UNIVERSITY OF CAPE TOWN

MASTERS THESIS

Predicting anomalous weather events using supervised machine learning

---

Author:

Edwina Williams

Supervisor:

A/Prof. Deshen Moodley

A thesis submitted in fulfilment of the requirements for the degree of Master of Philosophy in Information Technology in the

Department of Computer Science

26 October 2022

Declarations

I, Edwina Gillian Williams, hereby declare that the work on which this dissertation/thesis is based, is my original work (except where acknowledgements indicate otherwise) and that neither the whole work nor any part of it has been, is being, or is to be submitted for another degree in this or any other university.

Signature: Signed by candidate ………………………………………….

Date: 13 February 2022

Abstract

The complexity and variability of atmospheric processes make it difficult to predict weather anomalies. Early detection of weather anomalies is critical to ensure that the necessary precautions are taken to limit the impact on people and economic activities. There is a growing interest in the use of machine learning techniques as an alternative to traditional weather forecasting methods. In this study, the use of machine learning techniques to predict daily maximum temperatures and detect temperature anomalies is investigated. Machine learning techniques were trained to predict weather anomalies for three stations in the Gauteng and Northern Cape provinces of South Africa. Three machine learning techniques were selected based on their use and performance in the relevant literature. The techniques include the Support Vector Machine, Artificial Neural Network and Huber Regressor. Both regression and classification-based techniques were evaluated and compared to determine which provide optimal performance for predicting temperatures and detecting anomalies. The regression-based techniques were trained to predict the daily maximum temperatures (for the next day) based on the previous three day's conditions. The predictions were evaluated based on the next day prediction error and the anomaly detection rate in the predictions. Techniques based on classification were trained to classify whether an anomaly would occur the next day based on the previous three day's conditions. The results showed that the machine learning techniques performed well at predicting the next day's maximum temperatures. However, the techniques had a low success rate in detecting anomalies.

Table of Contents

List of Figures

List of Tables

Glossary of Terms

Term	Meaning
SVR	Support Vector Regression
SVC	Support Vector Classification
ANN	Artificial Neural Network
Feature	A property of an object in the dataset.
Validation data	A portion of the dataset that is kept aside and used to assess the performance of the machine learning technique.
Training data	A portion of the dataset that is used to train the machine learning technique.
Climate baseline	The average expected climate for a region.
Weather observations	Observed recordings of atmospheric conditions at a specific location and time.
SAWS	South African Weather Services
NCEI	National Centre for Environmental Information

Acknowledgements

My first acknowledgment goes to God, my Creator, for His provision and for giving me the strength and wisdom to continue and complete this work. Second, I would like to emphasize my gratitude to A/Prof. Deshen Moodley for his mentoring and guidance throughout the process. Your contribution has shaped me and expanded my field of knowledge. Through your support and guidance, I have learned a lot over the past years. Finally, I thank my family for their constant support and encouragement.

Chapter 1

1. Introduction

Climate data show an increase in the frequency, duration, and intensity of abnormal weather events. Temperature extremes such as heat waves are associated with increased mortality [2, 3] and pose a threat to food security [4]. A temperature anomaly is defined as an observation that deviates from the climate norm (also called the baseline) [3, 4]. The baseline defines what the normal or expected conditions are at a particular location. A temperature extreme, such as a heat wave, is defined as anomalous conditions that persist for an extended period of time [5, 6]. Timely prediction of anomalous temperature conditions would allow for early response to minimize potential impacts. However, predicting weather anomalies remains a difficult task due to the chaotic and volatile dynamics of the atmosphere [5]. Traditional forecast models use an understanding of physics and fluid dynamics to model the state of the atmosphere based on current weather conditions [6]. Any sudden perturbation in these conditions affects the models' ability to reliably stimulate the probabilities of anomalous weather conditions [6, 7]. Machine learning techniques are robust to such perturbations [7] and could provide a viable alternative for predicting weather anomalies.

A time series is a collection of observations recorded at specific times. An example of such a dataset is meteorological observations. In related work, machine learning has been most commonly used to predict precipitation [3, 8-14] while few studies have focused on predicting temperatures [15-19]. Some of the widely used machine learning techniques in weather prediction studies are the Artificial Neural Network (ANN) [3, 8, 10, 13-19] and, the Support Vector Machine (SVM) [8, 9, 11, 12, 14, 18-20]. Regression-based techniques were most used over classification-based solutions [8, 9]. Metrics measuring accuracy such as the Mean Absolute Error (MAE) [3, 12, 13], Mean Square Error (MSE) [3, 10, 15] and Root Mean Squared Error (RMSE) [3, 13] are most commonly used for regression, while Precision and Recall were used for classification-based solutions [8, 9]. Techniques used for evaluation include the hold-out validation [9-11, 13-15] and the k-fold cross validation, while some studies used techniques such as leave-one-station-out cross-validation and hindcast validation [17]. Some of the identified data pre-processing tasks include deletion [8] or interpolation [15, 17, 19] to deal with missing values and the sliding window technique [9, 11, 18] to prepare the data for machine learning. The studies found that machine learning is suitable for predicting atmospheric variables.

In this study, the use of machine learning techniques to predict the daily maximum temperatures are investigated, to evaluate which technique is best able to detect temperature anomalies.

1.1 Aims and Objectives

The aim of this study is to explore machine learning techniques for daily temperature prediction and temperature anomaly detection

The objectives of the research are to:

- Review and evaluate machine learning techniques used for weather prediction in related work.
- Evaluate the performance of the Support Vector Regressor, Artificial Neural Networks and Huber Regressor for next day temperature prediction and anomaly detection at three different weather stations in South Africa to determine the state of the art for weather prediction and anomaly detection.
- Analyse the influence of nearby stations on the performance of machine learning techniques.

1.2 Tools and Approach

1.2.1 Overall Approach

In this study, machine learning techniques that have been shown to be successful in related work were selected to evaluate and compare their performance for predicting daily temperatures and detecting temperature anomalies. The machine learning techniques include the Support Vector Machine, Artificial Neural Network, and linear regression techniques. This study will perform a review of different types of techniques to evaluate if the spatial attributes of the timeseries affect how well anomalies are predicted. In accordance with the World Meteorological Organization’s (WMO) definition of anomalies, anomalies are defined in this study as observations that deviate more than two standard deviations from the climate baseline.

1.2.2 Datasets

Datasets used for the experiments were obtained from the National Centre for Environmental Information (NCEI) and the South African Weather Services (SAWS). The NCEI dataset contained historical daily meteorological observations between 1981 and 2010 which was used to determine the climate values. The SAWS dataset contained more recent daily meteorological observations recorded between 1999 and 2018 that was used for training and validating the machine learning techniques.

