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

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

ABSTRACT

This project analyses weather and pest-prevalence data in South Africa to identify possible relationships between changing climatic patterns and pest populations using machine learning techniques. Three predominant datasets were received from the Forestry and Agricultural Biotechnology Institute (FABI) for contribution to this project, namely: temperature and rainfall records of roughly 6000 weather stations across South Africa, Sirex noctilio (Sirex) pest inspections, and Leptocybe invasa (Leptocybe) pest inspections. The weather and pest datasets provided had no linking attributes other than spatial references. The data was processed using a feature engineering algorithm to link the weather and pest datasets and prepare the data for modelling. The engineered parameters were then used to train XGBoost and Support Vector Machine (SVM) models. Thereafter, the models were tested for stability and prepared for deployment. Deployment allows the user to input weather parameters to determine a pest presence prediction probability at the user-defined location and across South Africa. The XGBoost model achieves 81% accuracy for Leptocybe and 64% for Sirex. XGBoost performs 10-15% better than the SVM when classifying both pests. However, both models classify Leptocybe more accurately than Sirex.

Keywords

Sirex noctilio, Leptocybe invasa, Forestry pest and disease management

1. INTRODUCTION

Over recent years, the South African forestry industry has been under threat from invasive pests. In this research, we focus on the Sirex noctilio and Leptocybe invasa pests, which target pine and blue gum trees. We partnered with the Forestry Agriculture and Biodiversity Institute (FABI), which provided us with the necessary data for this project:

1. Weather records from 1950-2020
2. Pest inspection records from 2012-2019

Sirex noctilio (shown in Figure 1) is a fairly large wasp (up to 4 cm long) which bores into the trunk of pine trees to lay eggs. The eggs are deposited with a mucoid substance which is toxic to trees. This substance ultimately leads to a decline in the tree's health. Leptocybe invasa (Figure 2) is a small black wasp around 1 mm in length. These wasps create abnormal growths known as galls on young blue gum trees' leaves, petioles and stems. Heavily infected trees may experience stunted growth or even death.

1.1 Problem Statement

The forestry industry in South Africa employs over 500 000 people and contributes to roughly 10% of the country's agricultural Gross Domestic Product (GDP) and 5% of the manufacturing GDP. The industry's export value is close to R40 billion, making it a key contributor to the South African economy [1]. However, the Sirex and Leptocybe pests adversely affect the timber yields on various plantations. We aim to develop a machine learning tool to identify high-risk areas so that the forestry industry can monitor and fight pest prevalence in these regions before pest levels are uncontrollable.

Figure 1: Sirex noctilio [2]

Figure 2: Leptocybe invasa [3]

2. LITERATURE REVIEW

2.1 Nearest Site Problems

One of the first challenges faced in this project was linking pest inspections to weather records to identify relationships with machine learning models. Weather stations and pest inspection records had geographic coordinates associated with their location. Unfortunately, these do not overlap, making it difficult to identify which weather conditions were experienced at a given location linked to a pest inspection. To address this issue, we use a concept from computational geometry called Voronoi diagrams.

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"* [4]. 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 [5].

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 3 visualises this concept.

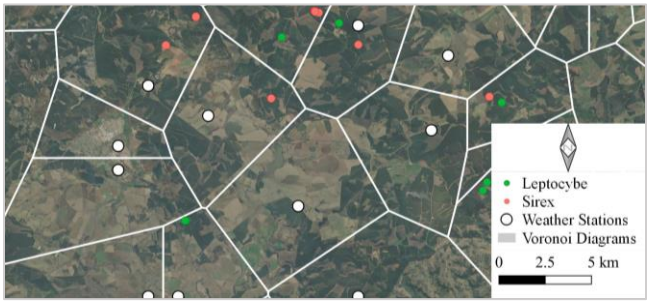

Figure 3: Voronoi Diagram

2.2 Gaussian Mixture Models

Gaussian mixture models (GMM) are a semi-parametric, unsupervised technique 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 several fields, including weather data analysis [6].

2.3 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 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.

2.4 XGBoost

XGBoost is a decision-tree-based ensemble Machine Learning algorithm developed at the University of Washington in 2016 [8]. This algorithm was designed with time and memory constraints in mind, leading to highly efficient modelling and training. It has been cited in several fields, such as medical imaging diagnosis, financial risk assessment and metagenomics classification [9]. However, the standard implementation of the XGBoost algorithm has been known to struggle with imbalanced classification [9]. Therefore, to model imbalanced data such as pest prevalence, a penalisation factor needs to be introduced for misclassification of the minority class.

2.5 Support Vector Machines

Support vector classifiers are based on a principle known as margin maximisation. These classifiers have seen great success in the industry as they have a wide array of parameters that allow them to model both linearly and non-linearly separable data [10,11]. This can be achieved by using a radial basis kernel to map the data onto a higher dimensional feature space that is linearly separable. Figure 4 and Figure 5 illustrate this concept to aid the reader's understanding. Furthermore, we can change the strictness of the margin and create what is known as a "soft margin", which allows the model to intentionally misclassify certain records for a gain in overall classification accuracy.

An empirical comparison between support vector classifiers and artificial neural networks found that neural networks only marginally outperformed support vector classifiers with the added cost of complexity. Furthermore, support vector classifiers were still able to process noisy data better than the neural networks due to the soft margin parameter, which allows for outliers to be ignored based on a user-defined threshold [12, 13].

