that can be graphically displayed in a *Dendrogram Diagram*. This diagram will represent a Binary tree that reflect the *agglomerative* and the *divisive* approaches, which are used to show the similarity and dissimilarity between groups [6] [13]. A visual illustration of this method is presented in Figure 3.

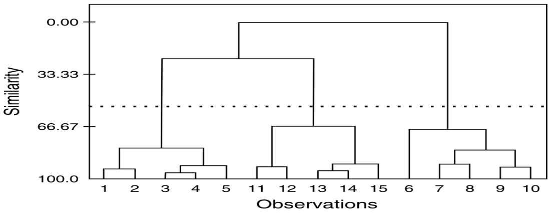

Fig. 3. Dendogram illustrating HCA data clustering [6]

B. K-means

K-means is considered as a *partitional clustering* algorithm, which attempts to obtain partitions in the data instead of a clustering structure, but the common problem with this approach is the need to specify the desired number of clusters before computation begins. The partitional techniques usually produce clusters by minimizing a criterion function defined either locally or globally. Cluster prototypes are not usually known beforehand, and are calculated by the algorithm simultaneously with the partitioning of the data [6]. The aim of the K-means algorithm is to divide M points in N dimensions into K clusters so that the within-cluster sum of squares is minimized. It is not practical to require that the solution has minimal sum of squares against all partitions, except when M, N are small and K = 2. We seek instead for a ”local” optima, a solution such that no movement of a point from one cluster to another will reduce the within-cluster sum of squares [12]. The K-means algorithm alternates the two steps [13]:

- for each center we identify the subset of training points (its cluster) that is closer to it than any other center
- the means of each feature for the data points in each cluster are computed, and this mean vector becomes the new center for that cluster

Several variants of the k-means algorithm have been reported and many attempt to estimate a good initial partition so that the algorithm is more likely to converge at the global minimum value. Another variation of the algorithm is to split and merge the resulting clusters. This type of variant allows an optimal partition starting from almost any arbitrary initial partition, providing well-determined threshold values are specified [6].

C. K nearest neighbour (KNN)

KNN is an algorithm that combines regression and classification, and is sometimes called memory-based learning, as they learn from set of events captured in the data [12]. Training the model simply consists of storing the training data by using *Euclidean* to compute similarities. The distance between an unclassified pattern vector and every sample from the training set is calculated, and it is assigned to the class of the k neighbours with the closest distances after taking the top N products based on the rating vector [6] [12]. Choosing K (classes) relies on the way that we use the algorithm. The most popular prototype is 1-NN [6], since it gives binary classes and the boundary near to be linear [13], but the variance is so high, so by training you can reach to optimal K.

D. Principal component analysis (PCA)

PCA is an unsupervised multivariate procedure and is a well known linear data compression and feature extraction technique, mainly used as a filter method to remove features without knowing the effect on the classification algorithm [13]. The scores produced may be plotted in two or three dimensions to inspect the data. PCA derives new, uncorrelated variables that are linear combinations of the original variable set ordered by reducing variability. It is mainly used to reduce the dimensionality of a data set while retaining as much information as possible by eliminating the lowest ranking variables. PCA is a simple and fast method but remains a linear approach, therefore any non-linear correlation between variables will not be retained [6].

E. Applications

According to Scott, James and Ali [6], these techniques were helpful when they implemented their data analyzing research on *E-nose system*, since every one of them has a feature that supports them on their duties:

- *HCA*: Full structure for all clusters that generated, with complete linkage.
- *PCA*: Utilizing PCA supported them to separate between groups and illustrated the interaction between them.
- *KNN*: Feature extraction method was used based on a curve fitting model of multiple gases with KNN as the classifier.
- *K-means*: Used as an explanatory method and it helped to detect the different behavior for some sensors based on their circumstances.

VI. Conclusions

Through the review of previous work done within this subject area by Wicker et al [1], [2] and similar fields [3], [4], [5], we have found many helpful techniques that may be useful to implement in our research. It was found in multiple papers [2], [1], [3], [4] that utilizing tree algorithms would be an easy way to perform variable screening or feature selection, and this would also help us understand the data and easily find relationships between variables. Takahashi and Sugimoto [3], [4], employed neural networks and support vector machines to help predict emotional states and Frantzidis et al [5] stress the importance of combining various different methodologies to arrive at the best outcome. These are all helpful methodologies which can be explored once we have decided what we wish to predict or find within our research.