1.2.3 Techniques

Three machine learning techniques, namely the Support Vector Machine, Artificial Neural Network, and the Huber regressor, were selected in this study based on their use and performance in the related literature. An analysis of previous work showed that these techniques are suitable for the task of weather prediction and provided satisfactory results.

1.2.4 Evaluation

The performance of the selected techniques was compared based on their ability to predict next day temperatures and their ability to detect temperature anomalies. Two validation techniques namely hold-out and walk-forward validation was used to evaluate the performance of the techniques. The results of the validation techniques were compared to determine which resulted in the most optimal performance. The evaluation metrics used for the next day temperature predictions included the MAPE, MAE, MSE and RMSE. The performance for anomaly detection was evaluated using the Precision, Recall and F1-score.

1.2.5 Tools

The Python programming language and its integrated libraries, namely Scikit-learn, Pandas and Plotly, were used to implement, compare, and evaluate the three machine learning techniques. The results were exported to CSV for further analysis.

1.3 Contributions

The main contribution of this dissertation is the application and evaluation of machine learning as a tool for detecting and predicting weather anomalies. As far as we are aware, this study is one of the few studies conducted in South Africa that evaluates the use of machine learning specifically for temperature anomalies. Thus, this study contributes to the body of knowledge on the use of machine learning to predict weather anomalies in South Africa. The results of the SVM, ANN and Huber regression as well as the evaluation of the hyperparameter configurations can be used in future work. In addition, the evaluation of the walk-forward validation technique [21-23] will show whether this method is suitable for weather prediction.

1.4 Structure of Dissertation

An overview of related work is given in Chapter Two. Chapter Three describes the design of the experiments. The results are presented in Chapter Four. An analysis and discussion of the results follows in Chapter Five. The conclusion is presented in Chapter Six.

Chapter 2

2. Literature Review

This chapter provides an assessment of reviewed literature for the prediction of weather anomalies using machine learning techniques. The chapter is organised as follows. It begins with an introduction to time series analysis. Section two describes how to use machine learning techniques to make predictions from time series data. Section three provides a discussion of weather anomalies and weather anomaly detection approaches. The final section provides an overview of the application of machine learning techniques in related work.

2.1 Time Series Analysis

2.1.1 Time Series Analysis

In this study, time series analysis is considered from the perspective of supervised machine learning. A time series is a process that is generated over time [3]. Each data point in the time series is recorded at uniform intervals and can have any frequency (i.e., hourly, daily, monthly, or yearly) [3]. Each instance is typically described by one (univariate) or by multiple (multivariate) attributes [4]. In time series analysis, the goal is to build a model that can capture the underlying relationships in the series over time [5]. The model receives as input the previous *n* values (where *n* is the number of days) and passes them through a function that returns the output value at a future time. The predictive model can be represented as a linear function that attempts to find the relationship between input variables (x_i) and output variables (Y_i) as expressed in Eq. (1) [19]

$$Y' = \propto \; + \beta_1 x_1 + \; \beta_2 x_2 \; + \cdots \; \beta_n x_n \qquad (1)$$

Where**:**

Y' is a predicted value,

$\propto$ the Y intercept,

$x_1, x_2, \cdots, x_n$ the input features, and

$\beta_1, \beta_2, \cdots, \beta_n$ the coefficients of the input features.

The components of a time series include its trend, seasonal, cyclical, and irregular components. Trend refers to the long-term characteristics of the data that change over time [24, 25]. The seasonal components are periodic changes that repeat at fixed or known times [1, 24, 25]. Seasonal patterns can occur hourly, daily, weekly, monthly, quarterly, or annually. The cyclical component is medium-term changes caused by events that repeat in cycles [25]. These changes are not fixed and do not repeat at known times, such as seasonal variations. The irregular component is unpredictable and irregular fluctuations in the time series [24, 25]. These fluctuations generate noise which can negatively affect the performance of machine learning techniques. This increases the complexity as the machine learning technique would attempt to account for the noise.

Stationarity is an essential requirement for a robust and reliable time series prediction model. When a time series is stationary, the statistical properties such as the mean and variance are constant over time [24, 25]. This means that the way the mean and variance change remain constant over time. Such a process can be described by a linear function in which the rate of change remains constant. A linear process is one in which the relationship between x_i and y_i can be represented by a straight line. In a non-linear process, there is no direct relationship between x_i and y_i. In this case, the rate of change of the output is not related to the rate of change of the input, making such a process difficult to predict because it is independent of its history [26]. If the series has stationarity and linearity, it can be modelled using an equation with fixed coefficients estimated from past data. To achieve stationarity, the time series must be detrended and/or deseasonalized. Estimation or differencing are proposed to remove the trend and/or seasonal components from the series [25, 26]. The implementation of these methods is discussed in more detail in Section 2.2.4.2.

2.1.2 Spatiotemporal Analysis

A time series with spatial attributes, such as weather data, is also called a spatiotemporal data set. Such a dataset consists of a collection of data points across time and space. The attributes include geometric and topological properties [27, 28]. The geometric properties provide information about the position and dimensions of the object, such as latitude and longitude coordinates. The topological properties give information about the existing spatial relationships. Spatiotemporal analysis gives us the ability to understand how the data at a particular location has changed over time.

There are two main types of spatial data, namely raster or vector [27, 28]. Raster data consists of raster cells identified by a row and a column. Examples of raster data include pixelated data such as satellite imagery, photographs, and scanned images. Each pixel is associated with a specific geographic location. Vector data is composed of points, polylines, and polygons. Points represent locations defined by latitude and longitude coordinates. Polylines are used to represent roads, rivers, and streams. Whereas villages and cities are represented by polygons.

Spatial data can be viewed as a three-dimensional data cube (also called a grid) that combines spatial data, such as latitude and longitude, with a temporal dimension, as shown in Figure 1 [29]. Operations can be performed by dicing or slicing the data. Data dicing involves zooming in on a subset of the data across all dimensions, e.g., if we want to know the temperatures at a particular station at a particular time of the year. Slicing data involves filtering the data to obtain information about a particular dimension, such as when we want to know the temperatures at a particular station over time.

