Expert Systems With Applications 207 (2022) 117881

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Evolving Gaussian on-line clustering in social network analysis

Igor Škrjanc[a], Goran Andonovski[a,*], José Antonio Iglesias[b], María Paz Sesmero[b], Araceli Sanchis[b]

[a] *Faculty of electrical engineering, University of Ljubljana, Ljubljana, Slovenia*
[b] *Department of Computer Science and Engineering, Universidad Carlos III de Madrid, Madrid, Spain*

ARTICLE INFO

Keywords:
Evolving clustering
Twitter data analysis
Online method
Gaussian probability

ABSTRACT

In this paper, we present an evolving data-based approach to automatically cluster Twitter users according to their behavior. The clustering method is based on the Gaussian probability density distribution combined with a Takagi–Sugeno fuzzy consequent part of order zero (eGauss0). This means that this method can be used as a classifier that is actually a mapping from the feature space to the class label space. The eGauss method is very flexible, is computed recursively, and the most important thing is that it starts learning "from scratch". The structure adapts to the new data using adding and merging mechanisms. The most important feature of the evolving method is that it can process data from thousands of Twitter profiles in real time, which can be characterized as a Big Data problem. The final clusters yield classes of Twitter profiles, which are represented as different activity levels of each profile. In this way, we could classify each member as *ordinary*, *very active*, *influential* and *unusual* user. The proposed method was also tested on the Iris and Breast Cancer Wisconsin datasets and compared with other methods. In both cases, the proposed method achieves high classification rates and shows competitive results.

1. Introduction

Since its inception, the social network Twitter has become one of the most popular networks where users discuss various topics such as politics, music, commercial products, sports, etc. Users express their thoughts through short text messages called *tweets*. Since November 2017, the company has doubled the number of characters available for a tweet from 140 to 280, allowing users to elaborate on their thoughts. In addition, users can use a series of related tweets as one Tweeter thread to express their thoughts in more detail. According to quarterly reports (Q1 2021) (Twitter, 2021b), there are 199 million daily active users and 500 million tweets sent per day. It is also interesting to note that 92% of US tweets come from 10% of Twitter users (Pew Research Center, 2020). Undoubtedly, Twitter is one of the most popular and influential networks where different opinions and beliefs are juxtaposed. Of course, the network is made up of the users and if we understand the users and their tweets, we can understand the network.

With the growth and increasing popularity of the Twitter network, academics and researchers have begun to collect Twitter data to study various topics. For this reason, Twitter has provided APIs (Application Programming Interfaces) that allow data collection where users can access public tweets. Moreover, in 2020, Twitter announced "Academic Research product track" platform (Twitter, 2021a) that provides more accurate, complete, and unbiased historical and real-time data. The authors of Karami et al. (2020) found 38 Twitter-related topics in around 18,850 manuscripts (unique abstracts written in English) published in three different databases (Web of Science, Google Scholar, and IEEE Explore) between 2006 and 2019.

Due to the availability of Twitter data and the diversity of the data, there are several data analysis problems in the literature. In the following, we will focus on works that deal with methods such as Machine Learning, Clustering, Text Mining and Intelligent Analysis using Twitter data. One of the most widely considered topics is *Sentiment Analysis* which studies whether the sentiment of a tweet is positive, negative or neutral (Bouazizi & Ohtsuki, 2019; Bravo-Marquez et al., 2017; Giachanou & Crestani, 2016) but not limited as shown in Bouazizi and Ohtsuki (2016) where the authors classify tweets into seven classes such as happiness, sadness, anger, love, hate, sarcasm and neutral. The author in Onan (2021) presented a Deep Learning based approach, which outperforms conventional Deep Learning methods, for sentiment analysis of Twitter product reviews. In Rehioui and Idrissi (2020), the authors use a combination of two clustering approaches, K-means and

* Corresponding author.
E-mail addresses: igor.skrjanc@fe.uni-lj.si (I. Škrjanc), goran.andonovski@fe.uni-lj.si (G. Andonovski), jiglesia@inf.uc3m.es (J.A. Iglesias), msesmero@inf.uc3m.es (M.P. Sesmero), masm@inf.uc3m.es (A. Sanchis).

https://doi.org/10.1016/j.eswa.2022.117881
Received 25 October 2021; Received in revised form 13 April 2022; Accepted 13 June 2022
Available online 22 June 2022

DENCLUE (Density based clustering), for analyzing Twitter sentiment. A similar work was presented in Alanezi and Hewahi (2020) where the authors focus on the analysis of public sentiment during the pandemic COVID-19 using K-means clustering. The authors in Onan (2019) and Onan and Tocoglu (2021) presented a Deep Learning approach for *sarcasm* identification on social media data. In Tutaysalgir et al. (2019), different clustering algorithms were used for predicting personality traits of users, e.g., K-means, K-means++, and Agglomerative clustering. In Cheong and Lee (2010), the authors propose a new clustering approach based on machine learning to identify demographics, habits, and sentiments of tweets related to popular topics discussed on Twitter. The authors in Liu et al. (2015) went one step further and published a topic-adaptive classification method for sentiment analysis. They note that a static classifier may perform better on one topic and worse on another. Therefore, they proposed an adaptive classification method that depends on the topic. There are also (convolutional) neural network applications for Twitter sentiment analysis (Jianqiang et al., 2018; Vateekul & Koomsubha, 2016; Zhang et al., 2017).

Next, the category of *Disaster Analysis* investigates the patterns in Twitter activity to detect, analyze, or predict user behavior during certain harmful events. For example, in Middleton et al. (2014), the authors use statistical analysis for data from multiple sources besides Twitter data to create a real-time crisis map. Another study, Li et al. (2014), examines the pattern of risk communication according to retweeting activity during the Fukushima nuclear radiation disaster. In Sakaki et al. (2013), a real-time earthquake notification system was developed. The event detection system is based on a support vector machine (SVM) classifier. Similarly, a real-time monitoring system based on SVM for traffic event detection was presented in D'Andrea et al. (2015). In Doulamis et al. (2016) two event detection scenarios (short-therm and long-therm) through graph partitioning were presented. Another work was presented in Liu et al. (2018), where the authors presented a new approach to study crisis response strategies in the social media domain. They investigated how the government uses Twitter as a medium for communicating with the public across various crisis stages of Hurricane Harvey.

