Anomaly Detection in Social Media using Recurrent Neural Network

Shamoz Shah Madhu Goyal

Centre of Artificial Intelligence, Faculty of Engineering and Information Technology
University of Technology Sydney, PO Box 123, Broadway, NSW 2007, Australia
shamoz.shah@student.uts.edu.au, madhu.goyal-2@uts.edu.au

Abstract. In today's information environment there is an increasing reliance on online and social media in the acquisition, dissemination and consumption of news. Specifically, the utilization of social media platforms such as Facebook and Twitter has increased as a cutting edge medium for breaking news. On the other hand, the low cost, easy access and rapid propagation of news through social media makes the platform more sensitive to fake and anomalous reporting. The propagation of fake and anomalous news is not some benign exercise. The extensive spread of fake news has the potential to do serious and real damage to individuals and society. As a result, the detection of fake news in social media has become a vibrant and important field of research. In this paper, a novel application of machine learning approaches to the detection and classification of fake and anomalous data are considered. An initial clustering step with the K-Nearest Neighbor (KNN) algorithm is proposed before training the result with a Recurrent Neural Network (RNN). The results of a preliminary application of the KNN phase before the RNN phase produces a quantitative and measureable improvement in the detection of outliers, and as such is more effective in detecting anomalies or outliers against the test dataset of 2016 US Presidential Election predictions.

Keywords: clustering, recurrent neural networks, Twitter, presidential election

1 Introduction

In today's information environment, data is being collected, stored and analyzed more extensively to make predictions about client behavior, weather patterns and other natural disasters, espionage and many other sequences that would be beneficial to detect and predict ahead of time. For many data mining algorithms to return beneficial information that has some utility in adding predictive value, the underlying dataset needs to be true, accurate and computable. For this reason, among many others, it is important that algorithms are in place which enable the detection of fake or otherwise anomalous data.[7]

News is pervasive, and fake new is even more so. Fake news at the very least contributes to the misinformation of society. Taken to extremes, fake news can damage

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22747-0_6

entire cohorts of people, societies, institutions and even whole countries. The most recent example of the seriousness of this damage can be seen with the 2016 US Presidential Elections. Many accusations have been made that foreign interests and local groups with vested interests have influenced the integrity and progression of the election. The net result of these accusations are the undermining of the confidence in US sovereign processes and a significant blow to the moral of the citizens of the United States in general.

The damaging effects of misinformation is well known and have been studied for some time. There are valid social, ethical and moral reasons to identify, contain and control the creation and dissemination of fake and misleading information. We use this rational to motivate our work on the identification and classification of false and misleading information.

In the second section of this paper, the related work is discussed. The current state of the art is examined and the methodologies, motivations and rational are compared. From these works, a novel methodology is derived, and presented in section three. The results of the methodology are then presented in section 4, followed by a discussion and conclusion in section 5. The paper closes with a discussion on further areas of research.

2 Related Work

Shu et al [7] outline the human cost of fake, false and misleading information on society as a whole, and we begin there by making a strong case for the need to identify and contain misinformation. Shu makes the point that fake news is very hard to detect from the news item itself, and we need to resort to meta-data such as likes and retweets, friends and followers.Further justification for the utility of meta-data can be found in Akcora [2]. This paper focuses on the clustered behavior of the friend and follower network in Twitter. Users tend to friend and follow other users with the content they are interested in, and in this way Akcora identifies clusters of users who may exhibit the same behavioral patterns.

Telang [8] introduces the use of location data to model the spatio-temporal characteristics of Twitter data, conducting experiments in the identification of spatial patterns in sentiment data and the identification of the temporal aspects of weather data. The innovation in Telang is that they model anomalies as spatial or temporal deviations from the normal distribution. That is to say, if an event occurs out of spatial or temporal locality to the mean, it is deemed to be anomalous. A further development by Oancea (thesis, cannot cite) is the use of statistical noise in combination with a Kalman Filter. The resultant algorithm is called an Extended Kalman Filter, which can not only process linear data, but data that is non-linear and differentiable.