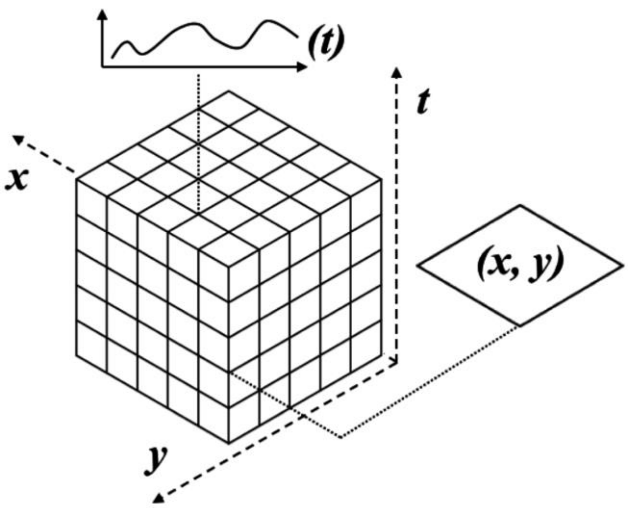

Figure 1 Representation of a three-dimensional spatial data cube [25]

2.2 Machine Learning Process Overview

The study of machine learning is concerned with the use of computer algorithms to gain insight into the future based on a series of training examples [1, 30]. The two main categories of machine learning techniques are supervised and unsupervised learning. In this study, supervised machine learning is used. Supervised learning can be defined as the mapping of input and output data based on the analysis of a set of labelled examples [31].

Chollet (2017) suggests that there are seven universal steps during each machine learning project [32]. The first step involves understanding the problem to define what the desired solution should be. The second step involves selecting an evaluation procedure to assess the performance of the machine learning technique. In the third step appropriate metrics are selected to measure the accuracy of the technique. The fourth step involves preparing the data by transforming it into a machine-readable format. The fifth step involves developing a model that is better than a baseline model. In the sixth step, the technique is scaled to ensure that it is statistically robust enough to model the problem. In the final step, parameter tuning is performed to determine which configuration produces the best results. During this process the hyperparameters are iteratively adjusted and evaluated until an optimal solution is found. Each step is described in more detail in the sections that follow.

2.2.1 Problem Definition

The first step is critical to the success of the project because it ensures that the most appropriate solution is designed. It involves determining the nature of the problem to be solved, e.g., whether it is a classification task (i.e., classifying data points as anomalous or normal) or a regression task (i.e., predicting a continuous output variable such as maximum temperature in degrees Celsius). At this stage, it is important to identify appropriate data to serve as input to the model and to figure out what the desired solution should be, i.e., to determine what the output of the model should be. In addition, suitable techniques and architectures should be identified.

2.2.2 Evaluation Protocol

The goal in machine learning is to create a model that generalizes well to unseen data. There are several techniques to evaluate how well the machine learning technique performs on unknown data. These include hold-out validation, cross-validation, and walk-forward evaluation [23]. In hold-out validation, the last part of the dataset is reserved for testing and the remaining part is used to validate the model [23]. For example, for a dataset of 20 years, a split of 80:20 would mean that 18 years are used for training and the remaining 2 years are used for validation. Cross-validation is a resampling technique that gives a better estimation of the error in the entire dataset as different subsets of the data are used for training and validation [23].

In k-fold cross validation, data is divided into k equal folds. At each iteration, a different fold is selected for testing and the remaining folds are used for validation [23, 33]. The overall accuracy of the technique is calculated by taking the mean accuracy of the k models across the different folds [23, 33]. Since more samples can be evaluated, a more robust evaluation and selection of machine learning models is possible [23]. However, this approach assumes that the data are independent and identically distributed, which is not the case for time series [23].

In the walk-forward evaluation procedure, the performance of the technique is evaluated with successive and overlapping training and validation partitions [21, 22, 34]. First, the number of folds (k) is determined, which determines the number of available training and validation datasets. During training, the first fold (f1) is used to train the technique, while f2 is used to validate the technique. After each iteration, the previous fold used for validation is included in the training set (f1 and f2). This process is repeated for all folds. The accuracy is measured by the mean accuracy of the k-models. Figure 2 illustrates the different validation techniques discussed, namely hold-out validation (Fig. 1(a)), k-fold cross-validation (Fig. 1(b)) and walk-forward validation (Fig. 1(c)).

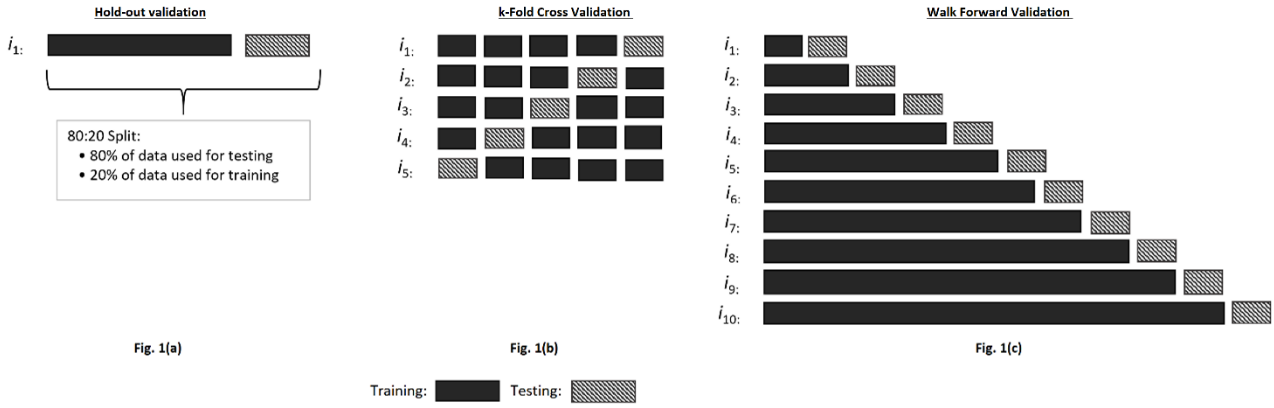

Figure 2 Comparison of the hold-out valuation with an 80:20 data partitioning, 5-fold cross validation and walk-forward validation with 10-folds

The walk-forward technique overcomes some of the limitations discussed of the hold-out and k-fold cross validation. Walk-forward validation allows for a larger part of the dataset to be used for validation which allows for an assessment of a more diverse set of examples as compared to hold out validation [23]. Furthermore, it preserves the temporal order of the time series as opposed to k-fold cross-validation, which does not consider the temporal dependencies of the data. Based on these facts, the technique could be a viable option for time series analysis.

2.2.3 Evaluation Measures

To measure the performance of a machine learning model, various metrics can be used. Metrics should be selected based on the nature of the problem being modelled. Suitable metrics identified for regression techniques include mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE), and mean absolute percentage error (MAPE). For classification techniques, accuracy, precision, recall, and F1 score can be used. A definition of each metric follows in the next section.