Twitter as a social platform is also considered as an important source of information in text mining. In Godfrey et al. (2014), the authors analyzed tweets before the start of the World Cup. They used four different algorithms to remove noise from the tweets and then clustered (k-means and Alternating Constrained Least Square - ACLS) the data into nine distinguish topics. In Lo et al. (2015), the tweets were analyzed from a commercial perspective, where the goal was to identify the potential social audience already interested in a particular topic. The authors used different methods, such as fuzzy keyword matching, statistical topic modeling, and support vector machine to analyze an account holder and its followers. Kim et al. in Kim et al. (2015) proposed a system to identify hot keywords in a particular location for the purpose of spatiotemporal trend detection. They also linked the identified trending keywords to Google and Wikipedia services for semantic explanation. Using the GPS metadata of the tweets, they identify the popularity distribution of these keywords.

In this paper, we address the problem of clustering Twitter users' activities on specific topics (e.g., politics, music, sports, etc.) using an evolving Gaussian clustering method (eGauss). In this way, we are not interested in the content of tweets as an aspect of sentiment analysis or text mining. This could be a next step, which is discussed in the Future Work section. As far as we know, there is rarely any work that deals with clustering users according to their activity. The basis of evolving methods is that the model changes its structure, not just the parameters of the local models/clusters. The usability of such models is evidenced by surveys (Leite et al., 2020; Škrjanc et al., 2019), books (Angelov, 2002, 2012; Angelov et al., 2010; Lughofer, 2011), chapter (Lughofer, 2015) and even a specific journal "Evolving Systems" by Springer. Moreover, the evolving systems are successfully used as a tool for solving different problems, such us clustering (Angelov & Zhou, 2008; Costa et al., 2016; Crespo & Weber, 2005; Dovžan & Škrjanc, 2011; Dutta Baruah & Angelov, 2012), model identification (Andonovski, Mušič et al., 2016; Angelov & Filev, 2004; Škrjanc, 2015), control (Andonovski, Angelov et al., 2016; Angelov et al., 2015; Leite et al., 2015; Lin et al., 2001), monitoring (Dovžan et al., 2016), etc. The structure of the models is very flexible and can be based on a fuzzy theory (rule-based, tree-based, granular, etc.), neural networks, or a combination of both as a neuro-fuzzy model.

The evolving Gaussian clustering method is based on a zero-order (Takagi–Sugeno) fuzzy model, which means that the final clusters are used as classifier, which maps the feature space directly to the class label space. Depending on the predefined specific topic (e.g., politics, music, etc.) and data features (e.g., number of posts, number of favorited tweets, number of followers, etc.), the structure of the clusters starts from scratch and evolves according to the data received. This means that each time the algorithm processes new data, new clusters are added or merged, depending on the evolving mechanisms implemented. We must emphasize that it is a one-pass algorithm, which means that this algorithm could be implemented in real time for real data. Moreover, the algorithm is very flexible and can handle data from thousands of Twitter profiles. We tested our approach on Twitter data and successfully clustered into four different types of users, e.g. *ordinary*, *very active*, *influential*, and *unusual* (outliers). Furthermore, we tested the effectiveness of the evolving clustering method on two well-known datasets, Iris and Breast Cancer Wisconsin, and compared it with established machine learning methods, such as Bagging Classifier (Breiman, 1999), Random Forest Classifier (Breiman, 2001), Gradient Boosting Classifier (Friedman, 2001), etc. For both datasets, the proposed method shows competitive results.

The main contribution of this work can be summarized as follows:

- An online (single-pass) evolving Gaussian clustering method based on a zero-order fuzzy model.
- Online data clustering and classification of Twitter users' activities.
- Competitive results compared to established machine learning methods.

The paper is structured as follows: At the beginning, Section 2 presents the evolving clustering algorithm eGauss+ and furthermore the evolving mechanisms for adding new clusters, adapting clusters with new samples and merging clusters. Section 3 presents the experimental results in two parts. First, the supervised classification results on two well-known datasets, namely Iris and Breast Cancer Wisconsin, are presented. Second, the results of eGauss+ clustering on real Twitter users are presented, where the users were classified into four typical classes. Finally, the conclusion and ideas for future work are given.

2. Methodology of evolving Gaussian clustering from data streams

In this section, we present an evolving method that processes data streams using supervised and unsupervised learning techniques in a single pass and is used for online clustering. The stepwise processed data updates the parameters of the clusters and adds new clusters, i.e., it evolves the structure of the model. Next, the small and close clusters are merged to obtain a more compact form of clusters. Since the volume and size of the clusters are not known in advance, this method initially creates small clusters which are from time to time merged into larger clusters (Škrjanc, 2020). Merging mechanisms are an infallible part of many evolving methods, as they allow for a more compact representation of the data and make the final structure of the model easier to understand and interpret.

2.1. Evolving classification: eGaussClass family

First, we will introduce the structure of the evolving Gaussian clustering method (eGauss+). As we mentioned before, in this work we use a fuzzy structure of zero-order Takagi–Sugeno system. The eGauss+ is based on an evolving method presented in Angelov and Zhou (2008), which offers several possible architectures, such as *eClass0* (class labels as outputs) and *eClass1* (performs regression over features). Both classes have the same antecedent part of the fuzzy rule-based system and differ only in a consequent part. In this approach, we use *eClass0*, which has a consequent of zero-order Takagi–Sugeno, so a fuzzy rule in our case has the following structure:

$$R_i \ : \ if \ x(k) \sim P_i \ \ then \ \ Cat = Cat_i$$

where i represents the number of rule; $x(k)$ is the normalized sample of the twitter profile to classify, and the P_i stores the properties of the prototype i. $Cat_i \in$ {set of different categories of profiles}. The prototype P_i is in our case defined by the hyper-ellipsoid with the center μ_i and the covariance matrix Σ_i.

When dealing with fuzzy models and fuzzy rules, it is important to mention their human interpretability. Moreover, with the evolving nature of the method, we can analyze how these rules change over time (how the structure evolves) and we can identify the main properties of each category.

