Enabling weather-based decision making for forestry pest and disease management: Modelling Report

Connor McDonald
University of Pretoria
Department of Computer Science
u16040725@tuks.co.za

Gené Fourie
University of Pretoria
Department of Computer Science
u20797274@tuks.co.za

Keywords

Sirex noctilio, Leptocybe invasa, Forestry pest and disease management

1. DISCUSSION

This research has been conducted in partnership with the Forestry and Agricultural Biotechnology Institute (FABI). We aim to analyse weather patterns across South Africa in conjunction with the pest prevalence records for the Sirex noctilio and Leptocybe invasa pests to determine if there is any relationship between changing climatic patterns and pest prevalence. Three datasets are provided for this research, these are listed below:

1. Temperature and rainfall records of roughly 6000 forestry plantations across South Africa
2. Sirex noctilio (Sirex) pest inspection samples
3. Leptocybe invasa (Leptocybe) pest inspection samples

1.1 Research Framework

The aim of this research is to identify any patterns or correlations between the changing climatic patterns of South Africa, and the prevalence of Sirex noctilio and Leptocybe invasa in Pine and Blue Gum plantations, respectively. To maintain the integrity of this research, we adopt the PCS framework proposed by Yu and Kumbier [1] in their paper *"Veridical Data Science"*. We focus specifically on stages B, C and D, which are detailed below.

B. Predictability

The first step after data collection/processing and exploratory data analysis is typically some sort of modelling to identify complex relationships in the data [1]. However, this requires a prediction function that models the relationship between independent and dependent variables. Since we have three datasets, with no linking attributes, this can be a difficult task. To link these datasets, we make use of Voronoi diagrams to generate convex polygons around the given weather stations; the pest records which fall in a given polygon will be assigned the weather records of the generating station within that polygon. The next problem is the way in which pests are recorded between the two datasets. Sirex observations are recorded with a simple binary indicator, while Leptocybe observations have a binary indicator as well as a severity metric. To maintain consistency between the two pest datasets, we focus on predicting the binary indicator only. First, we look at unsupervised approaches to cluster the data points based on similarity, and thereafter examine the percentage of positive pest inspections within different clusters to determine if certain characteristics lead to higher pest presence. Secondly, we develop various classification models which aim to predict whether a given record contains a positive pest inspection, based on the same features used to cluster the data points. To ensure the validity of our results, we train our classification models on 75% of data, and validate the results on the remaining 25% of data. Evaluation is difficult as the dataset is imbalanced with the majority class (that is, the non-pest presence) making up around 75% of the data. Clustering models are evaluated using the silhouette score which represents the goodness of fit for a given clustering algorithm. Alternatively, classification models are evaluated with the Receiver Operator Characteristic (ROC) curve and Area Under Curve (AUC) score, which measures how accurately the model can predict both classes. Lastly, models are subjected to a k-fold cross validation to determine the robustness of the results.

C. Computability

Training and optimising machine learning models can be computationally intensive tasks. To reduce the complexity of these tasks, we performed data-preprocessing and feature engineering to reduce noise, this is detailed in sections 1.2 and 1.3. This reduces the time taken to tune and train the models as there are less data points that the models need to learn from. When optimising parameters, an exhaustive search of the parameter featurespace would be computationally infeasible, and thus a *"Grid Search"* approach was taken. While this method can lead to better model performance, it can drastically increase the computational complexity of models. This was observed when trying to optimise too many parameters for only marginal performance gains. Lastly, trained models were saved as files that could be reused at a later stage. This was done to prevent having to train the model each time an independent party tries to replicate our results.

D. Stability

To evaluate the stability of an observed result we measure the change observed in the target under various perturbations of both the data and the learning algorithms [1]. We create three perturbations, the first perturbation contains no raw values, but rather differences from the average. The second perturbation contains the same data as the first perturbation with the addition of the raw data, and averages of the raw data, that are used to calculate the differences in the first perturbation. The third perturbation considers single measurements, such as the rainfall differences from average, or combinations of measurements. However, only favourable combination outcomes are included in this report due space restraints. We also perform a 3-fold cross validation on each model for all perturbations.

To generate algorithm perturbations, we assess four models, namely: Gaussian mixture models, time series K-means Analysis, Support Vector Machine and XGBoost classifier. We discuss why

and how these models are implemented in sections 1.4 and 1.5, and we evaluate their performance in section 1.6.