Zhao et al. [10] discusses the use of a novel graphical representation of anomalous data on Twitter. The data is presented as nodes in time, which change color and shape based on the activity surrounding or spawned by that node. In this way, it is possible to track the changing nature of the Twitter data landscape in real-time, with rich visual

DOI: 10.1007/978-3-030-22747-0_6

detail. Rere et al. [6] discusses the use of deep learning to make computing more efficient and to reduce the cost of computing chips, but doesn't remark on the mechanisms by which such occurs. The paper is not available in full, but the discussion centers around improving the training speed via a process called “simulated annealing”.

Thom et al. [9] further discusses the utilization of geolocation information from Twitter data. The paper focuses on using Twitter geolocation information to detect anomalies in spatiotemporal data. This approach can be useful for detecting emerging threats and disasters.

Macneil et al. [5] discusses the use of biologically probable weighting algorithms to improve the convergence behavior and stability of recurrent neural networks. Akbari et al. [1] delves into the detection of anomalies using a KNN algorithm, by detecting points that do not conform to a normalized cluster.

3 Methodology

3.1 Definition of Fake News

Fake news has existed since the existence of news itself. Where there has been the dissemination of news for the purposes of knowledge and information, there have been parties which wish to subvert the news for the purposes of propaganda and disinformation. It comes as no surprise then, that the occurrence of fake news increased exponentially with the advent of the printing press and the widespread adoption of the newspaper, printed books and other media. Today, there are many different forms of media being used to create, update and disseminate news. The trend is to move from a digest format to more real-time and close to real-time avenues. One such avenue is Twitter. In fact, Twitter stands today as the most breaking source of news available.

To detect and classify fake and anomalous news, what constitutes such news must be defined and quantified. As there is no standard definition of fake news, a discussion on what qualifies as fake news for the purposes of this paper needs to be defined. A working definition of fake news is that it is intentionally and verifiably false. That is, fake news has the deliberate intent to mislead the consumer and the authenticity of the news itself is deliberately falsified or, at the very least, questionable. This definition implies a two-fold test of accuracy. These points are *authenticity* and *intent*.

For this definition to be applicable, fake news has to be unquestionably verified as false, possibly by a human reader or a machine learning algorithm. The next criterion is that the news has been created to deliberately mislead consumers. Some literature treats satire as fake news, even though satire is deliberately designed to be entertainment oriented and makes its use of deceptiveness clear to the consumer. Other literature treats any deceptive news as fake news, including hoaxes, fabrications and satire. In this paper, we constrain ourselves to a very explicit definition of fake news -

Fake news is a news article that is intentionally and verifiably false.[7]

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22747-0_6

3.2 Proposed Algorithm

A methodology was proposed whereby the Euclidean proximity of data points could be exploited to reveal some hidden underlying features of the data. The spatially enhanced data can then be passed through a further processing phase that can augment the detection of probable anomalies. After some preliminary research on various current algorithms, it was proposed that a clustering algorithm be chained to an artificial neural network and the data evaluated against such a composite classifier.

A few clustering algorithms were considered, but the most favorable clustering algorithm for the purposes of this experiment was the K-Nearest Neighbor (KNN) algorithm due to its lack of parametric requirements [11,12]. As a result, the KNN algorithm makes no assumptions about the probability distributions of the data points being processed [12]. This is a very important feature of KNNs, since it ensures that any pertinent spatial linearity or coherence is preserved for any subsequent processing.

For the second phase of the methodology, it was proposed that the KNN data be passed through a Recurrent Neural Network (RNN). The RNN was trained on the output of a preceding KNN phase and then the results of the classifier chain were tested against a novel dataset.

3.3 Clustered RNN Pseudocode

The pseudocode for the classifier process is as follows:

```
Load(datafile);
Set optimizedRNN = rnn(seed=41, learningrate = 0.5, epochs = 5000, momentum = 0.1, transferfunction = “Gompertz”);
For k  = (1, 10, 20)
  model = KNN (k, datafile);
  trainedModel[k] = optimizedRNN(model);
End For
Return trainedModel[];
```

The data file is first loaded. Next, the RNN training node is created with optimized parameters passed as an argument. The training phase is now ready to begin. For each of the value for k, the KNN is trained and then passed to the optimized RNN node. This is done for three values of k, and the results are returned in a vector.