In order to classify the current data into one of the previously identified categories/clusters, we can calculate the distance between the new profile and the different prototypes describing the clusters. Since a category is represented by one or more prototypes, all the data are compared to the existing prototypes and the smallest distance determines the greatest similarity. This can be summarized with the following equation:

$$\begin{aligned} &Class(x(k)) = Class(P^*); \\ &P^* = max_{i=1}^{N_P}(dist(P_i, x(k))) \end{aligned} \tag{1}$$

where $x(k)$ represent the sample to classify, N_P determines the number of existing prototypes (number of rules/clusters), P_i represents the i_{th} prototype, and $dist$ represents the Mahalanobis distance, i.e. calculated as

$$P^* = max_{i=1}^{N_P} e^{(x(k)-\mu)^T \Sigma_i^{-1}(x(k)-\mu)} \tag{2}$$

If the value $P*$ is less than the predefined value, i.e. very often 0.04 is defined as the threshold, than the sample is categorized as outlier. The cluster defined as prototype is the cluster obtained after merging procedure.

For a more general representation, we consider a m-dimensional feature vector, i.e., the problem with m measured variables, where the raw data sample is written as $z = \left[z_1 \ z_2 \cdots z_m\right]^T$. Normally, the raw data is first processed by data cleaning, removing erroneous data, standardization/normalization or mean removal and variance scaling. In this approach, we only use a normalization technique where the data is scaled by the maximum expected value max_{z_i}, i.e., $x_i(k) = z_i(k)/max_{z_i}$, where $x_i(k)$ denotes the normalized value of the feature variable z_i and $i = 1, \ldots, m$. Generally, a cluster with n associated samples can be described with three properties: μ - mean or center, Σ - covariance matrix, and V - volume. Each of these properties is calculated as follows:

$$\mu = \frac{1}{n}\sum_{k=1}^{n} x(k) \ , \tag{3}$$

$$\Sigma = \frac{1}{n-1}(X - EM)^T(X - EM) \ , \tag{4}$$

and

$$V = \frac{2\pi^{m/2}}{m\Gamma(m/2)} \ \Pi_{i=1}^{m}\lambda_i. \tag{5}$$

In Eq. (4), the matrix X is defined as $X^T = [x(1), \ldots, x(n)]$ with dimension $n \times m$; M stands for diagonal matrix $M = diag(\mu_1, \ldots, \mu_m)$ and E stands for the $n \times m$ matrix with all elements equal to 1. In the last property (Volume) defined by Eq. (5) the variable Γ stands for the gamma function and λ_i for the ith eigenvalue of the covariance matrix Σ.

2.2. Calculation of membership values

Having defined the cluster structure in the previous subsection, the next step is to compute the membership values of the newly received data $x(k)$ to each currently existing cluster $i = 1, \ldots, c$, where c is the number of clusters. We refer to the "currently existing" number of clusters because the number changes due to the evolving mechanism, which will be explained in the next two subsections. The membership function is Gaussian based (also called typicality) and is calculated as follows:

$$\gamma_i(k) = e^{-d_i^2(k)} \ , \ \ i = 1, \ldots, c, \tag{6}$$

where $d_i^2(k)$ is a general form of squared distance measure between the current data $x(k)$ and the ith cluster. Depending on the number of data samples associated to the cluster n and a predefined threshold N_{max}, we use two different distance measures, Euclidean and Mahalanobis, as follows:

$$d_i^2(k) = \begin{cases} (x(k)-\mu_i)^T(x(k)-\mu_i), & n \le N_{max} \\ (x(k)-\mu_i)^T \Sigma^{-1}(x(k)-\mu_i), & n > N_{max} \end{cases} \tag{7}$$

We must note that the use of the Mahalanobis distance is tied to the non-singularity of the covariance matrix. When the Mahalanobis distance is used as a metric, the detected clusters have a rotated hyperellipsoidal shape. This means that the clusters become much more flexible and have a much higher approximation power (Škrjanc, 2020).

2.3. Evolving mechanisms - adding and merging clusters

Here we will explain the evolving mechanisms of the proposed method eGauss+ (Škrjanc, 2020). First, the adding mechanism leads to the parameter adaptation of the selected cluster or the initialization of the new cluster. After computing the membership function of the current data $x(k)$ to all existing clusters using Eq. (6), the cluster with the maximum membership value is selected. We denote this cluster by index j, where $1 \le j \le c$. Using the following condition, we can decide whether to add a new cluster or adapt an existing one:

$$\text{Add or Adapt} = \begin{cases} \text{Add new cluster}, & \text{if } \gamma_j(k) \le \Gamma_{max} \\ \text{Adapt } j\text{th cluster}, & otherwise, \end{cases}$$

where Γ_{max} is predefined threshold and $\gamma_j(k)$ the maximum membership value of jth cluster.

When we **add** a new cluster, it means that we need to increase the number of existing clusters $c \leftarrow c + 1$ and initialize the parameters of the new cluster as follows:

$$\begin{aligned} &n_c = 1, \\ &\mu_c = x(k), \\ &\Sigma_c = \mathbf{0}. \end{aligned} \tag{8}$$

The same initialization procedure is required when we start the clustering procedure, i.e. $c = 1$.

When the condition for adaptation of the jth cluster is fulfilled ($\gamma_j(k) > \Gamma_{max}$) then the current data sample $x(k)$ is associated to that cluster. Next, we need to update the parameters/properties of that cluster in a recursive way as follows:

$$e_j(k) = x(k) - \mu_j^{n_j}, \tag{9}$$

$$\mu_j^{n_j+1} = \mu_j^{n_j} + \frac{1}{n_j+1}e_j(k), \tag{10}$$

$$S_j^{n_j+1} = S_j^{n_j} + e_j(k)\left(x(k) - \mu_j^{n_j+1}\right)^T, \tag{11}$$

$$\Sigma_j^{n_j+1} = \frac{1}{n_j} S_j^{n_j+1}, \tag{12}$$

$$n_j = n_j + 1, \tag{13}$$

where n_j is the number of samples associated to the jth cluster. We have to note that Eq. (11) is used as unnormalized covariance matrix for more compact presentation and calculation of covariance matrix in Eq. (12).