1.2 Data Pre-processing

The data received has no linking attribute between the datasets containing the dependent variables (pest prevalence) and the dataset containing the independent variables (weather conditions). To overcome this disjoint, we use Voronoi diagrams, also known as Dirichlet tessellations. Voronoi diagrams work by *"partitioning a plane with n points into convex polygons, such that each polygon contains exactly one generating point and every point in a given polygon is closer to its generating point than to any other"* [2]. This computational geometry tool has been cited in many areas of research such as closest-site problems, triangulating sites, clustering point sites, and connectivity graphs for sites [4].

Voronoi diagrams allow us to group stations and pest records by a predetermined region rather than a single geographical coordinate or site name. This is known as a *"Nearest Site"* problem in literature and allows us to assign the weather records of a generating station to all pest records that fall within that station's polygon. Figure 1 visualises this concept.

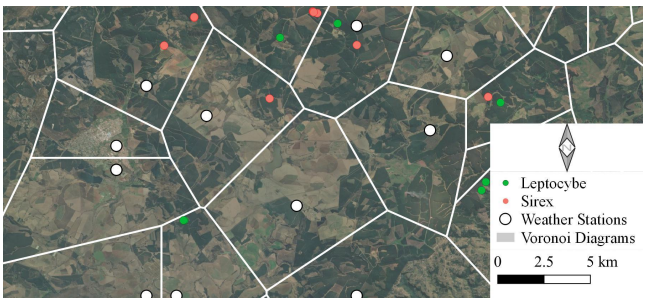

Figure 1: Voronoi Diagram

However, the exploratory data analysis indicates that weather stations have periods of activity and inactivity, where older stations are often replaced by newer stations with a different site identifier. Additionally, some weather stations capture a single measurement, such as rainfall or temperature, whereas, others capture both rainfall and temperature data. As a result, feature engineering is required to determine active stations throughout the measurement period and to assign a station's weather readings to pest inspection(s) within its Voronoi Diagram demarcation.

1.3 Feature Engineering

Section 1.3 provides the algorithm and assumptions made to interface the weather data with the pest data to ensure that only active stations are used in the analysis.

1.3.1 Step 1) Classify active weather stations

A weather station is deemed 'active' if a reading is provided for every day of a month - where the 'reading' is defined as either a rainfall, maximum temperature, or minimum temperature measurement. Readings equal to '-99.9' are deemed null entries [3] and excluded from analysis.

Measurements are aggregated per month and per active station, spanning the period between January 1950 and July 2019. For example, the measurements for station '0004694_A' are aggregated for Jan-1950, Feb-1950 and so forth. Rainfall measurements are summed per month, whereas temperature measurements are averaged per month. Additionally, the temperature difference per day is calculated and averaged per month. This results in four weather measurements for analysis: *rainfall*, *temperature_max*, *temperature_min, temperature_diff*.

1.3.2 Step 2) Match active stations to pests

This analysis is performed on a month-to-month granularity between January 1950 and July 2019, using only stations with complete measurements for that month. The location of each active weather station and pest inspection is reprojected to the WGS84/Pseudo-Mercator Coordinate Reference System (suited to South Africa [6]) to allow for Euclidean distance measurement in metres between station and pest coordinates. The algorithm identifies the closest active station per pest inspection, per measurement type (rainfall, maximum and minimum temperature) and per month-year period. This analysis ensures that if a station is found as inactive during a period, then the Voronoi diagram is redrawn, and the next closest station is assigned to provide weather measurements to a pest inspection location.

Table 1, Table 2, and Figure 2 provide an example of the station mapping (measuring rainfall) for one Leptocybe inspection site. Note that although station 1067 is the closest station to the pest inspection (distance=2674m), the algorithm reassigns station 5635 (distance=3038m) to the pest location due to the inactivity of station 1067 from August 2000 onwards. As a result, the expected rainfall conditions for Leptocybe ID 1's location are derived from Station ID 1067 prior to August 2000, and from Station ID 5635 for August 2000 and succeeding months - until Station ID 5635 is deemed inactive due to insufficient readings, or a closer station to the pest inspection is identified.