4 Results

4.1 Dataset

The dataset of 2016 US Presidential Election results is used for prediction and to test the proposed algorithm. The classifier is first trained on the actual election results and then tested against the result predictions to ascertain how effective the classifier optimizations were in predicting voting behavior. To ensure that our algorithm would be applicable to real-world scenarios, the data we obtained was from the 2016 US Presidential Elections. The first set of data that is obtained was the actual (final) result of the US election vote counting. This data was presented as the number of votes the Democrat Party had won, as a percentage of the total votes. We used this data set as the training input for both the KNN and the RNN phases .The second set of data that was obtained were predictions of the election results before the election had finished and the votes had been counted. There were a number of data points in the data set that was not required, and the data had to be cleaned and formatted for use in the experiments that were being conducted. The cleaned version of this data formed the testing set for both the KNN and RNN algorithms.

4.2 KNN Clustering Features

It was decided to use standardized algorithms available with R. This would ensure that the results were replicable, accurate and testable. To obtain an accurate idea of the behavior of the KNN algorithm with the dataset in question, a range of k values were used. The k-values used with the KNN algorithm were 1, 10 and 20. Following the training phase, the combined algorithm (Clustered RNN) was tested against a novel dataset to evaluate whether the classifier improved the accuracy of anomaly detection.

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22747-0_6

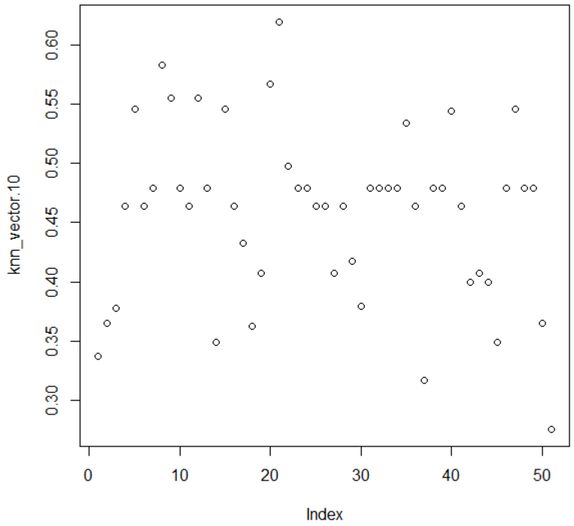

Fig. 1. - KNN Cluster (Mean 10)

As can be seen in Figure 1, an application of KNN with a mean of 10 does little to cluster the data sufficiently enough to be useful in the prediction of outliers. There are some patterns that are starting to form, and it is advantageous to attempt to increase the expression of these patterns with a further application of the KNN algorithm.

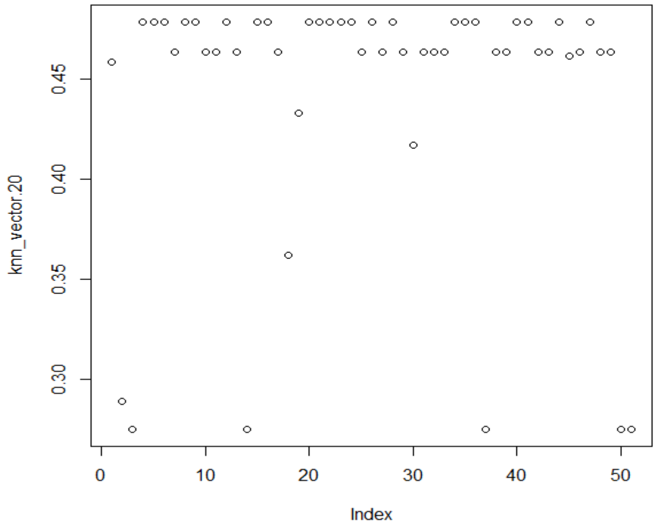

Fig. 2. - KNN Cluster (Mean 20)

In the next iteration of the experiment, we apply the KNN clustering algorithm with a mean of 20. As can be seen from Figure 2, the adjustment of the mean parameter has started to cluster the data points quite nicely. From this application of the KNN algorithm, it can be seen that there are three points in the middle of the graph that have not conformed to either the top cluster or the bottom cluster. For the purposes of the experiment, we consider these points as anomalous, or outliers to the normal distribution.