At the end of the data stream, as is the case in this paper (or from time to time), the merging algorithm described in Alg. 1 is triggered. First, we need to compute the possible candidates (pair of clusters p and q) for merging. This is done using the similarity measure (or overlapping measure), which is defined as follows:

$$\kappa_{pq} = \frac{V_{pq}}{V_p + V_q} \tag{14}$$

where V_p and V_q are the volumes of pth and qth cluster, respectively; the volume of the potentially merged clusters is denoted as V_{pq}. We have to note that V_{pq} is calculated using Eq. (5) taking in consideration equations from (16) to (20). The Eq. (14) is calculated for all possible combination of the existing clusters and then the most similar (overlapping) clusters with the minimal value of κ_{pq} are chosen as the best candidate:

$$(p^*, q^*) = arg \min_{p,q} \kappa_{pq}, \quad p, q \in \{1, \dots, c\} \wedge p \neq q, \tag{15}$$

and the minimal value of overlapping ratio is denoted as κ_{p^*,q^*}. The cluster candidates for merging (p^*, q^*) are merged if the value κ_{p^*,q^*} is lower than predefined threshold k_{merge}. Next, using the equation from (16) to (20) the new properties of the merged cluster are calculated. For more clear representation the subscripts of the merging candidate clusters (p^*, q^*) are replaced with (p, q). First the number of samples are joined:

$$n_{pq} = n_p + n_q \ , \tag{16}$$

and the mean value of the merged cluster is calculated as:

$$\mu_{pq} = \frac{n_p \mu_p + n_q \mu_q}{n_{pq}} \ , \tag{17}$$

The joint covariance matrix is calculated as follows:

$$\Sigma_{pq} = \frac{1}{n_{pq} - 1}\left(X_p^T X_p + X_q^T X_q - M_{pq}^T E_{pq}^T E_{pq} M_{pq}\right) \ . \tag{18}$$

where

$$X_p^T X_p = (n_p - 1)\Sigma_p + M_p^T E_p^T E_p M_p \ , \tag{19}$$

and

$$X_q^T X_q = (n_q - 1)\Sigma_q + M_q^T E_q^T E_q M_q \ . \tag{20}$$

At the end we need to decrease the number of existing clusters $c \leftarrow c-1$.

The full evolving Gaussian clustering algorithm is shown in Alg. 2, where the merging algorithm is performed at the end of the dataset. This could also be performed between the clustering phase.

3. Experimental results

In this section, we will present the effectiveness and potential of the eGauss+ clustering algorithm. First, we show the results on two well-known datasets, Iris and Breast Cancer Wisconsin, and then the results on Twitter data are presented.

Algorithm 1 Merging clusters.

```
Choice of $\kappa_{join}$
Initialization: $merge = 1$
while $merge == 1$
    Computation of $\hat{V}_i = \Pi_{j=1}^{m} \lambda_j^i,\ i \in \mathcal{A}$,
    where $\lambda_j^i$ stands for the $j$th eigenvalue of $\Sigma_i$
    For every $p, q \in \{1, \dots, c\} \wedge p \neq q$, compute
        equations from (16) to (20)
    Find merging candidates (15)
    if $\kappa_{p^*q^*} < \kappa_{join}$
        Merge clusters $p^*$ and $q^*$ in ($\Sigma_{new}$, $\mu_{new}$, $n_{new}$)
        $c \leftarrow c - 1$
        $merge = 1$
    else
        $merge = 0$
    end
end
```

Algorithm 2 Algorithm of Evolving Gaussian Clustering.

```
Choice of $\Gamma_{max}$ or $N_r$, $N_{max}$
Initialization:
$c \leftarrow 1$, $\mu_c \leftarrow x(1)$, $k \leftarrow 1$
repeat $k \leftarrow k + 1$
    for i = 1 : c
        Distance computation - Eq. (7)
        Membership function computation - Eq. (6)
    end
    Choice of maximal typicality $j = arg \max_i \gamma_i(k)$
    if $\gamma_j(k) \geq \Gamma_{max}$
        Update the $j$-th cluster with new data sample:
            Eq. from (9) – (13)
    else
        Add and initialize new cluster ($c \leftarrow c + 1$):
            Eq. (8)
    end
until $k > N$
Merging clusters
```

3.1. Examples of classification

As mentioned earlier, in this subsection we present the comparative results of eGauss+ on two different datasets. We also compare the results with some established machine learning methods. The clusters are trained in a supervised manner. Part of the data is used for learning purposes and another part is used for validation. In the first example (Iris dataset R. A. Fisher, Sc.D., 1936), we use perhaps the best known database in the pattern recognition literature to show the properties of the eGauss+ algorithm in classification. This dataset contains three classes with 50 instances of each class. The measured attributes for each data sample are:

- Sepal length in cm
- Sepal width in cm
- Petal length in cm
- Petal width in cm

The predicted attribute of iris plant are from three different classes: Iris Setosa, Iris Versicolor and Iris Virginica, i.e. Cat_{set} = {Iris Setosa, Iris Versicolor, Iris Virginica}.

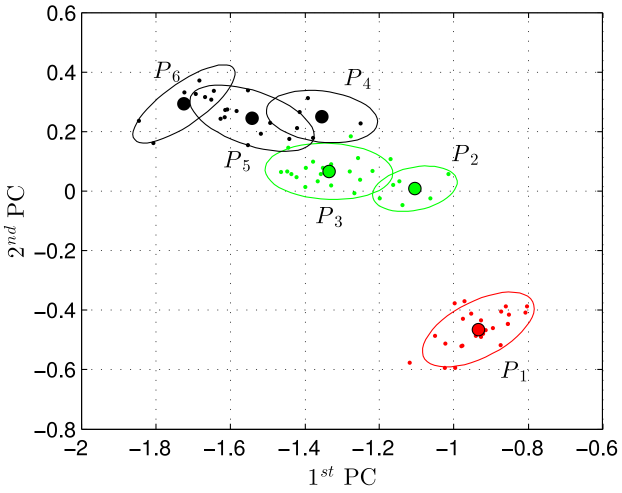

Fig. 1. The first three features of iris data and final clusters for all classes, with prototypes in the case of two principal components clustering, 2σ contour, and $\kappa_{join} = 2.0$, $\Gamma_{min} = 0.96$.

Table 1
The results of classification on iris data set.