2.2.3.1 Performance Metrics for Regression

The main purpose of evaluating regression models is to quantify the error between the predicted value and actual value. Based on the review of the available literature, the following four evaluation metrics were identified:

- The MAE measures the absolute error between an actual value and a predicted value [35]. A small MAE indicates that the model predictions are close to the actual values [35]. The MAE is calculated by taking the average absolute difference between the actual (y) and predicted value ($\hat{y}$), and can be calculated by Eq. (2),

$$\text{MAE} = \frac{1}{n}\sum |\, y - \hat{y} \,|. \tag{2}$$

- The MSE measures the squared average difference between the actual and predicted values, a MSE closer to zero indicates better performance [35]. The MSE can be calculated by Eq. (3),

$$\text{MSE} = \frac{1}{n}\sum (\, y - \hat{y})^2. \tag{3}$$

- The RMSE is used to determine the spread of the error [33]. To calculate the RMSE the average square error is obtained first – this yields the MSE – and then the square root of the MSE [35] Large errors have a great influence on the total squared error, resulting in the squared error growing as the number of large errors increase. This makes RMSE an inappropriate indicator of the average error [35]. The RMSE can be calculated by Eq. (4),

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum (\, y - \hat{y})^2.} \tag{4}$$

- The MAPE is a scale independent measure which makes it a useful measure to compare different techniques. The MAPE represents the mean absolute average error and is calculated by obtaining the sum of the differences between the actual and predicted values divided by the actual value [1, 36]. The value is multiplied by 100 to express it as a percentage, and can be calculated by Eq. (5),

$$MAPE = \sum \left| \frac{y - \hat{y}}{y} \right| * 100. \tag{5}$$

2.2.3.2 Performance Metrics for Classification

Classifiers are evaluated to assess their ability to correctly label instances. To evaluate the accuracy of a classifier, the following metrics can be used:

- The accuracy (A) is an aggregated measure of the classifiers performance [1]. The accuracy is defined as the total number of true positives (T_p) and true negatives (T_n) over the total number observations. The total number of observations include all true positives, true negatives, false positives (F_p) and false negatives ($\mathrm{F_n}$), and can be calculated by Eq. (6)

$$A = \frac{T_p + T_n}{T_p + T_n + F_p + F_n}. \tag{6}$$

- The precision (P) provides a measure of how many actual positive cases were predicted out of all the positive predicted labels [30]. The precision is defined as the number of true positives divided by the total number of true positives and false positives, and can be calculated by Eq. (7)

$$P = \frac{T_p}{T_p + F_p}. \tag{7}$$

- The recall (R) is the measure of how many true cases were predicted out of the actual positive labels [1]. The recall is defined as the number of true positives over the number true positives plus the number of false negatives. The goal is to have a classifier that is good at predicting both classes, such a model would have a high precision and recall, and can be calculated by Eq. (8)

$$R = \frac{T_p}{T_p + F_n}. \tag{8}$$

- The F1-score is a combined metric of precision and recall that is used to measures how well one class was predicted compared to another class [30]. The F1-score is defined in Eq. (9) as precision multiplied by recall over precision plus recall times 2

$$F1 = 2\frac{P \times R}{P + R}. \tag{9}$$

A confusion matrix provides a visual representation of the number of correctly and incorrectly predicted classes, as shown in Table 1. The information contained in the confusion matrix is used to calculate the performance scores discussed in this section.

Table 1 Confusion Matrix

	Predicted positives	**Predicted negatives**
Actual positive	True Positives (TP)	False Positives (FN)
Actual negatives	False Negative (FP)	True Negatives (TN)

2.2.4 Data Preprocessing

Data transformations are required to convert raw data into a format that can be read by machine learning techniques. Real datasets are often unstructured and may contain noise and missing instances that require pre-processing to improve the quality of the data and thus increase the accuracy, speed and efficiency of machine learning techniques [18]. The following section describes some of the key transformations required for time series data. These include dealing with missing data, ensuring stationarity, normalising the data, selecting features, and applying data transformations such as the sliding window transformation.

2.2.4.1 Handling Missing Values

A common problem with time series data is the problem of missing data. Data may be missing due to technical problems or human errors [37]. Many machine learning models, especially neural networks, do not perform well when the dataset contains missing values [1]. There are a number of techniques to deal with missing data, including removing the data instance from the dataset or imputing missing values [37]. Imputing is the process of filling in missing values. In spatially continuous processes, this process is called interpolation [38]. There are many methods for interpolating missing data, some of which are described below:

- Mean/median interpolation: in mean or median interpolation, missing values are replaced by the mean or median of the observed data [37, 39].
- Linear interpolation: in linear interpolation, a straight or curved line is fitted between two data points surrounding the missing instance, essentially taking the mean of the surrounding values [40].
- Hot and Cold Deck Imputation: hot deck imputation replaces missing values with observed data that have similar variables within the same dataset [37, 39]. Cold-deck imputation is like hot-deck imputation but uses external data [37, 39].
- Spline interpolation: spline interpolation uses a piecewise polynomial as the interpolant. Spline interpolation helps avoid the Runge phenomenon caused by fluctuations between points due to high degree polynomials [15, 40].
- KNN Imputation (KNNI): KNNI imputes missing data points by considering a certain number of instances that are most like the instance of interest [37, 39].

2.2.4.2 Data Stationarity

Stationarity is an important requirement for developing a model that provides useful predictions for the future. A process is said to be stationary if its statistical properties, such as mean and variance, are independent of time [25]. A process can be strongly stationary (also called strict stationarity) or weakly stationary [25]. Strong stationarity means that the distribution of the process is constant over time. Weak stationarity means that the mean and variance of the process are independent of time. The Dickey and Fuller test can be used to detect stationarity in a time series [25]. Techniques such as differencing can be used to account for stationarity in time series [26].

Differencing removes the noise sequence generated over multiple lags for each data point. Noise is determined by calculating the difference (also called the residual) between an observation and the corresponding data point in the previous year [25, 26]. Various methods such as moving averages and exponential smoothing can be used to calculate the residual [26]. In moving averages, it is assumed that the next observation is the average of the previous instances. Thus, the residual value is calculated using the average of the previous values. Exponential smoothing is based on the principle of moving averages, but each value is assigned a weight so that more recent values have a greater impact on the residual value.

2.2.4.3 Data Normalisation

The process of data normalization is usually required when data has different scales. Data normalization ensures that larger or smaller values do not have a disproportionate impact on other values [1, 41]. Min-max scaling can be used to convert the values to the same scale without skewing the results. In most cases, the values are scaled between 0 and 1 [1].