Table 1: Weather station ID per Leptocybe inspection

Leptocybe ID	May 2000	Jun 2000	Jul 2000	Aug 2000	Sep 2000
1	1067	1067	1067	5635	5635

Table 2: Number of rainfall measurements per station ID

Station ID	May 2000	Jun 2000	Jul 2000	Aug 2000	Sep 2000
1067	31	30	31	0	0
5635	31	30	31	31	30

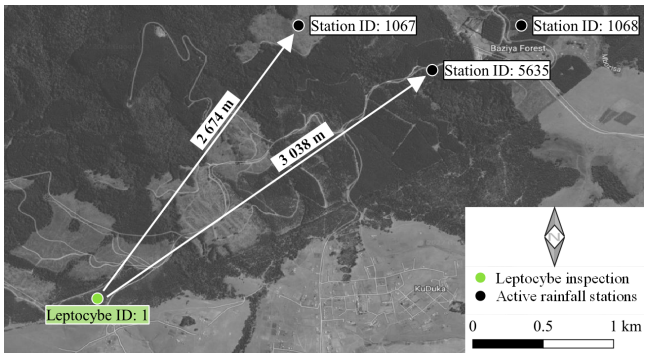

Figure 2: Pest to station mapping

1.3.3 Step 3) Calculate average readings per month

Using the aggregate readings (Step 1) derived from the active stations per month and per pest ID (Step 2), the monthly average rainfall, and maximum and minimum temperature expected at pest sites can be calculated. The algorithm generates four parameters, each with 12 unique data points per pest inspection (unless stations are shared amongst two or more inspections). The parameters generated are: *rainfall_avg*, *temperature_max_avg*, *temperature_min_avg*, *temperature_diff_avg*.

1.3.4 Step 4) Determine actual readings per month

The pest inspection data is captured yearly, with the inspection date absent from the data. To correlate the pest inspection 'year' attribute to weather readings, the algorithm collects the actual weather readings per month-year period (Step 1) from the assigned active station (Step 2), for three years prior to and during the inspection year. This results in 48 data points per pest ID, that is, one reading per month. The timeseries attribute is termed *timeframe* to represent the months prior and including the inspection year. For example, a pest inspection performed in 2017 has weather data points for January 2014 (timeframe=1), February 2014 (timeframe=2), through to December 2017 (timeframe=48). The parameters generated per timeframe and pest ID are: *rainfall*, *temperature_max*, *temperature_min*, and *temperature_diff*.

1.3.5 Step 5) Calculate difference between actual and average readings per month

To determine a change from normal conditions, the algorithm subtracts the average readings per month (Step 3) from the actual readings per month (Step 4). For example, the average reading for January would be subtracted from the actual readings for the following timeframes: 1, 13, 25, and 37. Following the same naming convention as above, the parameters generated per timeframe and pest ID are: *rainfall_diff*, *temperature_max_diff*, *temperature_min_diff*, and *temperature_diff_diff*.

1.3.6 Step 6) Format data for modelling

Table 3 provides an extract of the output of the feature engineering stages described in Step 1 to 5 above. The *temparature_min* and *temperature_diff* columns are appended to the right of Table 3, with the remaining pests IDs and timeframes below Table 3. Advantages of the data in this form include:

- Allows for addition of other attributes from the pest data, using the Pest ID. For example, models can include the binary indicators for Sirex Presence or percentage infestation of Leptocybe inspections.
- Assists in modelling the weather patterns collectively, rather than per weather type (rainfall/temperature), or on a yearly, or site-specific basis.
- Prepares the data for unsupervised clustering and time-series modelling, as well as supervised methods.

Table 3: Extract of engineered data

Pest ID	time frame	rainfall_avg	rainfall	rainfall_diff	temperature_max_avg	temperature_max	temperature_max_diff
1	1	184.79	151.5	-33.29	27.11	29.34	2.23
2	1	281.92	219.5	-62.42	27.78	28.43	0.65
3	1	253.37	301.5	48.13	27.87	28.43	0.56

1.4 Unsupervised Methods

The algorithms discussed in this section are primarily used to cluster the data points based on similarity. An initial hypothesis suggests that certain clusters may have higher pest prevalence, which then allows for investigation into a cluster's defining characteristics as a potential cause of elevated pest presence.