Method	Average accuracy
Bagging Classifier (Breiman, 1999)	96.08
Gradient Boosting Classifier (Friedman, 2001)	96.08
Random Forest Classifier (Breiman, 2001)	96.08
Ridge Classifier (Tikhonov, 1943)	88.24
SVC (Hearst et al., 1998)	**98.04**
eGauss+ with PCA	**95.94**

Classification was performed using 100 samples for learning the cluster structure and the remaining 50 samples for validation the classifier. The convolution of the data was repeated 20 times and the average was calculated. In the first experiment, the data were preprocessed using principal component analysis (PCA), considering the first two principal scores. The scores of the iris data set and the final clusters with centers in the case of two principal component clusters, 2σ contour and $\kappa_{join} = 2.0$, $\Gamma_{min} = 0.96$ are shown in Fig. 1. As can be seen in the same Fig. 1, the first class is represented by one cluster, the second class by two clusters and the third class by three clusters. The average accuracy of the classification in this case is 95.94%. The final set of three rules which define the classification model for Iris data is the following:

$$R_1 : if\ x(k) \sim P_1\ then\ Cat = Cat_1$$

$$R_2 : if\ x(k) \sim (P_2\ or\ P_3)\ then\ Cat = Cat_2$$

$$R_3 : if\ x(k) \sim (P_4\ or\ P_5\ or\ P_6)\ then\ Cat = Cat_3$$

Due to the popularity of the Iris dataset we found in literature several methods for classification, e.g. Bagging Classifier (Breiman, 1999) with average accuracy is 96,08 %, Gradient Boosting Classifier (Friedman, 2001) results in 96,08 %, Random Forest Classifier (Breiman, 2001) in 96,08 %, Ridge Classifier (Tikhonov, 1943) has accuracy 88,24 % and SVC (Hearst et al., 1998) results in 98,04 %. The results are summarized in Table 1.

The second classification example was performed using also the well-known dataset Breast Cancer Wisconsin (Diagnostic). This dataset was created by Dr. William H. Wolberg (Wolberg & Mangasarian, 1990), physician at the University Of Wisconsin Hospital in Madison, Wisconsin, USA. Fluid samples from patients with solid breast masses and an easy-to-use graphical computer program called Xcyt, which is capable of performing cytologic feature analysis based on a digital scan, were used to create the dataset. The dataset includes a total of 699 samples which may belong to either benign or malignant stage. The measured nine features are in the interval of 1 to 10 and all the features are listed as follows:

Table 2
The results of classification the Breast Cancer Wisconsin (Diagnostic) dataset.

Method	Average accuracy
KNN model	95.88
NN model	96.08
NNET with PCA	**97.65**
SVM with radial kernel	92.94
Naive Bayes	91.18
eGauss+	**94.56**
eGauss+ with PCA	**95.99**

- Clump Thickness
- Uniformity of Cell Size
- Uniformity of Cell Shape
- Marginal Adhesion
- Single Epithelial Cell Size
- Bare Nuclei
- Bland Chromatin
- Normal Nucleoli
- Mitoses
- **Class:** (benign or malignant cancer)

The overall goal of this database is to correctly diagnose whether breast cancer is benign or malignant.

As in the previous example, we learned the eGauss+ evolving clustering algorithm in a supervised manner. One part of the data was used for learning and training purposes, while the second part was used for validation purposes. In this case, we used only the eGauss+ algorithm (without PCA) and detected two clusters (two classes) with 8 and 13 sub-clusters, respectively. The parameters chosen for this experiment were $\kappa_{join} = 1.20$ and $\Gamma_{min} = 0.75$. The average accuracy of classification in this case is 94.56%. Next, we combined eGauss+ with the PCA preprocessing method using the first three components. In this case, the accuracy increased to 95.99% but we detected two clusters with 21 and 49 sub-clusters, respectively.

For comparison with the eClass+ method, we found classification results in the literature that refer to the dataset Breast Cancer Wisconsin (Diagnostic). The comparison results are summarized in the Table 2. We can conclude that the results of eClass+ (with and without PCA) can compete with those of other methods. Also, it should be noted that eClass+ is a one-pass method, i.e., we process the data online. Moreover, the representation and understanding of the fuzzy rules (and sub-classes) is much more intuitive than the structure of other methods.

3.2. Twitter data clustering

In this subsection, we first introduce the Twitter data extraction approach presented in Iglesias et al. (2017) and then we present the results of the eGauss+ clustering algorithm.

As shown in Fig. 2, the Twitter data is extracted using the Twitter API according to the two inputs: *search word* (Twitter topic, i.e., politics, music, sports, etc.) and the *geographic areas* (list of global coordinates). Thus, the data is matched to the community that we want to analyze. The detailed procedure for extracting the data is presented in Iglesias et al. (2017). The list of tags extracted for each Twitter user is shown below:

- **Number of tweets posted by the user**,
- **Number of user favorite tweets**,
- **Number of followers of the user**,
- **Number of twitter users followed by user**,
- **Account creation date**,

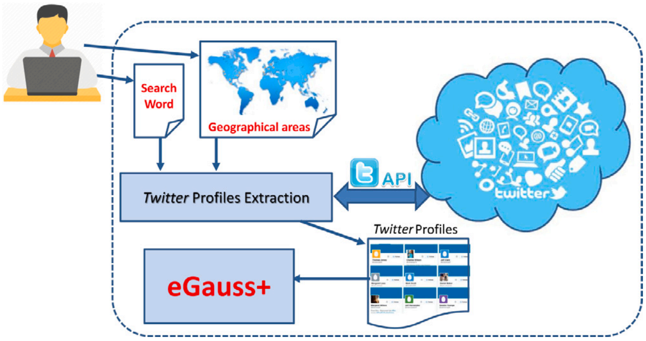

Fig. 2. General structure of the approach for Twitter data acquisition and eGauss+ clustering.

- Color of the background of the user profile,
- Color of the links,
- Color of the sidebar,
- Color of the text,
- Time Zone of the user,
- Languague chosen by the user,
- Number of public lists of the user,
- User URL,
- Boolean: the background has been modified,
- Boolean: the background is tiled,
- Boolean: the background image has been modified.

The tags in bold have been selected taking into account that the goal in this experiment is clustering user activity with eGauss+. In this sense, the other data are not considered since they do not affect the final result. Due to some limitations imposed by the free license of our Twitter API account, we collected data from 2866 Twitter users.

The analysis of Twitter profiles was performed in such a way as to identify the different clusters of users that appear more frequently and those that behave more reservedly and can be defined as outliers. To analyze this type of behavior, the following variables were used: the number of tweets posted by the user, defined as z_1, z_2 represents the number of user favorite tweets, z_3 is the number of followers the user has, z_4 is the number of Twitter users the user follows, and z_5 represents the account creation date. Using z_5 the age of each account can be determined and therefore all variables of a particular account are first normalized by its age and then normalized by the maximum expected value of each variable $\max z = [5000\ 1200\ 5100\ 550]$ to obtain the normalized variables x_i, $i = 1, \dots, 4$. The normalized variables form the data matrix X, which is subsequently used for the analysis.