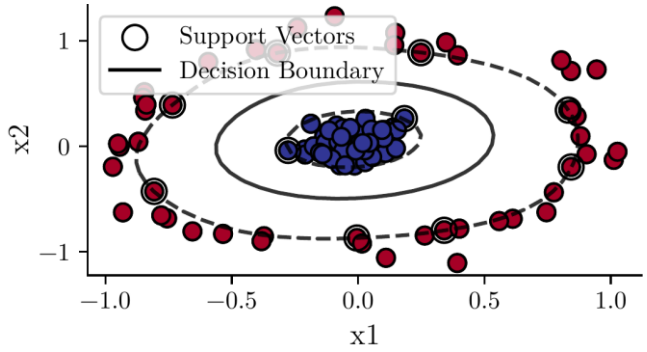

Figure 4: Support Vector Classifier ~ Non-linearly Separable Data

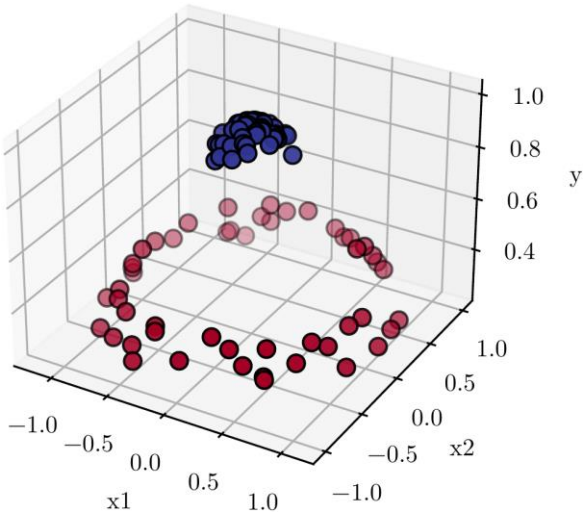

Figure 5: Mapping data onto higher dimension

3. EXPLORATORY ANALYSIS

3.1 Weather Dataset

This data consists of approximately 107 million daily temperature and rainfall readings from 6137 weather stations scattered across South Africa. The data were collected over 70 years between 1950 and mid-2019. There are three types of stations, namely: RAIN, TEMP and TEMP_RAIN, which indicate the type of data collected by the station. The proportion in which the stations were distributed was relatively even until 2001 when government funding was reduced. This led to a drastic decline in the number of stations and a shift in the proportion of station types, as seen in Figure 6 and Figure 7.

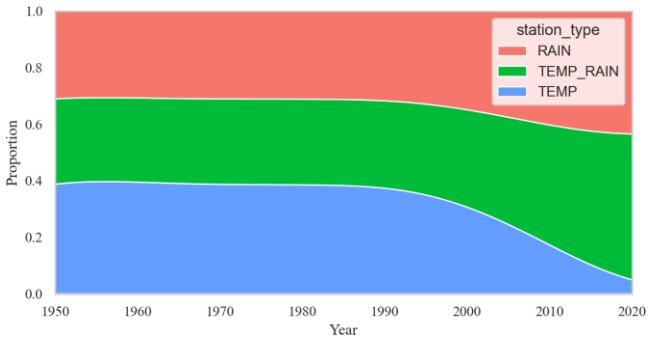

Figure 6: Station Type Distribution

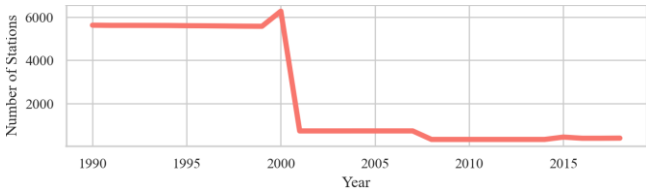

Figure 7: Number of Stations per Year

This decrease in stations coincides with a period of missing data for both maximum and minimum daily temperatures between 2001 and 2004, as shown in Figure 8, which resulted in null value temperatures for 2.89% of the total temperature data.

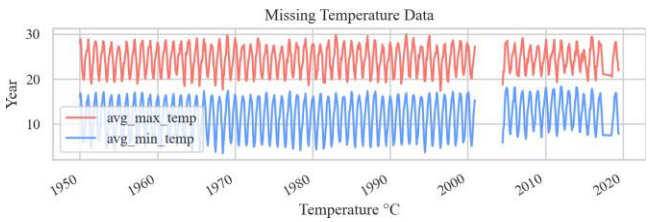

Figure 8: Average Monthly Temperature

The rainfall data only had a small percentage of null values (approximately 0.08%). However, a faulty station was detected due to its annual rainfall exceeding 150000 mm for seven consecutive years. Considering that the highest average rainfall for any region in South Africa is around 3000 mm annually [14], this station was subsequently filtered out of the dataset as it drastically impacted any statistical metrics derived from the data.

The remaining records were visualised with box plots in Figure 9 to highlight the distribution of annual rainfall readings per decade. There is a consistency in the readings per decade in terms of magnitude. However, changes in the median and interquartile ranges indicate movement in the data and potential underlying trends in annual rainfall patterns.

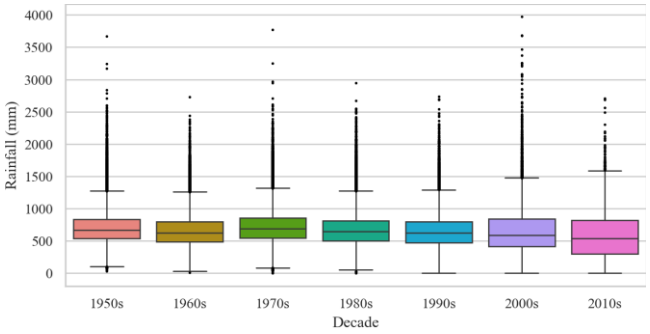

Figure 9: Annual Rainfall Per Station Grouped by Decade

3.2 Sirex inspection samples

The Sirex dataset includes 3780 inspections taken across five South African provinces. An average of 473 inspections are recorded annually, with approximately 24% of inspections indicating a positive pest finding. The attributes of note include: location, pest presence, severity, and affected tree species.