The first graph (Figure 1) shows the result of training an un-optimized RNN against the 2016 US election datasets. The RNN is already somewhat effective in predicting the spatio-temporal distribution of the test dataset (predictions)against the benchmark dataset (winners). However, we wish to delve a little further into the parameters that we train our RNN with, in an attempt to present the most optimal training scenario for our data mining experiment.

To improve the behavior of our prediction model, we next attempted to optimize the RNN portion of the algorithm before adding the clustering step. We assume that the RNN optimization behavior correlates linearly with the application of a clustering algorithm, so it stands to reason that optimizing the RNN would yield more accurate results. We investigated a number of scenarios to optimize the RNN and through experimentation, have come to use the following parameters for the RNN –

- Seed – 41 (for replicability)
- Learning Rate – 0.5
- Epochs – 5000
- Momentum – 0.1

DOI: 10.1007/978-3-030-22747-0_6

- Transfer Function - Gompertz

We present this optimized RNN with respect to the first (unoptimized) RNN and note the improvement in accuracy.

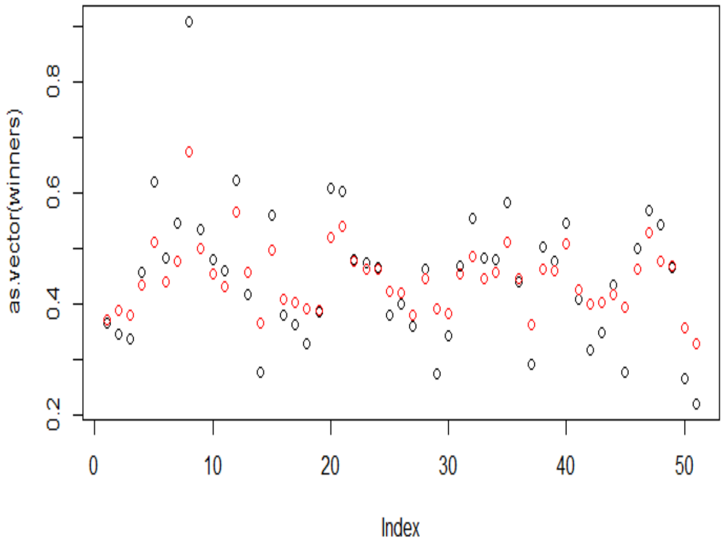

Fig. 3. - RNN without Optimization

As we can see from the graph (Figure 2) the optimizations applied to the second RNN enable it to produce results with more spatio-temporal coherence than the first iteration. The prediction accuracy of the optimized RNN has increased significantly. It follows that any clustering algorithm applied to the RNN as a pre-processing step will benefit from this optimization in at least a linear fashion.

For the next phase in the experiment, we aim to combine the optimized RNN with a k nearest neighbor clustering algorithm (KNN). We used a number of different parameters for the KNN, with different distance parameters. What we found was that the final clustered RNN result was highly sensitive to the KNN distance parameter used. For the KNN training phase of the algorithm, we used the following parameters –

- K value – 10, 20

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22747-0_6

We show the result of running the complete algorithm (KRNN) with a mean value of 1. From what we can see, there is very little difference between the optimized RNN and a KRNN with a mean of 1. The results are shown graphically in Figure 3.

To examine what significance the clustering parameters have on the final results, we attempt another trial with our experiment, setting the mean value to 10. As can be seen in Figure 4, the final result is more spatio-temporally coherent than running the algorithm with a mean of 1.