For the analysis, the clustering parameters $\Gamma_{min} = 0.96$ and $\kappa_{join} = 1.1$ were set. The final number of clusters in this case is 8, with the following centers μ

$$\mu = \begin{bmatrix} 0.1738 & 0.0333 & 0.1372 & 0.2342 \\ 0.6866 & 0.0108 & 0.0203 & 0.1703 \\ 0.0266 & 0.0181 & 0.0016 & 0.0089 \\ 0.0720 & 0.2169 & 0.0041 & 0.0168 \\ 0.2399 & 0.0150 & 0.0103 & 0.0256 \\ 0.0407 & 0.6794 & 0.0044 & 0.0067 \\ 0.0391 & 0.0004 & 0.0751 & 0.5792 \\ 0.3644 & 0.4580 & 0.0088 & 0.0917 \end{bmatrix} \qquad (21)$$

with the following number of samples in each cluster $n = [9\ 20\ 2635\ 81\ 109\ 8\ 2\ 2]$ and the corresponding covariance matrix Σ_i, $i = 1, \dots, 8$.

Each pair of the center μ_i and the covariance matrix Σ_i define the prototype hyper-ellipsoid in general, P_i. The user are clustered into a set of categories $Cat_{set} = \{ordinary, \textit{very active}, influential, unusual\}$.

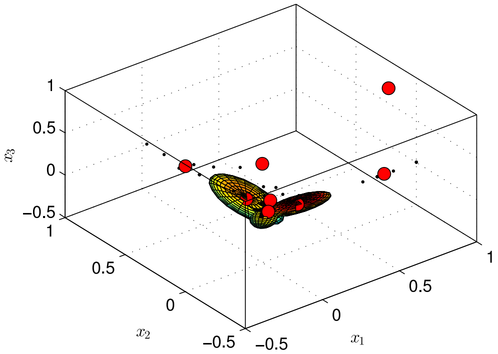

Fig. 3. The clusters and the means in the data space of normalized first three variables. The clusters are of dimension 3σ.

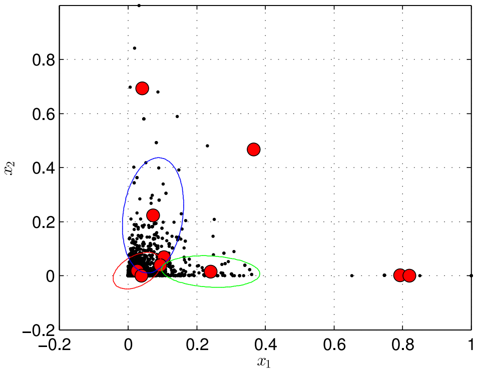

Fig. 4. The clusters and the means in the data space of normalized first two variables. The clusters are of dimension 3σ.

The final set of rules which define the classification model is the following

$$\begin{aligned} R_1 &: \ if\ x(k) \sim P_3 \ then\ \ Cat = Cat_1 \\ R_2 &: \ if\ x(k) \sim P_4 \ then\ \ Cat = Cat_2 \\ R_3 &: \ if\ x(k) \sim P_5 \ then\ \ Cat = Cat_3 \\ R_4 &: \ else\ \ Cat = Cat_4 \end{aligned}$$

This means that the center for the ordinary user defined by P_3 in the original variables is as follows: $\mu = [133\ 21\ 8\ 5]$, which means that the ordinary Twitter user in Cat_1 writes 133 tweets, notifies 21 favorite tweets, has 8 followers, and follows 5 other users. A very interesting observation is the case of the *very active* user, Cat_2, described by P_4. This is an extroverted user who has a normal number of 360 tweets, but a very high number of 260 favorite tweets and 21 followers, and who follows 9 other users. There is also an example of an *influential* user belonging to the class Cat_4 defined by P_2, who writes 3450 tweets, notifies 13 favorite tweets, has 103 followers, and follows 94 other users.

The clusters and the mean values in the data space of the normalized first three variables are shown in Fig. 3 in 3D space. The clusters have dimension 3σ, and Fig. 4 shows the centers of the clusters and clusters in 2D space for variables x_1 and x_2.

The obtained results, given in Table 3, show us an useful information about the user profiles, and by considering the different variables from the dataset, the different aspects can also be explained.

4. Conclusions and future work

In this paper, we present a flexible evolving clustering algorithm based on the Gaussian distribution. The structure of the model changes

Table 3
The results of classification the twitter users.

Category	Num. of tweets	Num. of favorite tweets	Num. of followers	Num. of followed users
Ordinary	133	21	8	5
Very active	360	260	21	9
Influential	3450	13	103	94

and evolves according to the newly obtained data. Therefore, the proposed method (eGauss+) is suitable for online clustering and classification. Using the two evolving mechanisms for adding new clusters and merging existing clusters, the structure (number of clusters) changes/evolves according to the current state of the data. Only two parameters need to be set for this method. We evaluated the capabilities of the proposed method with two well-known datasets, Iris and Breast Cancer Wisconsin, in a supervised manner and compared the results with established machine learning methods. For both datasets, eGauss+ shows truly competitive results, regardless of the fact that it is an online and one-pass method. Next, we tested the clustering capabilities of eGauss+ on real Twitter data extracted via the Twitter API protocol. Analyzing the data, we discovered four typical user classes, e.g. *ordinary*, *very active*, *influential*, and *unusual*. One of the main advantages of this method is its flexibility and scalability. Moreover, the current state of the clusters can be easily adapted to the new data. This paper also represents one of the first attempts (to our knowledge) to cluster Twitter users/accounts according to their activity in this social network.

CRediT authorship contribution statement

Igor Škrjanc: Conceptualization, Methodology, Software. **Goran Andonovski:** Literature review, Data curation, Writing – rewriting, Editing. **José Antonio Iglesias:** Twitter data preparation, Conceptualization. **María Paz Sesmero:** Twitter data preparation, Conceptualization. **Araceli Sanchis:** Validation, Reviewing, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work has been supported by the Program Chair of Excellence of Universidad Carlos III de Madrid and Bank of Santander and the Spanish Ministry of Economy, Industry and Competitiveness, projects RTI2018-096036-B-C22/AEI/10.13039/501100011033 (Government of Spain), PEAVAUTO-CM-UC3M (Government of Spain), PID2019-104793RB-C31 founded by MCIN/AEI/10.13039/501100 011033 and UE “NextGenerationEU/PRTR”, and EPUC3M17 (Region of Madrid’s Excellence Program).