3.2.1 GPS inspection location

The dataset provides approximate centroid coordinates of plantation compartments in which inspections were performed. Several errors - including interchanged latitude and longitude coordinates, null coordinates and GPS inaccuracies - are seen in the data. However, inference can be made to correct faulty coordinates with site numbers and compartments.

3.2.2 Sirex presence binary indicator

The binary indicator represents the outcome of an inspection, with 920 inspections indicating a positive Sirex presence. Figure 10 displays the spatial distribution and inspection results.

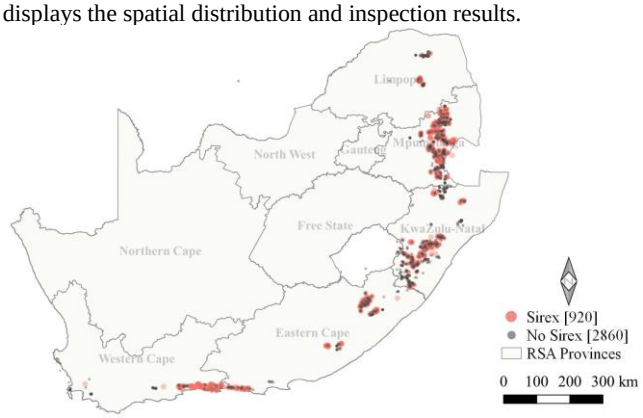

Figure 10: Sirex inspection distribution

3.2.3 Condition of trees and pest severity

Table 1 indicates the condition of the trees inspected. The data indicates that only the samples with a Sirex positive result have trees that have been declared dead or dying. However, the high percentage of unclassified stems provides uncertainty in the analysis of living versus dead or dying stems due to the Sirex pest. Consequently, the severity of the Sirex pest is poorly represented by the condition of trees within the sample inspections.

Table 1: Sirex condition status

Presence	Living	Dead/dying	Unclassified	Total stems
No Sirex	88.7%	0.0%	11.3%	**100%**
Sirex	88.1%	3.5%	8.5%	**100%**

3.2.4 Tree species and Sirex prevalence

Figure 11 provides the Sirex prevalence per pine tree species. The three predominant species are: P PAT (*Pinus patula*), P RAD (*Pinus radiata*) and P ELL (*Pinus elliottii*), with the P PAT and P RAD species having a higher positive Sirex prevalence.

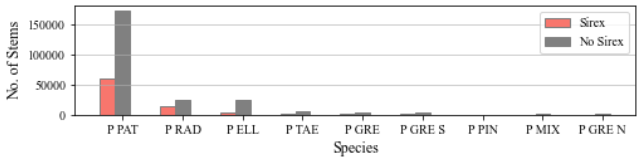

Figure 11: Sirex prevalence per tree species

3.3 Leptocybe inspection samples

The Leptocybe dataset includes 747 inspections, concentrated in four South African provinces. Inspections vary largely per year, from a minimum of 46 inspections in 2017 to a maximum of 187 inspections in 2019. Approximately 40% of inspections indicate a positive finding of the pest.

The dataset attributes are similar to the Sirex data. However, GPS coordinates are more consistent and pest severity more detailed.

3.3.1 GPS inspection location and pest presence

Figure 12 displays the distribution of Leptocybe inspections, with 314 inspections indicating a positive Leptocybe presence. The number of inspections is fewer than the Sirex inspections. However, inspections overlap in multiple eastern plantations. The binary pest indicator and pest GPS location are corresponding attributes between pest datasets.

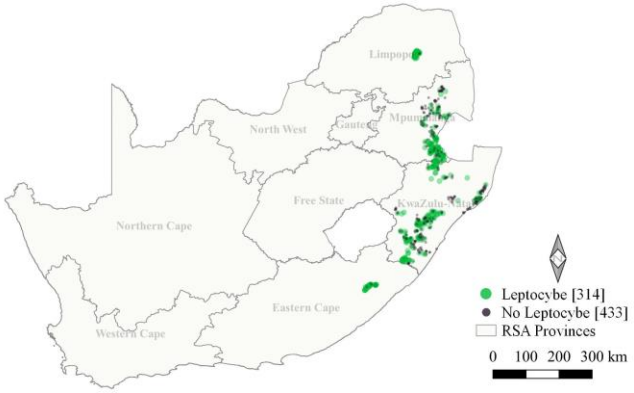

Figure 12: Leptocybe inspection distribution

3.3.2 Leptocybe intensity

The number of trees infected by the pest and the total number of trees inspected per inspection are used to determine the intensity of the pest infestation. Figure 13 indicates that infestation levels were highest in 2017, following a higher rainfall season in 2016. Therefore, there is preliminary evidence to suggest that pest intensity is dependent on rainfall patterns.