Techniques from the machine learning [7] domain incorporates methods from the statistic discipline as well as concepts from artificial intelligence. *Supervised learning* [7] and *Unsupervised learning* [8] can be useful in analyzing the data in this study, depending on the objective that decides how the data is partitioned and whether a target response variable is involved. Supervised learning are suitable for predictive models, and approaches involving multiple techniques such as *linear regression* and *logistic regression*, *k-nearest neighbour*, *random forests*, *decision trees*, *support vector machines* and *neural networks* are commonly used in parallel and evaluated at the end to determine the model with the best predictions. Unsupervised learning on the other hand are commonly used with pattern mining objectives, with techniques such as *Factor Analysis* ,*Independent Components Analysis (ICA)* and *Principal Component Analysis (PCA)* that explores underlying patterns and potential hidden variables with each of their own advantages.

Derivative dynamic time warping technique from Keogh and Pazzani [11] can be useful in effectively aligning the various gas traces from difference movie sessions.

Clustering and Pattern recognition Techniques from Scott, James and Ali [6] can be useful to define patterns and show the interaction between them. These techniques have different features, they can be selected and used within the implementation process, visualization, reduce dimensioning and identifying the most effective subset of the original features.

References

[1] J. Williams, C. Stönner, J. Wicker, N. Krauter, B. Derstroff, E. Bourtsoukidis, T. Klpfel, and S. Kramer, "Cinema audiences reproducibly vary the chemical composition of air during films, by broadcasting scene specific emissions on breath," *Scientific Reports*, vol. 6, no. 25464, 2016.

[2] J. Wicker, N. Krauter, B. Derstorff, C. Stönner, E. Bourtsoukidis, T. Klüpfel, J. Williams, and S. Kramer, "Cinema data mining: The smell of fear," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, (New York, NY, USA), pp. 1295–1304, ACM, 2015.

[3] K. Takahashi and I. Sugimoto, "Feasibility of emotion recognition from breath gas information," in *2008 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 625–630, July 2008.

[4] K. Takahashi and I. Sugimoto, *Remarks on Emotion Recognition Using Breath Gas Sensing System*, pp. 49–62. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[5] C. A. Frantzidis, C. Bratsas, M. A. Klados, E. Konstantinidis, C. D. Lithari, A. B. Vivas, C. L. Papadelis, E. Kaldoudi, C. Pappas, and P. D. Bamidis, “On the classification of emotional biosignals evoked while viewing affective pictures: An integrated data-mining-based approach for healthcare applications,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, pp. 309–318, March 2010.

[6] S. M. Scott, D. James, and Z. Ali, “Data analysis for electronic nose systems,” *Microchimica Acta*, vol. 156, pp. 183–207, Dec 2006.

[7] A. Géron, *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems.* " O’Reilly Media, Inc.", 2017.

[8] R. Gentleman and V. J. Carey, *Unsupervised Machine Learning*, pp. 137–157. New York, NY: Springer New York, 2008.

[9] G. Bontempi, S. B. Taieb, and Y.-A. Le Borgne, “Machine learning strategies for time series forecasting,” in *European Business Intelligence Summer School*, pp. 62–77, Springer, 2012.

[10] D. J. Berndt and J. Clifford, “Using dynamic time warping to find patterns in time series.,” in *KDD workshop*, vol. 10, pp. 359–370, Seattle, WA, 1994.

[11] E. J. Keogh and M. J. Pazzani, *Derivative Dynamic Time Warping*, pp. 1–11.

[12] K. Ramasubramanian and A. Singh, *Machine Learning Theory and Practices*, pp. 219–424. Berkeley, CA: Apress, 2017.

[13] T. Hastie, R. Tibshirani, and J. Friedman, *Unsupervised Learning*, pp. 485–585. New York, NY: Springer New York, 2009.