References

Alanezi, M. A., & Hewahi, N. M. (2020). Tweets sentiment analysis during COVID-19 pandemic. In *2020 international conference on data analytics for business and industry: way towards A sustainable economy, ICDABI 2020, (September 2020)*. http://dx.doi.org/10.1109/ICDABI51230.2020.9325679.

Andonovski, G., Angelov, P., Blažič, S., & Škrjanc, I. (2016). A practical implementation of robust evolving cloud-based controller with normalized data space for heat-exchanger plant. *Applied Soft Computing*, *48*, 29–38.

Andonovski, G., Mušič, G., Blažič, S., & Škrjanc, I. (2016). On-line evolving cloud-based model identification for production control. *IFAC-PapersOnLine*, *49*(5), 79–84.

Angelov, P. (2002). *Evolving rule-based models: A tool for design of flexible adaptive systems* (first edn.). (p. 213). London: Springer-Verlag.

Angelov, P. (2012). *Autonomous learning systems: from data streams to knowledge in real-time* (first edn.). (p. 298). John Wiley & Sons, Ltd.

Angelov, P. P., & Filev, D. P. (2004). An approach to online identification of Takagi-Sugeno fuzzy models. *IEEE Transactions on Systems, Man, and Cybernetics.*, *34*(1), 484–498.

Angelov, P., Filev, D., & Kasabov, N. (2010). *Evolving intelligent systems: methodology and applications* (p. 444). New Jersey: John Wiley & Sons.

Angelov, P. P., Škrjanc, I., & Blažič, S. (2015). A robust evolving cloud-based controller. In J. Kacprzyk, & W. Pedrycz (Eds.), *Springer handbook of computational intelligence* (pp. 1435—-1449). Berlin, Heidelberg: Springer Berlin Heidelberg.

Angelov, P. P., & Zhou, X. (2008). Evolving fuzzy-rule-based classifiers from data streams. *IEEE Transactions on Fuzzy Systems*, *16*(6), 1462–1475.

Bouazizi, M., & Ohtsuki, T. (2016). Sentiment analysis: From binary to multi-class classification: A pattern-based approach for multi-class sentiment analysis in Twitter. In *2016 IEEE international conference on communications, ICC 2016*. IEEE, http://dx.doi.org/10.1109/ICC.2016.7511392.

Bouazizi, M., & Ohtsuki, T. (2019). Multi-class sentiment analysis on twitter: Classification performance and challenges. *Big Data Mining and Analytics*, *2*(3), 181–194. http://dx.doi.org/10.26599/BDMA.2019.9020002.

Bravo-Marquez, F., Frank, E., & Pfahringer, B. (2017). From opinion lexicons to sentiment classification of tweets and vice versa: A transfer learning approach. In *Proceedings - 2016 IEEE/WIC/ACM international conference on web intelligence, WI 2016* (pp. 145–152). IEEE, http://dx.doi.org/10.1109/WI.2016.0030.

Breiman, L. (1999). Pasting small votes for classification in large databases and on-line. *Machine Learning*, *36*(1), 85–103. http://dx.doi.org/10.1023/a:1007563306331.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Cheong, M., & Lee, V. (2010). A study on detecting patterns in Twitter intra-topic user and message clustering. In *Proceedings - international conference on pattern recognition* (pp. 3125–3128). IEEE, http://dx.doi.org/10.1109/ICPR.2010.765.

Costa, B. S. J., Bezerra, C. G., Guedes, L. A., Angelov, P. P., & Norte, N. Z. (2016). Unsupervised classification of data streams based on typicality and eccentricity data analytics. In *IEEE international conference on fuzzy systems (FUZZ-IEEE)* (pp. 58–63).

Crespo, F., & Weber, R. (2005). A methodology for dynamic data mining based on fuzzy clustering. *Fuzzy Sets and Systems*, *150*(2), 267–284. http://dx.doi.org/10.1016/j.fss.2004.03.028.

D’Andrea, E., Ducange, P., Lazzerini, B., & Marcelloni, F. (2015). Real-time detection of traffic from Twitter stream analysis. *IEEE Transactions on Intelligent Transportation Systems*, *16*(4), 2269–2283.

Doulamis, N. D., Doulamis, A. D., Kokkinos, P., & Varvarigos, E. M. (2016). Event detection in Twitter microblogging. *IEEE Transactions on Cybernetics*, *46*(12), 2810–2824. http://dx.doi.org/10.1109/TCYB.2015.2489841.

Dovžan, D., Logar, V., & Škrjanc, I. (2016). Evolving fuzzy model (efumo) method for on-line fuzzy model learning with application to monitoring system. *SNE Simulation Notes Europe*, *26*(4), 205–220. http://dx.doi.org/10.11128/sne.26.tn.10352.

Dovžan, D., & Škrjanc, I. (2011). Recursive clustering based on a gustafson-kessel algorithm. *Evolving Systems*, *2*(1), 15–24.

Dutta Baruah, R., & Angelov, P. (2012). Evolving local means method for clustering of streaming data. In *IEEE international conference on fuzzy systems* (pp. 10–15). IEEE.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, *29*(5), 1189–1232.

Giachanou, A., & Crestani, F. (2016). Like it or not: A survey of Twitter sentiment analysis methods. *ACM Computing Surveys*, *49*(2), 1–41.

Godfrey, D., Johns, C., Meyer, C., Race, S., & Sadek, C. (2014). A case study in text mining: Interpreting Twitter data from world cup tweets. *Machine Learning*, *5*, 1–11, URL http://arxiv.org/abs/1408.5427.

Hearst, M., Dumais, S., Osuna, E., Platt, J., & Scholkopf, B. (1998). Support vector machines. *IEEE Intelligent Systems and their Applications*, *13*(4), 18–28.

Iglesias, J. A., Garcia-Cuerva, A., Ledezma, A., & Sanchis, A. (2017). Social network analysis: Evolving Twitter mining. In *2016 IEEE international conference on systems, man, and cybernetics, SMC 2016 - conference proceedings* (pp. 1809–1814).

Jianqiang, Z., Xiaolin, G., & Xuejun, Z. (2018). Deep convolution neural networks for Twitter sentiment analysis. *IEEE Access*, *6*, 23253–23260.