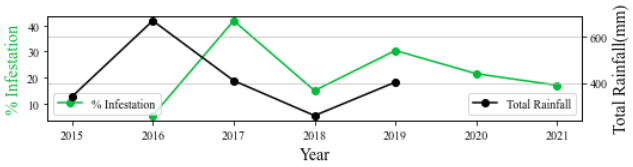

Figure 13: Leptocybe intensity vs total rainfall

3.3.3 Tree species and Leptocybe prevalence

Figure 14 shows Leptocybe prevalence per Eucalyptus (E.) species. Through considering the four predominant species, Leptocybe is most prevalent in EGXN (*E. grandis x E. nitens*) and EGRA (*E. grandis*), and least in EGXU (*E. grandis x E. urophylla hybrid*) and EDUN (*E. dunnii*), suggesting that certain species are more prone to pest infestation than others.

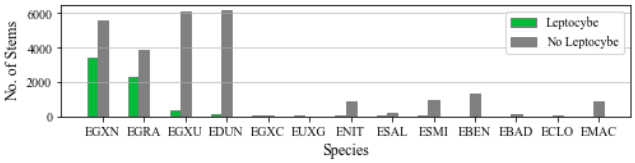

Figure 14: Leptocybe prevalence per tree species

3.4 Data integration

Comparison of the three datasets indicates that the pest inspection data (recorded annually) does not have the same level of granularity as the weather data (recorded daily). This complicates identifying relationships between weather conditions and pest prevalence due to the absence of seasonal differences in pest populations. However, the preliminary analysis outlined in Section 3.3.2 above indicates that possible trends may be identified (albeit the difference in granularity) by considering, for example, a change from normal conditions.

The 'Nearest Site' problem discussed in Section 2.1 suggests that the weather experienced at a pest site can be derived from the nearest active weather station. The proximity of pest inspections to weather stations is shown in Figure 15 and Figure 16, highlighting that most pest inspections are within 10 km of a weather station. This nearness is deemed sufficient to reflect the approximate weather conditions at pest sites.

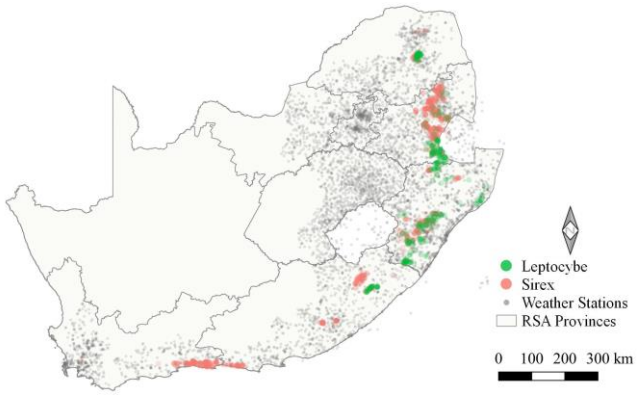

Figure 15: Weather Station Distribution vs Pest Distribution

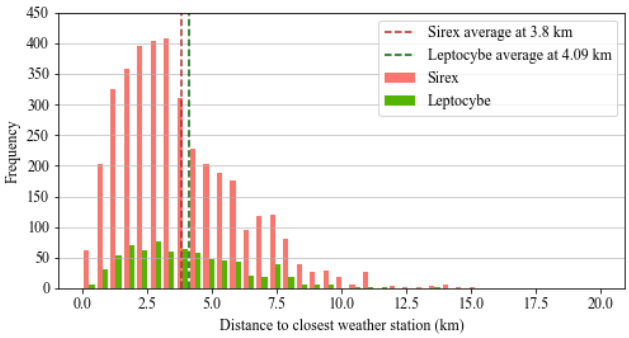

Figure 16: Weather Station Distribution vs Pest Distribution

4. MODELLING AND VISUALISATION

4.1 Feature Engineering

Section 4.1 provides the algorithm and assumptions made to interface the weather data with the pest data and prepare the data for modelling.

4.1.1 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 [15] and excluded from the 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 monthly. Additionally, the daily temperature difference is calculated and averaged per month. This results in four weather measurements for analysis: *rainfall*, *temperature_max*, *temperature_min*, *and temperature_diff*.

4.1.2 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 [16]) to allow for Euclidean distance measurement in metres between station and pest coordinates. The algorithm identifies the closest active station per pest inspection, measurement type (rainfall, maximum and minimum temperature), and month-year period. This analysis ensures that if a station is inactive during a period, the Voronoi diagram is redrawn, and the next closest station is assigned to provide weather measurements to a pest inspection location.

Table 2, Table 3, and Figure 17 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 before 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 2: 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 3: 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

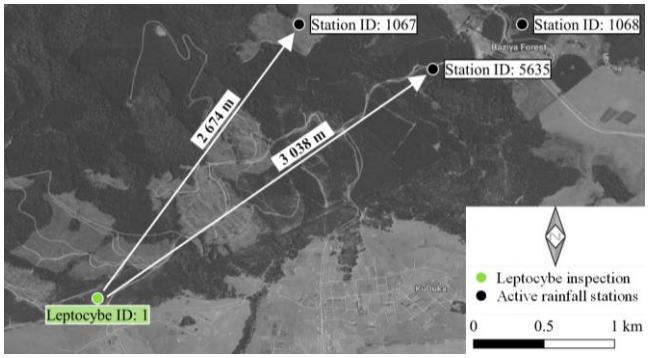

Figure 17: Pest to station mapping

4.1.3 Step 3) Calculate average readings per month

Using the aggregate readings (Step 1) derived from the active stations per month and 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*.

4.1.4 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 before and during the inspection year. This results in 48 data points per pest ID, that is, one reading per month. The time series 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*.

4.1.5 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*.

4.1.6 *Step 6) Format data for modelling*

Table 4 provides an extract of the output of the feature engineering stages described in Steps 1 to 5 above. The *temparature_min* and *temperature_diff* columns are appended to the right of Table 4, with the remaining pests IDs and timeframes below Table 4. Advantages of the data in this form include:

- Allows for adding 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, time-series modelling, and supervised methods.