1.4.1 Gaussian Mixture Model

Gaussian mixture models (GMM) are a semi-parametric technique that is used to cluster multimodal data. GMMs use overlapping multivariate Gaussian distributions, which model the probability of a point belonging to the cluster associated with a given distribution, and have been used in a number of fields including weather data analysis [5]. Table 4 shows the variation of percentage positive cases across the clusters, the lower values indicate that the percentage of positive cases was uniformly distributed across clusters. The second perturbation of the Leptocybe dataset has been highlighted due to its abnormally high standard deviation (σ) in comparison to the first perturbation. This is largely due to how the data were clustered, in this specific case, one cluster contained ~90% of data. The remaining 3 clusters had relatively high proportions of positive Leptocybe inspections but only contained ~13% of all positive inspections. This sample size was too small to justify that the characteristics of those clusters correlated with pest presence. Ultimately, GMM was rejected as a possible modelling algorithm due to the poor silhouette scores achieved in all data perturbations, indicating a poor fit.

Table 4: Standard Deviation of pest proportion across clusters

Perturbation	Leptocybe σ	Silhouette score (Leptocybe)	Sirex σ	Silhouette score (Sirex)
1	3.19%	0.02	1.48%	0.06
2	20.96%	-0.1	2.53%	0.23

1.4.2 Time-series K-means Analysis

K-means clustering performs partitional clustering through a repetitive process of identifying the closest centroid to a data point and repeating the process until clusters are unchanged [7]. The use of the technique was deemed advantageous to model the pest and weather data due to its simplicity and robustness, as well as the ability to interpret the output of the model for explainable results. Considering this, the number of parameters for perturbation 2 is deemed too complex for explainability of clustering results, therefore, combinations of perturbation 3 are explored. Table 5 indicates the variation of positive cases across the clusters. Note the higher σ for Leptocybe which indicates a higher percentage of positive inspections in a particular cluster. Perturbation 3 for Leptocybe is used to explore the large σ seen for Perturbation 1.

Table 5: Standard Deviation of pest proportion across clusters

Perturbation	Leptocybe σ	Silhouette score (Leptocybe)	Sirex σ	Silhouette score (Sirex)
1	26.30%	0.17	4.28%	0.10
2	-	-	-	-
3	23.92%	0.21	3.03%	0.08

For Perturbation 3, Figure 3 provides the best performing cluster centroids of three clusters identified for Leptocybe inspections, with a silhouette score of 0.21, using *rainfall_diff* and *timeframe*<37, that is, before the Inspection Year (IY). Note that Cluster 1, the blue cluster, contains ~20% of data, however,

contains ~45% of all positive Leptocybe inspections. The highlighted section in Figure 3 indicates a definite lower rainfall region compared to normal conditions.

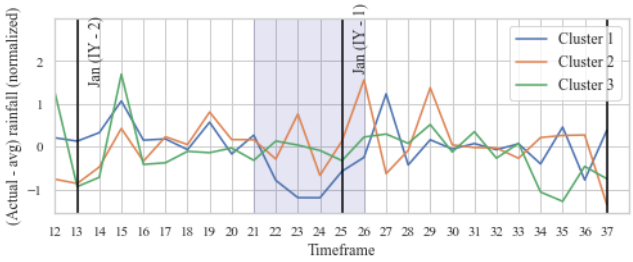

Figure 3: Leptocybe cluster centroids

Alternatively, the highest silhouette score for the time-series K-means clustering for Sirex data is 0.10. In general, clustering over all timeframe periods and parameters yields non-favourable results in terms of separating positive and negative Sirex inspections. All clusters on average, with varying K, show ~30% positive Sirex inspections, which inhibits the identification of any changing climatic conditions from clustering techniques alone.

Consequently, K-means is rejected due to insufficient data for Leptocybe, and poor clustering of Sirex. However, the lower-than-average rainfall period shown in Figure 3 indicates potential for future modelling of drought and pest presence.

1.5 Supervised Methods

In this section we investigate two supervised classifiers. The task at hand is sensitive to type II errors (false negative), therefore we aim to achieve high classification accuracy of the minority class.