2.2.4.4 Feature Selection

The selection of meaningful features is crucial to the modelling process, as non-contributing and insignificant features affect the quality of predictions [13]. There are several methods that can be used to identify significant features, including the Pearson Correlation Coefficient (PCC) and the Principal Component Analysis (PCA) [1, 42]. The PCC measures the strength of the linear relationship between every pair of features. The PCC is used to identify the most related features to the feature being predicted [42]. Among the possible values, a value closer to 1 indicates that two variables are highly correlated, and a value of 1 means that two variables are perfectly correlated. The PCA reduces dimensionality by combining correlated characteristics into a single variable called principal components. For modelling purposes, the principal components with the most information are retained.

2.2.4.5 Sliding Window Transformation

The sliding-window technique converts the data series into a supervised learning problem by connecting input and output sets of instances. The sliding-window technique defines a set of consecutive time series data points as a window and uses a process called back-shifting to obtain the previous values (also called lag) [1, 22]. The input data (x) for each data point represents the nth previous observations and the output data (y) represents the future value to be predicted. When implementing the sliding window technique, the following parameters are specified:

- Window size: The number of data points in the window.

- Step size: The number of data points between successive windows.
- Horizon: The number of data points in the output.
- Skip: The number of data points between the window and the horizon. If the value is zero, the next data point after the window becomes the target variable.

An example of how the sliding technique works is presented in Figure 3 which illustrates the transformation of a data set with a window size of four, a step of one, a horizon of one, and a skip of zero.

Table 2 provides an overview of the identified data pre-processing tasks and the methods used to implement them.

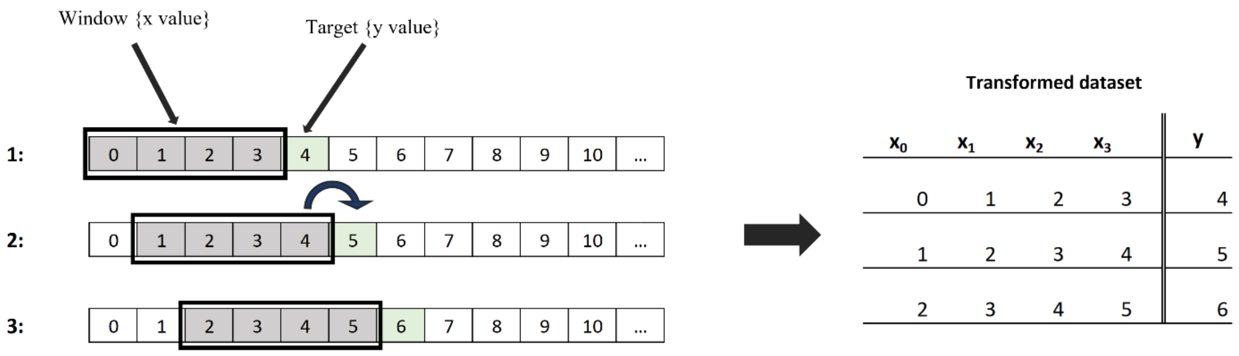

Figure 3 Illustration of the sliding window transformation

Table 2 Summary of data pre-processing techniques for time series.

Data issue	**Technique**	**Method**	**Ref**
Missing values	Deletion	Missing value is removed from the dataset.	[37]
	Mean/Median interpolation	Missing value is derived from the population mean or median.	
	Linear interpolation	Missing values is estimated using points surrounding the missing value.	
	Hot and cold deck imputation	Missing values is replaced with similar values from the same or external data set.	
	Spline interpolation	Missing values is estimated using a piecewise polynomial.	
	KKN Imputation	Missing value is replaced with similar instances.	
Data stationarity	Differencing	Removes the noise sequence from each data point.	[25, 26]
Data normalisation	Min-max scaling	The data is scaled between a certain range.	[1, 41]
Feature selection	PCC	Measures the strength between pairwise variables, variables with the strongest correlation are selected.	[1, 42]
	PCA	Performs dimensionality reduction by combining related variables.	
Sliding window	Back shifting	The method performs lagging to obtain previous days' values.	[1, 11]

2.2.5 Model Development

During training, each training instance is fed to the machine learning technique where internal operations are performed to approximate the mapping function. This section provides an overview of how each of the machine learning techniques selected in this study achieves this mapping function. The techniques, as identified through the review of related work discussed in Section 2.4, are the ANN, SVM, and Huber Regression.

2.2.5.1 Artificial Neural Network

The Artificial Neural Network (ANN) is a self-adaptive, data-driven machine learning technique inspired by biological systems that generalizes from experience [2, 43]. Due to their generalization capabilities and nonlinearity, ANN is well suited for prediction tasks [2, 41]. A neural network consists of multiple layers of interconnected components called neurons that learns the relationship between input and output variables by mimicking the biological processes of the brain [1, 43]. The most commonly used neural network, as shown in Figure 4, is the multilayer feed-forward network (MLP) [1, 2].

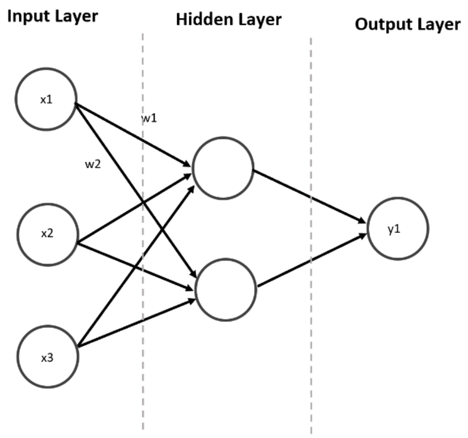

Figure 4: MLP architecture with one hidden layer [1, 2]

In an MLP, input moves in one direction through the network without loops between nodes. The network shown in Figure 4 consists of three layers. The first layer is called the input layer, the second layer is called the hidden layer and the last layer is called the output layer [1]. Each node in the input layer represents an input feature. An additional input feature is added to the incoming features which is called bias. A weight is added to all incoming inputs, which is passed to the hidden layer to perform internal computations. The weights serve as a scaling factor or coefficient that controls the influence of the input on the output. The information from the hidden layer is then passed to the output node, which performs an aggregation function to determine the output. For certain problems, such as classification, the output layer could also have a transfer function that scales the output to the desired range. The neural network learns by adaptively adjusting the weights between nodes. The process of weight adaptation is done by estimating the error between the predicted and actual values of each training instance and adjusting the weights to minimize the error for the next data point. This process is repeated until the minimum error is reached.