Table 4: 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

4.2 Unsupervised Methods

Section 4.2 discusses the algorithms that are primarily used to cluster the data points based on similarity. An initial hypothesis suggests that certain clusters may have higher pest prevalence, which allows for investigation into a cluster's defining characteristics as a potential cause of elevated pest presence.

4.2.1 *Gaussian Mixture Model*

Table 5 shows the variation of percentage positive cases across the GMM 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. One cluster contained ~90% of the data in this specific case. 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 5: 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

4.2.2 *Time-series K-means Analysis*

Time-series K-means Analysis is selected to allow for explainable results (as discussed in Section 2.3). Considering this, the number of parameters for perturbation two is deemed too complex to explain clustering results. Therefore, combinations of perturbation three are explored. Table 6 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 6: 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 18 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). Cluster 1, the blue cluster, contains ~20% of the data. However, it contains ~45% of all positive Leptocybe inspections. The highlighted section in Figure 18 indicates a definite lower rainfall region than normal conditions.

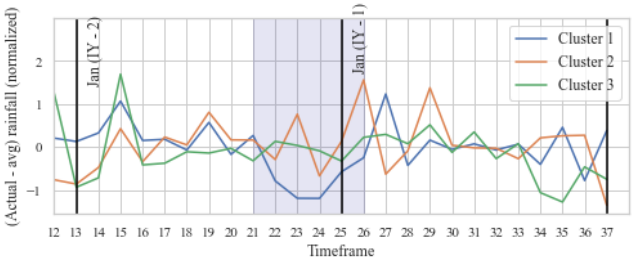

Figure 18: Leptocybe cluster centroids

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 separating positive and negative Sirex inspections. All clusters, on average, with varying K, show ~30% positive Sirex inspections, inhibiting 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 18 indicates potential for future drought and pest presence modelling.

4.3 Supervised Methods

This section investigates two supervised classifiers. The task is sensitive to type II errors (false negative). Therefore, we aim to achieve a high classification accuracy for the minority class.

4.3.1 *Support Vector Machine*

As discussed in Section 2.5, SVMs can classify non-linearly separable data using soft margins and kernel methods. This is the primary reason for using this algorithm to model our data. The final model uses a radial basis function kernel, which consists of two parameters, namely the tolerance (C) and the influence (ɣ). 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 7 shows the model's performance 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 modelled scenario

is susceptible to type II errors. Table 7 highlights the model's poor performance when predicting the majority class, especially with the Sirex pest.

Table 7: Classification Accuracy of SVM model

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

4.3.2 XGBoost

XGBoost is a relatively recent algorithm that makes use of gradient boosted decision trees. 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 8 details the results of this model. The results indicate that XGBoost performed significantly better than SVM when classifying both pests due to the higher AUC score for the XGBoost in both perturbations of the dataset. Furthermore, the model runs significantly faster than the SVM model.

Table 8: Classification Accuracy of XGBoost model

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

4.4 Visualisation

The visualisations provide both a historical and future perspective of the effect of changing weather patterns and pest prevalence across South Africa. The historical perspective allows the user to analyse trends that may have led to a positive pest inspection. While the future perspective allows the user to extrapolate predictions across South Africa, including inaccessible regions covered in dense forest.

All visualisations included in this report are dynamic (subject to user input) and interactive through the *streamlit* web application built for this project. The web application has four main pages: *Home*, *Prediction*, *Visualisation*, and *Upload* - which can be accessed through the GitHub Repository.

4.4.1 Historical Visualisation

4.4.1.1 Pest presence

Figure 19 is a snapshot of the historical presence of Sirex, which is deployed on the *Home* page. A similar map is also available for Leptocybe. The visualisations allow the user to select '*All'* years of inspection, or a single year, to perform their own exploratory data analysis for pest prevalence. Additionally, the visualisations spatially display the pest data used to build the models.

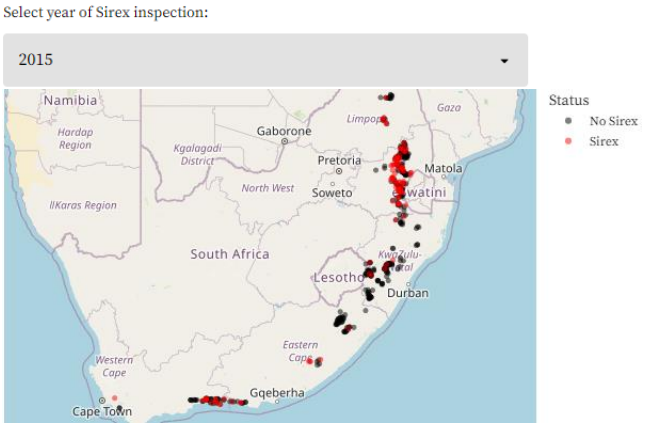

Figure 19: Sirex presence in 2015

4.4.1.2 Weather measurements

Figure 20 and Figure 21 are snapshots of the historical total rainfall and average temperatures, which are deployed on the *Visualisation* page of the web application. However, to generate the graphs, the user must enter the selected month, total monthly rainfall, average maximum and minimum temperatures for the month, and an approximate location of where the measurements were taken through the web application interface. After running the model, the feature engineering script (similar to that explained in Section 4.1) is invoked to find the nearest active weather stations for the point location provided. The script retrieves the historical rainfall and temperature measurements from the relevant stations. Note that the average measurements are shown in Table 9 (discussed in Section 4.4.3.1) are calculated using the results of this analysis.