1.5.1 Support Vector Machine

Support Vector Machines (SVM) are a type of classification algorithm which finds a decision boundary between classes such that the margin between the classes is maximised. SVMs can classify non-linearly separable data by using soft-margins and kernel methods, this is the primary reason for using this algorithm to model our data. The final model makes use of a radial basis function kernel, which consists of two parameters, namely the tolerance (C) and the influence (γ). We use a 3-fold cross-validated grid search to assess the effect of various values of these parameters from 0.1 - 1000 and find that the default values of ($C = 1$; $\gamma = \frac{1}{(\#features \times variance)}$) work best. Lastly, a penalisation factor is added for misclassification of the minority class, which is performed due to the imbalance of the data. Table 6 shows the performance of the model over the two perturbations of data, the model performs best on the second perturbation for both pest types. We opt for higher accuracy in the minority class at the expense of accuracy in the majority class as the scenario being modelled is highly sensitive to type II errors. Table 6 highlights the model's poor performance when predicting the majority class, especially with the Sirex pest.

Table 6: Classification Accuracy of SVM model

Perturbation	Class	Leptocybe	Sirex
1	Minority	62%	69%
	Majority	58%	43%
2	Minority	71%	71%
	Majority	64%	51%

1.5.2 XGBoost

XGBoost is a relatively recent algorithm that makes use of gradient boosted decision trees. The algorithm has seen great success in industry as it was built with speed and performance in mind. Much like the SVM model, the default parameters - with the addition of a penalisation factor for misclassification of the minority class - produce the best results.

However, it should be noted that the XGBoost model is incredibly complex and has over 30 parameters which can be optimised, therefore it is not computationally feasible to search the entire parameter space for the optimal configuration. Instead, we construct a 3-fold cross-validated grid search focused on: gamma, learning_rate, max_depth, n_estimators, reg_lambda, reg_alpha.

Table 6 details the results of this model. The results indicate that XGBoost performed significantly better than SVM when classifying both, due to the higher AUC score for the XGBoost in both perturbations of the dataset and for both types of pest classification. Furthermore, the model runs significantly faster than the SVM model in all instances.

Table 7: Classification Accuracy of XGBoost model

Perturbation	Class	Leptocybe	Sirex
1	Minority	66%	85%
	Majority	67%	38%
2	Minority	82%	85%
	Majority	85%	57%

1.6 PCS CHECK

1.6.1 Model Stability

To test the stability of the models, we perform a 3-fold cross validation. The overall accuracy of each fold is recorded, and the standard deviation of all accuracies is calculated. Table 8 and Table 9 indicate the fold accuracies for Leptocybe and Sirex respectively. The fold accuracies are shown to have very low standard deviation which indicates a high degree of stability in the models.

The XGboost model trained on the second data perturbation performs 10-15% better than the SVM model trained on the same data; this is observed when classifying both pests.

Table 8: Leptocybe Models Cross Validation

		Fold			
Model	Perturbation	1	2	3	st.dev
SVM	1	57%	56%	57%	0.53%
	2	63%	65%	63%	0.57%
XGBoost	1	66%	65%	66%	0.64%
	2	83%	81%	81%	1.03%

Table 9: Sirex Models Cross Validation

		Fold			
Model	Perturbation	1	2	3	st.dev
SVM	1	50%	51%	49%	0.67%
	2	56%	56%	55%	0.37%
XGBoost	1	50%	52%	51%	0.57%
	2	65%	64%	64%	0.33%

1.6.2 Model Repeatability

To ensure repeatability, the data should be subjected to the same preprocessing and feature engineering steps detailed in sections 1.2 and 1.3. Furthermore, the default parameters should be used for the models with a penalisation factor equal to $\frac{Number\ of\ Majority\ class\ records}{Number\ of\ \ Minority\ class\ records}$. Lastly, all the necessary packages and their versions, as well as the code used to generate the models in 1.4, 1.5 and 1.6, have been detailed in the README file on the repository containing the code for this research, which can be found at: GitHub Repository

1.6.3 Model Evaluation

Ultimately, the models are evaluated on their ability to accurately classify both classes, to do this we plot the ROC-AUC curves of all models used to classify a given pest. Figure 3 and Figure 4 show the ROC-AUC curves for Leptocybe and Sirex respectively.

In general, the models can classify the Leptocybe pest more accurately than the Sirex pest. This finding is evident in the ROC-AUC as well as the class accuracies detailed in Tables 6 and 7. It should also be noted that the classification accuracy of negative Sirex inspections is especially poor, between 38% and 57% for all models across both perturbations. Models that are trained on the second data perturbation perform significantly better than those trained on the first perturbation. This is likely because the second perturbation contains 13 features while the first only contains 5 features.