Selecting the appropriate network architecture is critical to the success of the neural network. Determining the optimal architecture involves selecting the number of layers and the number of nodes in each layer. Choosing too few layers and nodes may result in the model being unable to learn complex patterns, while too many layers may result in the model learning noise, leading to poor generalization [41]. Figure 5 illustrates how the size of the hidden layers affects the performance of the network. To find the optimal architecture, Stathakis (2009) suggests using one of the following four approaches [44].

- The trial-and-error method: In the trial-and-error method, configurations are gradually adjusted until the optimal combination is reached.
- The heuristic search: In heuristic search, a formula is used to estimate the number of nodes based on the number of input and output nodes.
- The exhaustive search: In the exhaustive search, all possible topologies are tested.
- Pruning and constructive techniques: Pruning and constructive techniques add or remove connections in a network step by step.

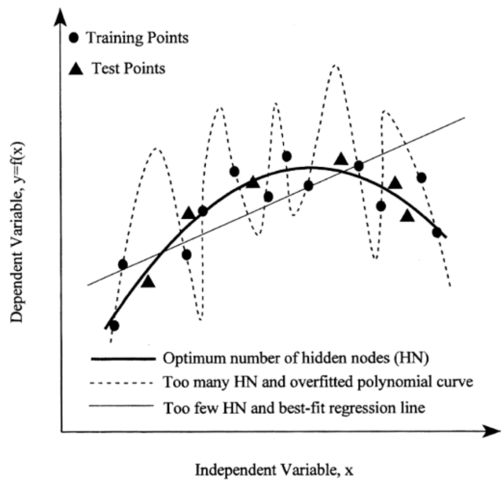

Figure 5 The effects that the size of the hidden layer has on the networks ability to generalise [29]

Once an optimal architecture has been determined, an appropriate activation function should be selected. The activation function determines whether a neuron is activated based on a firing threshold.

Each node constructs a linear classifier to determine whether the neuron should be activated or not based on the incoming values. The activation function of the network can be expressed as [43, 45],

$$\mathrm{H(x)} = \begin{cases} 1 \; if \; x \geq 0 \\ 0 \; if \; x \leq 0 \end{cases}. \tag{10}$$

Here, the neuron is activated when the input received from its connected nodes is greater than zero, and when the input received is less than zero, the neuron will not be activated. The sigmoid and hyperbolic tangent functions are the two main activation functions proposed in literature [43].

Before the training process starts, initial values are assigned to the weights of the model. During the training process, the values of the weights are adjusted to minimize the cost function of the network. The adjustments to the weights are performed through a process called backpropagation. Backpropagation is a systematic training method for multilayer ANN, where information about the error of each training instance is returned to the network [43]. The network uses this information to dynamically penalise weights that do not minimize the error. This process is repeated until a local minimum of the cost function is found [2]. The changes applied to the weight values are controlled by the learning rate. The learning rate controls how the network adjusts the weights during training [41]. A high learning rate results in larger weight changes that can result in the global minimum being missed. However, a low learning rate results in smaller weight changes, which may cause the network to take a long time to converge or not converge at all. The learning rate can be set to a constant value or determined through adaptive learning [41].

2.2.5.2 Support Vector Machine

The SVM is a supervised machine learning technique based on the statistical learning theory of Vapnik et al [46, 47]. During training, the primary goal of the SVM is to minimize the loss function. The SVM attempts to place a boundary, also called a hyperplane, around a group of points that belong to the same class [1]. This is achieved by finding a core set of points that identifies and establishes the boundary. These points are called support vectors. The best hyperplane is the one that maximizes the distance between classes with minimal error. SVMs are suitable for both regression and classification tasks, but the training objective differs.

The SVM for classification (SVC) solves a binary classification problem that aims to maximize the distance (or margin) between classes while minimizing the error [1, 47]. First, the SVM finds the class boundary by combining all data points into a convex hull. Next, the SVM identifies all the support vectors to find the hyperplane that best separates the classes. Once the boundary is found, the test pattern is classified by calculating which side of the hyperplane it falls on. Figure 6 shows an illustration of an SVM for classification.

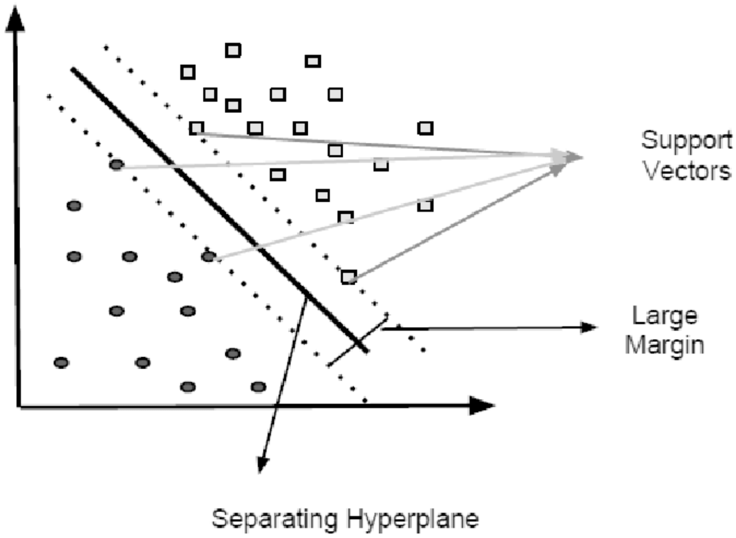

Figure 6 Illustration of the SVM for classification [44]

The SVM for regression (SVR) finds the function that best approximates a continuous variable [47]. In regression, a flexible tube is constructed around the estimator using Vapnik's e-insensitive approach [47, 48]. In contrast to classification, the goal in regression is to fit the flattest tube with the most training instances within the bounds of the tube [47]. In this case, the support vectors are the training points closest to the boundaries of the tube which controls the shape of the tube. By applying the symmetric loss function, errors above and below the boundaries are penalized equally [47, 48]. Figure 7 illustrates how the loss function works, the solid line represents the estimation function, defined in Equation 11

$$w \cdot x + b = 0 \tag{11}$$

and the dashed lines represent the upper and lower bounds within ε distance of the function as defined in Equation.12.