Figure 20 and Figure 21 are then generated, which reflect the weather measurements specific to the location provided by the user. Additionally, the graphs allow the user to graphically view the change in rainfall and temperature trends compared to the actual measurements for the provided location.

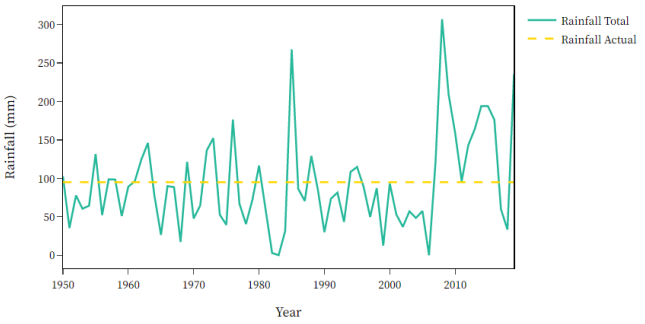

Figure 20: Rainfall trends for February at selected location

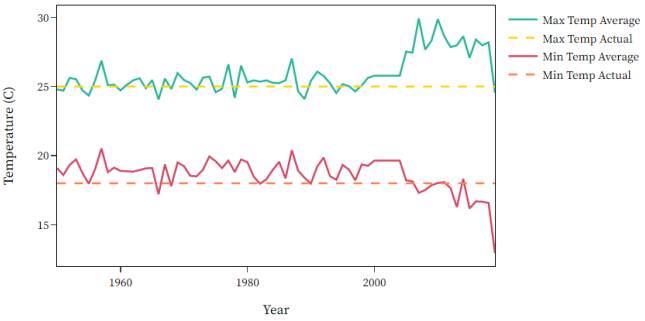

Figure 21: Temperature trends for February at location

4.4.1.3 *Active weather stations*

Figure 22 displays the distance of the nearest active weather stations to the user-defined location from 1950 to 2019. This visualisation explains to the user that active stations change monthly and yearly. Only stations with valid and complete measurements are used to determine the expected weather conditions at a location.

Additionally, the distance of the station to the user-defined location provides a qualitative indication of the applicability of the measurement. Note that a quantitative confidence analysis is futile given that other environmental factors, such as altitude, humidity, and wind, will affect local weather conditions.

Figure 22 shows that, generally, the stations used for rainfall measurements are closer to the point of interest than the stations used for temperature measurements. Additionally, the stations providing the measurements for rainfall (Station ID: 1012 at 20 km from the location) and temperature (Station ID: 1021 at 61 km from the location) are consistent before 2001. Thereafter, the active stations change, indicating that the prior stations were either decommissioned or complete data is unavailable for the subsequent period.

Additionally, Station ID: 5635 is inactive in 2018, thus using the rainfall readings at Station ID: 5975 for the year, but active again in 2019, therefore, defaulting to the readings of the nearest active station (that is, Station ID: 5635). Overall, Figure 22 indicates that the user-defined location uses data from 6 unique stations throughout the period of consideration, with the furthest station located 109 km away from the point of inspection (Station ID: 5695 in 2019).

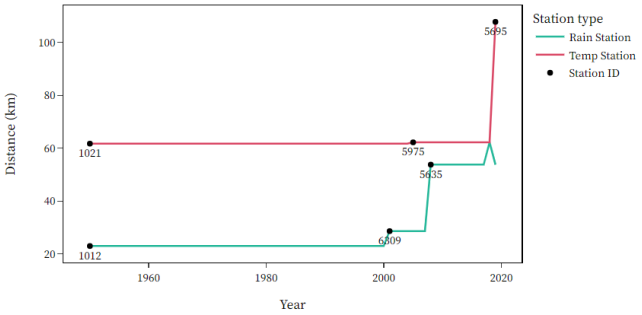

Figure 22: Distance from user-defined location to the nearest active station per year

Figure 23 spatially displays the stations identified in the analysis above against the user-defined location. The visualisation aims to show the proximity of the weather stations from Figure 22 above to the user-defined location. The map allows the user to hover over the station or inspection point (that is, the user-defined point) to show the Station ID and the straight-line distance to the inspection point. Figure 23 shows this feature for Station ID: 1021 (as discussed in Figure 22 above), located approximately 61 km from the user-defined location.

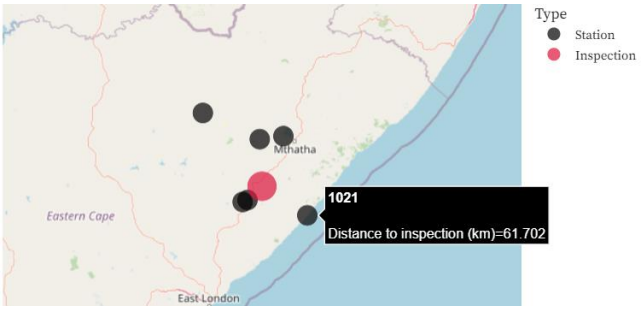

Figure 23: Nearest active stations

4.4.2 *Model Visualisation*

When navigating to the *Predictions* page, the user is presented with inputs such as rainfall, maximum temperature, minimum temperature, longitude, latitude, and a map that shows the point location of the provided longitude and latitude. The user can select one of two pre-trained models, either XGBoost or SVM, to reflect the selection of the two best-performing models from Section 4.3. The inputs are fed into a feature engineering function (similar to that explained in Section 4.1) that transforms the inputs into 13 unique features. These features as inputs to predict both pests simultaneously. It is important to note that each pest has a dedicated model, so when a user selects XGBoost, two pre-trained XGBoost models will be loaded from files, one for the Leptocybe pest and one for the Sirex pest.