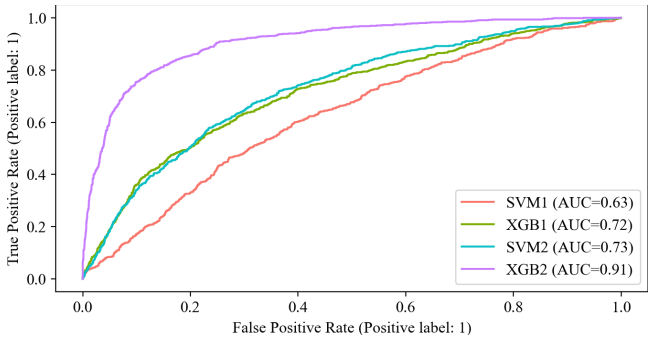

Figure 3: Leptocybe Classification Performance

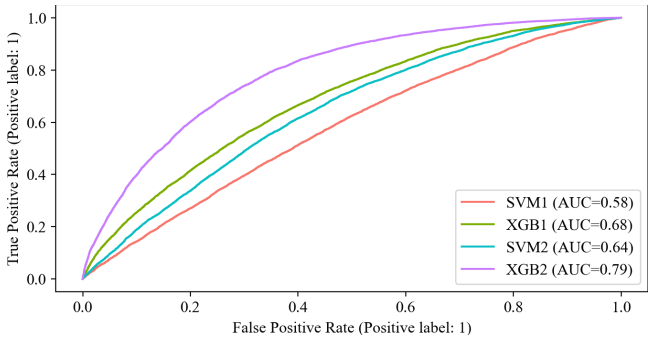

Figure 4: Sirex Classification Performance

2. CONCLUSION

The objective of this research task is to link the given pest and weather records for the Sirex noctilio and Leptocybe invasa pests and use machine learning modelling techniques to uncover relationships between pest prevalence and changing climatic conditions in South Africa. Two unsupervised methods, namely Gaussian mixture modelling and Time-series K-means Analysis are investigated in an attempt to cluster the data to identify higher pest prevalence in specific clusters. However, these methods are rejected due to negligible differences in pest presence across clusters, and insufficient data for Leptocybe. Additionally, two supervised methods, namely, XGBoost and Support Vector Machine, are investigated, with the aim to classify pest presence based on given weather characteristics. The XGBoost model trained on the second data perturbation performs best with an overall classification accuracy of 81% for Leptocybe and 64% for Sirex.

2.1 Future Work

The models investigated in this research are based on the data points generated through the feature engineering exercise detailed in this report. However, an assumption is made that only three years of weather data prior to the inspection year of pest records is sufficient to identify correlations between climatic conditions and pest prevalence. Future work can include considering weather patterns in years prior to the current 3 years selected, or in the months directly prior to a pest inspection - only if the exact inspection date, rather than the inspection year, is known.

The models primarily focus on classification and clustering, with only the Time-series K-means Analysis approach considering how and if relationships between pests and weather change over time. We suggest that future work include additional time-series modelling such as K-shape analysis, which clusters time-series, or Gaussian mixture regression, to construct a regression model for each cluster in a Gaussian mixture model.

The high prevalence of positive Leptocybe inspections after a drought period (seen in the Time-series K-means Analysis approach) is as a result of only a few Leptocybe inspections. We suggest that the model is applied to additional Leptocybe data collected in years to come to investigate the potential trend.

Furthermore, all models struggle to accurately predict the majority class, which was done by design to prevent type II errors, however it leaves a gap for improvement in future models.

3. REFERENCES

[1] Yu, B. and Kumbier, K., 2020. Veridical data science. *Proceedings of the National Academy of Sciences*, *117*(8), pp.3920-3929.

[2] Weisstein, Eric W. "Voronoi Diagram." From MathWorld--A Wolfram Web Resource. https://mathworld.wolfram.com/VoronoiDiagram.htm

[3] Germishuizen, I., personal communication, April 21, 2022.

[4] Aurenhammer, F., 1991. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Computing Surveys (CSUR)*, *23*(3), pp.345-405.

[5] Li, Z., 2011. *Applications of Gaussian mixture model to weather observations*. The University of Oklahoma.

[6] EPSG.io, 2015. EPSG:3857. Available from: https://epsg.io/3857

[7] Wu, J. 2012. Advances in K-means clustering: A Data Mining Thinking. Springer Publishing Company, Incorporated