$$w \cdot x + b = +\varepsilon \tag{12}$$
$$w \cdot x + b = -\varepsilon$$

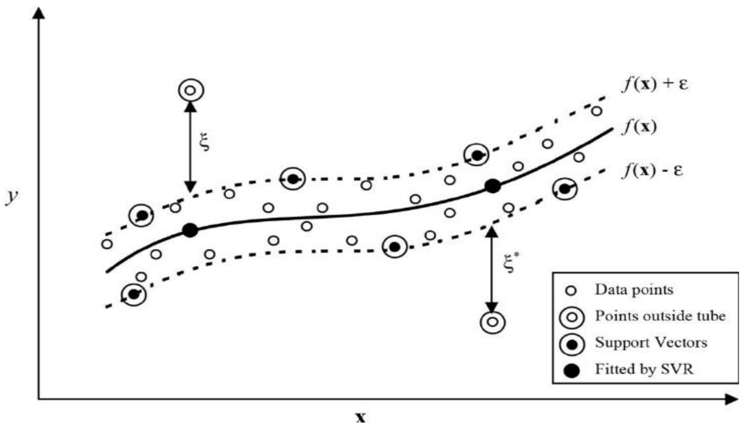

Figure 7 Illustration of the SVM for regression using the ε -sensitive loss function [46]

The hyperparameters ε, C, and kernel most strongly influence the predictive performance of the SVM. The parameters ε and C control the number of support vectors and the error tolerance of the model [46, 47]. It should be noted that a large number of support vectors may not result in a good model, so a good balance should be achieved [9]. The kernel can be used to extend the decision function to nonlinear cases by mapping the input data into a higher dimensional space, as shown in Figure 8 [46, 47]. Time series analysis is most commonly performed using the following kernel functions [49]:

- The Linear function can be expressed as

$$K(x_i, x_s) = x_i x_s \tag{13}$$

- The Radial Basis Function (RBF) can be expressed as

$$K(x_i,\ x_s) = \exp\left(y - \left|\left|x_i - x_s\right|\right|\right)^2 \tag{14}$$

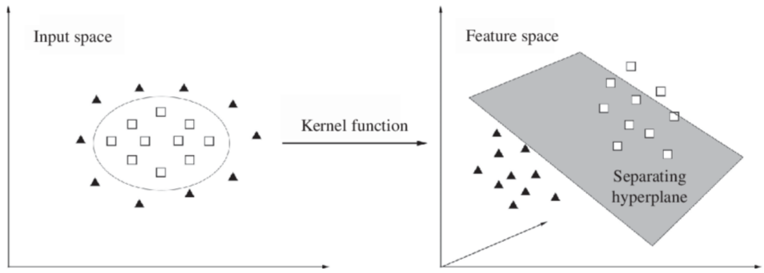

Figure 8 Kernel transformation from the input space to a n-dimensional feature space for non-linear separable data [48]

2.2.5.3 Huber Regressor

Regression is the process of determining the relationship between one or more independent variables and a dependent variable. Huber regression is a linear regression function (defined in Eq. 1) based on robust regression theory. The presence of heavy-tailed errors affects the consistency of high-dimensional regression methods such as the ordinary least squares function, which is less robust to outliers [50]. Wang et al. (2019), Sun et al. (2020), Chen et al. (2020), and Fan et al. (2019) suggest the use of the robust loss function because it reduces the negative effects of outliers [50-53].

The Huber regression function does not treat all data points equally but assigns a weight to each residual based on its magnitude. Outliers are given smaller weights because the loss function assigns smaller weights to these values [52]. A robustness parameter is used to control the stability of the model [52]. The value of the parameter can be fixed or adaptive. Sun et al. suggest the use of an adaptive value because a fixed value may lead to an unacceptable bias in the estimation. When the robustness parameter is adaptive, the values of the weights are adjusted based on the sample size and its dimensionality to ensure an optimal trade-off between bias and robustness [52].

There are several variations of the Huber loss function, including hybrid, scaled, and smooth non-convex functions [53]. Recent research has largely focused on Adaptive Huber Regressor (AHR), based on the work of Sun et al. In this work, the authors used an adaptive Huber regression model for robust estimation and inference that achieved robustness and unbiasedness [52]. In addition, Fan et al. based their work on the AHR to evaluate how well the estimator performed on high-dimensional datasets with temporal dependencies [50]. The study by Chen et al (2020). was based on a novel approach to deal with collinearity and outliers in multivariate analysis by using a regularized linear Huber regression model with elastic net. The models were able to reduce the negative effect of outliers and grouped correlated variables well.

2.2.6 Model Scaling

Model scaling is performed to ensure that the technique is statistically powerful to model the problem. The goal in scaling is to find the best balance between optimization and generalization. Machine learning techniques tend to overfit, which can lead to poor generalization [1, 33]. A model that overfits, memorized the training data instead of learning the underlying patterns in the data [1, 33]. Such a model would usually perform well during training but will have lower accuracy when tested on unseen data. Underfitting occurs when the model is not complex enough to explain the relationship between inputs and outputs during training. A good model lies between the boundaries of underfitting and overfitting.

2.2.7 Model Parameter Tuning

Model tuning is a trial-and-error process in which the settings of the model are repeatedly changed until an optimal result is found. The goal of this step is to find the best fitting configuration settings. Parameter tuning involves experimenting with different hyperparameters, the performance of the model is continuously evaluated, allowing for an informed selection process [32]. When the most satisfactory settings are determined, the model is evaluated against the validation data.

2.3 Weather Anomaly Overview and Detection Approaches

2.3.1 Introduction to Weather Anomalies

The occurrence of weather anomalies and extreme events is related to atmospheric instabilities that can disrupt certain economic sectors (e.g., agriculture, mining, manufacturing, and construction) and daily human activities. This section is intended to provide a comprehensive definition and discussion of weather anomalies and extreme events.

Weather anomalies are observations that deviate from the baseline average for a given day and weather station [54]. Baseline values indicate how the atmosphere has behaved over time and represent the expected climate in a given region [55]. The baseline average for meteorological studies is determined by calculating the daily average temperature values during the standard reference period. The standard reference period established by the WMO is between 1981 and 2010. When calculating daily averages, the WMO practice guidelines recommend that no more than eleven days in a month or five or more consecutive days in a month should be missing [56].

Extreme weather events are characterized by anomalous weather conditions that deviate significantly from expected and persist for an extended period [9]. To measure how much an observation deviates from expected conditions, a threshold is used [5]. In South Africa, the South African Weather Services (SAWS) is responsible for setting these thresholds. There are several extreme weather events that can occur in South Africa, including floods, thunderstorms, heat waves, and droughts [5, 57-60]. The following sections describe the definitions for temperature extremes.