Once both models have finished classifying the input data, the final prediction is printed on-screen with its respective class probabilities, shown in Figure 24. This provides context behind predictions and allows the user to make their own judgement on borderline cases. Additionally, since not all app users will have a technical background, we briefly explain the chosen algorithm (on the application) and write out the final prediction with its probability. For example, *The model predicted a positive Leptocybe inspection with an 82% probability.*

Figure 24: Prediction results for Leptocybe scenario

Lastly, the user is shown the ROC-AUC curve of the chosen algorithm trained on each pest (as in Figure 25). The concepts behind the ROC-AUC curve are explained with simple terminology so the user can compare algorithms and their ability to predict pest prevalence across the two pest species.

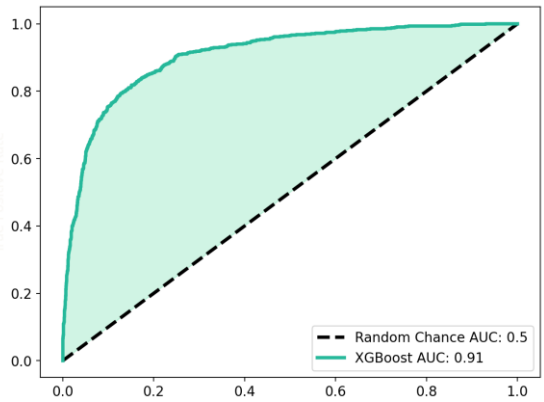

Figure 25: ROC-AUC curve of XGBoost model for Leptocybe

4.4.3 Future Visualisation

4.4.3.1 At the user-defined point

Table 9 provides a snapshot of the table on the *Visualisation* page. The table displays the user's input measurements (actuals) on the *Prediction* page and the average monthly measurements for the user-defined location from 1950 to 2019. The table also indicates the difference between the actual and the average measurements. The table compares the provided measurements, and the average typically experienced for the user-defined location (as derived from the analysis in Section 4.4.1.2).

All parameters shown in the table, with the addition of the timeframe parameter, comprise the input parameters for the prediction model – where the timeframe parameter is calculated using the month and a prediction period provided by the user. Note that this transformation of parameters follows the same logic as the feature engineering process used to prepare the data to train the models (defined in Section 4.1).

Table 9: Example of a set of user input parameters

	Total rainfall (mm)	Average Maximum Temperature (°C)	Average Minimum Temperature (°C)
Input measurements	95.0	25.0	18.0
Average for location	92.6	25.9	18.6
Difference	2.4	-0.9	-0.6

Figure 26 provides a snapshot of the model results. This visualisation repeats the results from the *Prediction* page (discussed in Section 4.4.2 above) to allow the user to compare the results at the user-defined location to the predicted results across South Africa (further discussed in Section 4.4.3.2).

Figure 26: Repeat of prediction results

4.4.3.2 Extrapolation across South Africa

Figure 27 appears on the *Visualisation* page and shows the initial user-defined location and the positive prediction probabilities if the input measurements provided on the *Prediction* page are experienced across South Africa. A similar map is also available for Sirex on the *Visualisation* page. This visualisation enables the user to compare the prediction probability at the user-defined point and the prediction probability across the country if the same weather conditions are experienced.

Note that Figure 27 only shows the positive probabilities. For example, the probability of the grid block containing the user-defined point used in Table 9 is approximately 81%. Each grid block shown in Figure 27 has dimensions of 50km by 50km, with 689 grid blocks used in the visualisation. Each grid block centroid represents the location passed to the feature engineering script (as explained in Section 4.1). The spatial representation is produced using the following process:

1. The nearest active stations per centroid are identified, and their values are used to determine averages per point.
2. The input measurements, averages (calculated above), and the timeframe parameter are used in the pre-trained models for Leptocybe and Sirex to determine the prediction per centroid and hence per grid block.

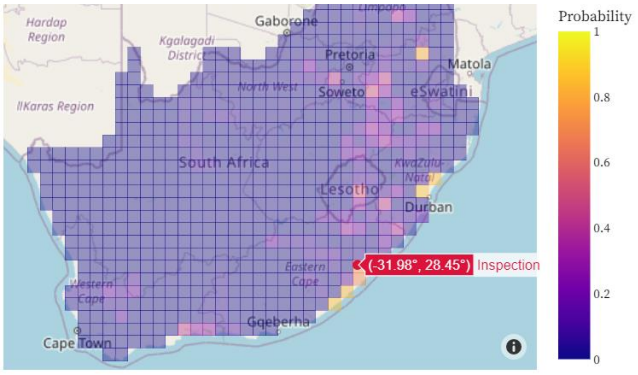

Figure 27: Positive probability of positive Leptocybe inspection for given measurements

5. PCS DISCUSSION

5.1 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 10 and Table 11 indicate the fold accuracies for Leptocybe and Sirex, respectively. The fold accuracies have a very low standard deviation, indicating a high degree of model stability. Additionally, 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 10: 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 11: 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%

5.2 Model Repeatability

To ensure repeatability, the data should be subjected to the same feature engineering steps detailed in Section 4.1. 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, the GitHub Repository includes all code used to generate the models and visualisations for this project, with the README file detailing all necessary packages and their versions, as well as the code for this research.