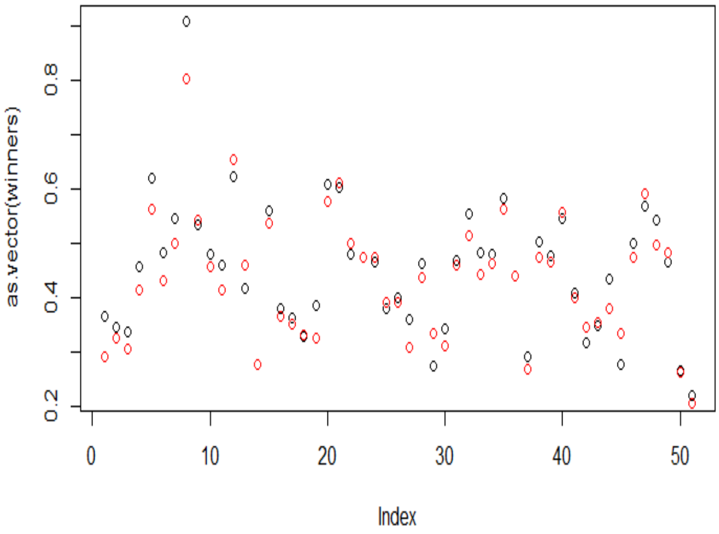

Fig. 4. - Optimized RNN

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22747-0_6

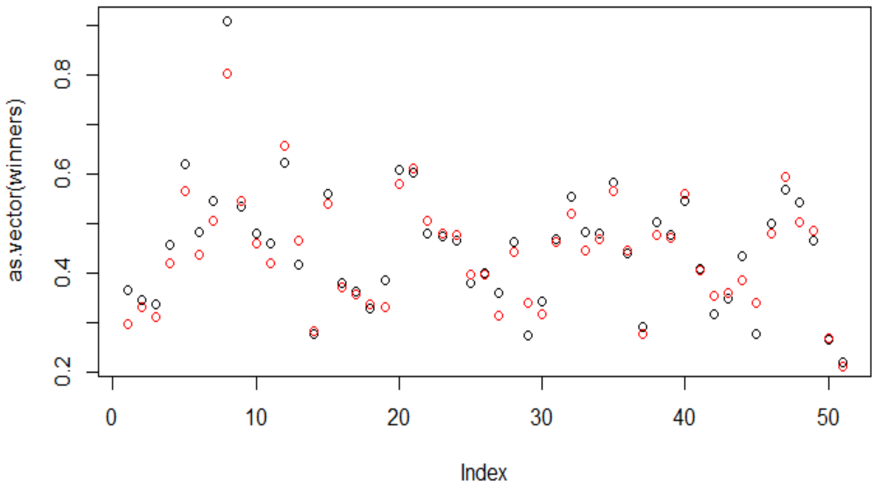

Fig. 5. -KRNN with Mean 1

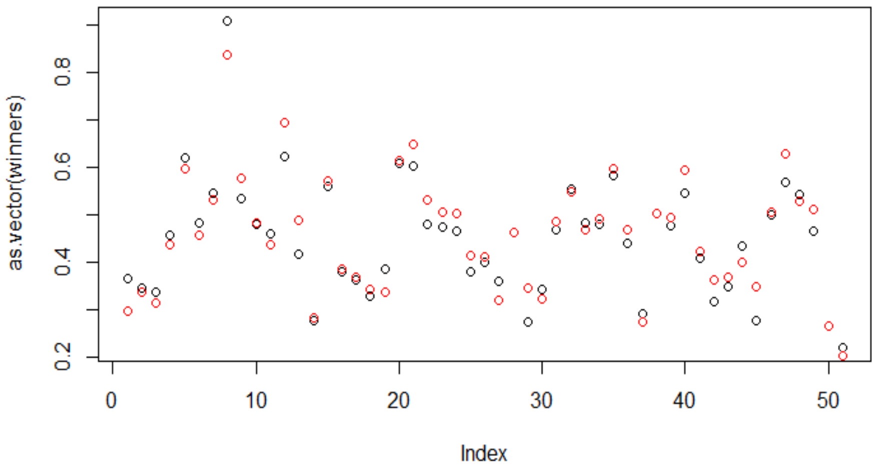

Fig. 6. - KRNN with Mean 10

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22747-0_6

For the final test, we change the k parameter to a distance of 20 and re-execute the algorithm. As can be seen in Figure 5, there is qualitatively an improvement of the spatial behavior of the data, but not a great decrease in the prediction error.