Karami, A., Lundy, M., Webb, F., & Dwivedi, Y. K. (2020). Twitter and research: A systematic literature review through text mining. *IEEE Access*, *8*, 67698–67717. http://dx.doi.org/10.1109/ACCESS.2020.2983656.

Kim, D., Kim, D., Hwang, E., & Rho, S. (2015). Twittertrends: a spatio-temporal trend detection and related keywords recommendation scheme. *Multimedia Systems*, *21*(1), 73–86. http://dx.doi.org/10.1007/s00530-013-0342-0.

Leite, D., Palhares, R. M., Campos, V. C. S., & Gomide, F. (2015). Evolving granular fuzzy model-based control of nonlinear dynamic systems. *IEEE Transactions on Fuzzy Systems*, *23*(4), 923–938.

Leite, D., Škrjanc, I., & Gomide, F. (2020). An overview on evolving systems and learning from stream data. *Evolving Systems*, *11*(2), 181–198.

Li, J., Vishwanath, A., & Rao, H. R. (2014). Retweeting the fukushima nuclear radiation disaster. *Communications of the ACM*, *57*(1), 78–85. http://dx.doi.org/10.1145/2500881.

Lin, F. J., Lin, C. H., & Shen, P. H. (2001). Self-constructing fuzzy neural network speed controller for permanent-magnet synchronous motor drive. *IEEE Transactions on Fuzzy Systems*, *9*(5), 751–759.

Liu, S., Cheng, X., Li, F., & Li, F. (2015). TASC: Topic-adaptive sentiment classification on dynamic tweets. *IEEE Transactions on Knowledge and Data Engineering*, *27*(6), 1696–1709.

Liu, W., Lai, C. H., & Xu, W. W. (2018). Tweeting about emergency: A semantic network analysis of government organizations' social media messaging during hurricane harvey. *Public Relations Review*, *44*(5), 807–819. http://dx.doi.org/10.1016/j.pubrev.2018.10.009.

Lo, S. L., Cornforth, D., & Chiong, R. (2015). Identifying the high-value social audience from Twitter through text-mining methods. In H. Handa, H. Ishibuchi, Y.-S. Ong, K. C. Tan (Eds.) *Proceedings of the 18th asia pacific symposium on intelligent and evolutionary systems, Volume 1, Volume 1* (pp. 325–339).

Lughofer, E. (2011). *Evolving fuzzy systems methodologies, advanced concepts and applications* (first edn.). Springer-Verlag Berlin Heidelberg.

Lughofer, E. (2015). Evolving fuzzy systems - fundamentals, reliability, interpretability, useability, applications. In P. P. Angelov (Ed.), *Handbook on computational intelligence* (pp. 67–136). Berlin Heidelberg: Springer Verlag, chapter 3.

Middleton, S. E., Middleton, L., & Modafferi, S. (2014). Real-time crisis mapping of natural disasters using social media. *IEEE Intelligent Systems*, *29*(2), 9–17. http://dx.doi.org/10.1109/MIS.2013.126.

Onan, A. (2019). Topic-enriched word embeddings for sarcasm identification. In *Advances in intelligent systems and computing, vol. 984* (pp. 293–304).

Onan, A. (2021). Sentiment analysis on product reviews based on weighted word embeddings and deep neural networks. *Concurrency and Computation Practice and Experience*, 1–12.

Onan, A., & Tocoglu, M. A. (2021). A term weighted neural language model and stacked bidirectional lstm based framework for sarcasm identification. *IEEE Access*, *9*, 7701–7722.

Pew Research Center (2020). *Differences in how democrats and republicans behave on twitter*: *Technical report*, (pp. 1–18).

R. A. Fisher, Sc.D., F. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, *7*(2), 179–188.

Rehioui, H., & Idrissi, A. (2020). New clustering algorithms for twitter sentiment analysis. *IEEE Systems Journal*, *14*(1), 530–537. http://dx.doi.org/10.1109/JSYST.2019.2912759.

Sakaki, T., Okazaki, M., & Matsuo, Y. (2013). Tweet analysis for real-time event detection and earthquake reporting system development. *IEEE Transactions on Knowledge and Data Engineering*, *25*(4), 919–931.

Škrjanc, I. (2015). Evolving fuzzy-model-based design of experiments with supervised hierarchical clustering. *IEEE Transactions on Fuzzy Systems*, *23*(4), 861–871.

Škrjanc, I. (2020). Cluster-volume-based merging approach for incrementally evolving fuzzy Gaussian clustering-eGAUSS+. *IEEE Transactions on Fuzzy Systems*, *28*(9), 2222–2231. http://dx.doi.org/10.1109/TFUZZ.2019.2931874.

Škrjanc, I., Iglesias, J., Sanchis, A., Leite, D., Lughofer, E., & Gomide, F. (2019). Evolving fuzzy and neuro-fuzzy approaches in clustering, regression, identification, and classification: A survey. *Information Sciences*, *490*, 344–368.

Tikhonov, A. N. (1943). On the stability of inverse problems. *Doklady Akademii Nauk Sssr*, *39*, 195–198.

Tutaysalgir, E., Karagoz, P., & Toroslu, I. H. (2019). Clustering based personality prediction on Turkish tweets. In *Proceedings of the 2019 IEEE/ACM international conference on advances in social networks analysis and mining, ASONAM 2019* (pp. 825–828).

Twitter (2021a). Academic research product track. In *Twitter API* (pp. 1–7).

Twitter (2021b). Q1 2021 letter to shareholders. *Technical report*, (pp. 1–20).

Vateekul, P., & Koomsubha, T. (2016). A study of sentiment analysis using deep learning techniques on Thai Twitter data. In *2016 13th international joint conference on computer science and software engineering, JCSSE 2016* (pp. 1–6). IEEE, http://dx.doi.org/10.1109/JCSSE.2016.7748849.

Wolberg, W. H., & Mangasarian, O. L. (1990). Multisurface method of pattern separation for medical diagnosis applied to breast cytology. *Proceedings of the National Academy of Sciences of the United States of America*, *87*(23), 9193–9196. http://dx.doi.org/10.1073/pnas.87.23.9193.

Zhang, S., Zhang, X., & Chan, J. (2017). A character-based convolutional neural network for language-agnostic Twitter sentiment analysis. In *2017 international joint conference on neural networks (IJCNN)* (pp. 2384–2391). http://dx.doi.org/10.1145/3166072.3166082.