5.3 Model Evaluation

Ultimately, the models are evaluated on their ability to accurately classify both classes using ROC-AUC curves. Figure 28 and Figure 29 show the ROC-AUC curves for Leptocybe and Sirex, respectively. In general, the models can classify Leptocybe more accurately than Sirex. This finding is evident in the ROC-AUC and the class accuracies previously detailed in Table 7 and Table 8. The classification accuracy of negative Sirex inspections is especially poor, between 38% and 57% for all models across both perturbations. Models 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 five features.

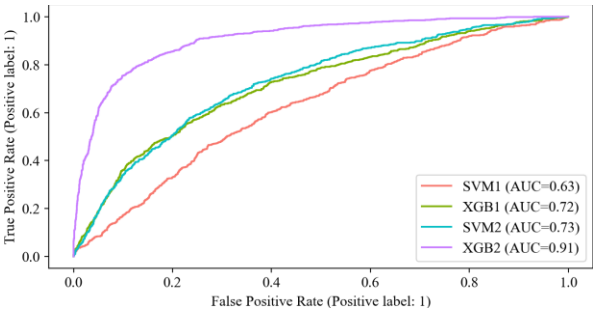

Figure 28: Leptocybe Classification Performance

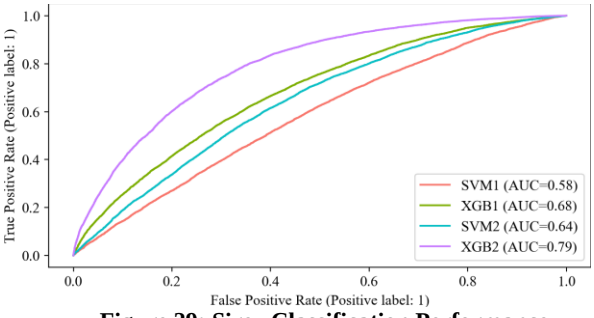

Figure 29: Sirex Classification Performance

6. CONCLUSION

This research task aims to link Sirex noctilio and Leptocybe invasa inspection to historical weather records and use machine learning modelling techniques to uncover relationships between pest prevalence and changing climatic conditions in South Africa. The data was pre-processed using a feature engineering algorithm. Thereafter, the engineered parameters were used to train several models, with XGBoost and SVM reflecting better performance. The XGBoost model achieves 81% accuracy for Leptocybe and 64% accuracy for Sirex. XGBoost performs 10-15% better than the SVM when classifying both pests. However, both models classify Leptocybe more accurately than Sirex.

7. REFERENCES

[1] Forestry South Africa. "Forestry Explained: Our Economic Contribution." forestrysouthafrica.co.za. https://www.forestrysouthafrica.co.za/economic-contribution/ (accessed Apr. 2, 2022).

[2] FABI. "Sirex Woodwasp." fabinet.up.ac.za. https://www.fabinet.up.ac.za/index.php/tpcp/forest-threats/sirex-noctilio (accessed Apr. 20, 2022).

[3] FABI. "Blue gum chalcid." fabinet.up.ac.za. https://www.fabinet.up.ac.za/index.php/tpcp/forest-threats/leptocybe-invasa (accessed Apr. 20, 2022).

[4] E.W. Weisstein. "Voronoi Diagram." mathworld.wolfram.com. https://mathworld.wolfram.com/VoronoiDiagram.htm (accessed Mar. 13, 2022).

[5] F. Aurenhammer, "Voronoi diagrams—a survey of a fundamental geometric data structure," *ACM Computing Surveys (CSUR)*, vol. 23, no. 3, pp. 345-405, 1991.

[6] Z. Li, "Applications of Gaussian mixture model to weather observations," Ph.D. dissertation, School of Elec. and Comp. Eng., Univ. Oklahoma, Oklahoma, USA, 2011.

[7] J. Wu, *Advances in K-means clustering: A Data Mining Thinking*, 1st ed. Berlin, Heidelberg: Springer Publishing Company, Inc., 2012.

[8] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proc. 22nd acm sigkdd int. conf. on knowledge discovery and data mining*, 2016, pp. 785-794.

[9] C. Wang, C. Deng and S. Wang, "Imbalance-XGBoost: Leveraging weighted and focal losses for binary label-imbalanced classification with XGBoost," *Pattern Recognition Letters*, vol. 136, pp.190-197, 2020.

[10] T. Mullen and N. Collier, "Sentiment analysis using support vector machines with diverse information sources," in *Proc. 2004 conf. on empirical methods in natural language processing*, Jul. 2004, pp. 412-418.

[11] L. Chen. "Support Vector Machine — Simply Explained." towardsdatascience.com. https://towardsdatascience.com/support-vector-machine-simply-explained-fee28eba5496 (accessed Apr. 30, 2022).

[12] R. Moraes, J.E. Valiati and W.P.G. Neto, "Document-level sentiment classification: An empirical comparison between SVM and ANN," *Expert Systems with Applications*, vol. 40, no. 2, pp.621-633, 2013.

[13] M. Ahmad, S. Aftab, S.S. Muhammad and S. Ahmad, "Machine learning techniques for sentiment analysis: A review," *Int. J. Multidiscip. Sci. Eng*, vol. 8, no.3, p.27, 2017.

[14] Water Research Commission. "Rainfall in South Africa." southafrica.co.za. https://southafrica.co.za/rainfall-south-africa.html (accessed Apr. 30, 2022).

[15] I. Germishuizen, private communication, Apr. 21, 2022.

[16] EPSG.io. "EPSG:3857." epsg.io. https://epsg.io/3857 (accessed Apr. 21, 2022).