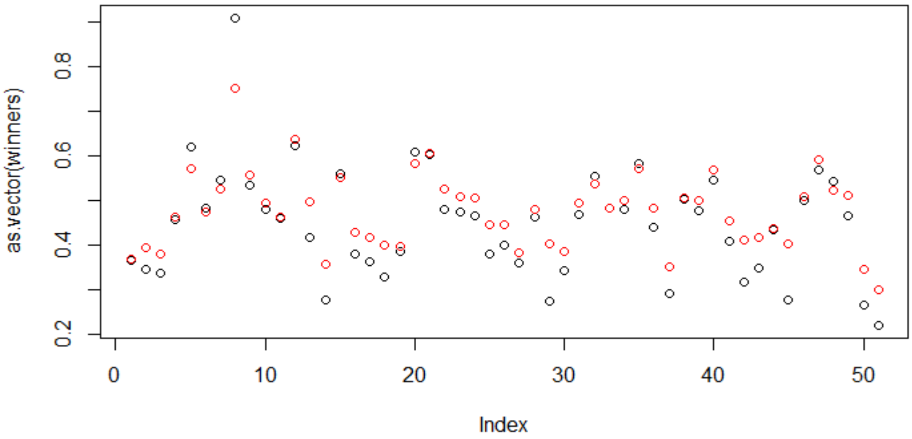

Fig. 7. -KRNN with Mean 20

From inspection of the KRNN.10 and KRNN.20 results, we find that there is a significant improvement in spatial coherence when we first cluster the data before feeding it through an RNN phase. The predicted data points have better spatial coherence and are aligned to the training set a lot more closely than they were with a single application of RNN alone. From the results of the experiment, it is obvious that applying a clustering phase before the RNN phase results in predictions that are spatially closer to the actual results.

5 Conclusions

The experiments and results show that clustered RNNs are highly sensitive to the initial clustering step. The distance chosen affects the distribution of the final points significantly. It was also noted that the distribution of the test dataset does not vary in a linear fashion with respect to the Euclidean distance selected in the KNN step. Further work in this area can focus on generating distribution maps of the KNN stage to further optimize the results that are obtained. Another area that begs examination is the use of different clustering algorithms or even the application of more pre-processing algorithms before the RNN step. Our future work will also focus on comparison with other Fake news detection techniques.

References

1. Akbari M, Overloop PJV, Afshar A (2010) Clustered K Nearest Neighbor Algorithm for Daily Inflow Forecasting. Water Resources Management 25:1341–1357. doi: 10.1007/s11269-010-9748-z
2. Akcora CG, Carminati B, Ferrari E, Kantarcioglu M (2014) Detecting anomalies in social network data consumption. Social Network Analysis and Mining. doi: 10.1007/s13278-014-0231-3
3. Chen C-H, Huang W-T, Tan T-H, et al (2015) Using K-Nearest Neighbor Classification to Diagnose Abnormal Lung Sounds. Sensors 15:13132–13158. doi: 10.3390/s150613132
4. Hu L-Y, Huang M-W, Ke S-W, Tsai C-F (2016) The distance function effect on k-nearest neighbor classification for medical datasets. SpringerPlus. doi: 10.1186/s40064-016-2941-7
5. Macneil D, Eliasmith C (2011) Fine-Tuning and the Stability of Recurrent Neural Networks. PLoS ONE. doi: 10.1371/journal.pone.0022885
6. Rere LM, Fanany MI, Arymurthy AM, (2015) Simulated Annealing Algorithm for Deep Learning, The Third Information Systems International Conference 72:137-144. doi: 10.1016/j.procs.2015.12.114
7. Shu K, Sliva A, Wang S, et al (2017) Fake News Detection on Social Media. ACM SIGKDD Explorations Newsletter 19:22–36. doi: 10.1145/3137597.3137600
8. Telang A, Deepak P, Joshi S, et al (2014) Detecting localized homogeneous anomalies over spatio-temporal data. Data Mining and Knowledge Discovery 28:1480–1502. doi: 10.1007/s10618-014-0366-x
9. Thom D, Bosch H, Koch S, et al (2012) Spatiotemporal anomaly detection through visual analysis of geolocated Twitter messages. 2012 IEEE Pacific Visualization Symposium. doi: 10.1109/pacificvis.2012.6183572
10. Zhao J, Cao N, Wen Z, et al (2014) #FluxFlow: Visual Analysis of Anomalous Information Spreading on Social Media. IEEE Transactions on Visualization and Computer Graphics 20:1773–1782. doi: 10.1109/tvcg.2014.2346922
11. (2017) Nonparametric statistics. In: Wikipedia. https://en.wikipedia.org/wiki/Nonparametric_statistics. Accessed 11 Dec 2017
12. (2017) K-nearest neighbors algorithm. In: Wikipedia. https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm. Accessed 11 Dec 2017