2.3.2 Temperature events classification in South Africa

The thresholds given below applies to South Africa:

- A heat wave is an extended period of unusually hot weather lasting three or more consecutive days [5, 6, 55]. Heat waves typically occur during the summer months of the year, depending on local climatological conditions [55]. The threshold of SAWS is set at 5°C above the average temperature recorded during the warmest month for the station [61].
- A warm spell is defined as persistent warm weather, but unlike a heat wave, a warm spell can occur at any time of year [55]. A warm spell is defined as the 90th or 95th percentile of the thermal index [55]. The index can be a single meteorological element, such as daily maximum temperatures, or a combination of elements [55].
- Cold waves are characterized by a sharp decline in surface temperatures, resulting in low temperatures and an increase in atmospheric pressure, which leads to strong winds due to the increase in gradient force [5, 55]. The threshold value of SAWS for the cold season is when temperatures are below -2.5°C for three or more consecutive days [55, 61].

- Cool spells are defined by the WMO as persistent temperatures below average during the warm season [55].

2.3.3 Current Prediction Approaches

Prediction of extreme weather events is challenging due to the multiscale interactions associated with these events [62]. In traditional anomaly prediction, statistical, numerical, and empirical physical modelling techniques can be used to study extreme events [62]. One of the most common methods used to study climate extremes is statistical modelling, also known as extreme value theory [62]. Extreme value theory determines the probability of extreme weather events based on current and historical atmospheric conditions. Numerical modelling is based on statistical considerations to determine the probability of extreme events [62]. Both extreme value theory and numerical approaches are often used to study extremes in meteorological data. Empirical physical reasoning is based purely on empirical knowledge and physical reasoning about extremes [62]. However, this approach lacks rigour compared to extreme value theory and numerical modelling.

Anomaly prediction in machine learning can be performed through supervised (labelled data indicating normal and abnormal points), semi-supervised (training the data with normal instances only) and unsupervised learning (unlabelled data consisting of normal and abnormal instances). It could be said that anomaly prediction in machine learning is based on a combination of extreme value theory and numerical approach, as it uses historical information to determine the probability of an extreme event based on the underlying mathematical models.

2.3.4 Anomaly Detection

The goal of machine learning is to find a model that best describes the data; therefore, the model can also be considered an underlying process that generated the data. Outliers or anomalies are data points that are unlikely to have been generated by the model and deviate from other observations. They are usually eliminated from the data because they add complexity to the model, as the model will also try to explain the outlier. An anomaly is a data point that was generated by a different process to the one that generated the rest of the data. Anomaly prediction attempts to identify patterns in the data that are significantly different from the general trend and/or pattern in the data set [63, 64]. The Z-score can be used to quantify how much a data point deviates from the normal distribution [63]. The Z-score is calculated using Eq. (15):

$$Z_i = \frac{|\, X_i - \mu \,|}{\sigma}. \tag{15}$$

Where:

(X_i) is the observed value,

μ is the mean and,

σ the standard deviation.

Different types of machine learning techniques can be used for anomaly prediction [1]. The first group of techniques are distance-based. Distance-based techniques measure the distance to the nearest neighbours of the data point to detect anomalies. If the distance is higher than normal data points, the instance is classified as anomalous. The second group of techniques is density based. Density based techniques assume that anomalies occupy low density areas because they are less frequent, while normal data points occupy high density areas. The next group of techniques are cluster-based. Cluster-based techniques treats data points that are not part of a cluster as an anomaly. The last group of techniques are classification-based. Classification-based techniques learn what is considered an anomaly based on a labelled data set.

The goal is to develop a machine learning model that can distinguish between instances that fit the collective trends and patterns of the dataset and instances that do not. The training process can be performed using one of the following methods [64]:

- Supervised anomaly prediction:

  In supervised anomaly prediction, a machine learning model is trained using labelled training data. Labelled training data in time series are pairs of historical (input value) and future observations (output value). The data labels of each data instance indicate whether the instance is normal or anomalous. The labels can be a score indicating the extent to which an instance is anomalous or a label indicating whether a data point is anomalous or normal. The machine learning model derives knowledge about the labelled examples to predict whether an unknown point is normal or anomalous.

- Semi-supervised anomaly prediction:

  In semi-supervised anomaly prediction, a machine learning technique is trained using only normal data points. This approach results in the model learning what the collective trend and pattern is. When tested on unseen data, any data point that does not follow the normal pattern and trend is flagged as an anomaly.

- Unsupervised anomaly prediction:

  In unsupervised anomaly prediction, a machine learning technique is trained with an unlabelled data set. Unsupervised learning assumes that normal instances are more common than anomalies. When an instance occurs that does not fit the normal criteria, it is classified as anomalous. If this assumption is not true, it would likely result in a model with a high false positive rate.

2.4 Applications of Machine Learning for Weather Prediction

A comprehensive summary of the work described in this Section can be found in Appendix A1. The results of the literature review show that machine learning is a viable solution for weather prediction. Machine learning has been applied to a variety of weather problems including extreme precipitation warning systems, maximum and minimum temperatures, precipitation price derivations, and drought detection. Most of the literature reviewed conducted a comparative evaluation of different machine learning techniques for predicting weather events. The most common techniques used included the ANN, and SVM.

The objective of this section is to analyze how machine learning techniques have been implemented in related studies. This includes how anomalies or extreme events were classified, what input variables were used, how various data problems were solved during the preprocessing step and how the machine learning techniques were trained and evaluated.

2.4.1. Machine Learning techniques

Of the thirteen studies included in this review, only two used classification-based solutions [8, 9]. A total of fourteen different techniques were evaluated in the literature reviewed, with ANN [3, 8, 10, 13-19], SVM [8, 9, 11, 12, 14, 17-19], regression [3, 8, 14, 19] and random forest [3, 8, 14, 17] being predominately used.

For the validation of the machine learning techniques, hold-out validation and k-fold cross validation was predominately used and evaluated using one or more of the following metrices: MAE, MSE and RMSE for regression-based techniques and precision and recall for classification-based techniques. The data pre-processing tasks applied included filling missing values using various interpolation methods, feature extraction and the sliding-window to perform lagging of the dataset. To measure the statistical abilities of the techniques, a baseline model was used to set the benchmark for expected performance. Methods applied to improve performance included experimenting with different partition sizes, window sizes, different selection of features and tuning of hyperparameters.

There are many features that can be used as input during the training process, such as temperature, precipitation, humidity, air pressure, wind speed, wind direction, and dew point. Auxiliary variables such as date, time, latitude, and longitude can also be included. In the regression tasks, the output for precipitation was either daily/hourly/seasonal precipitation or cumulative precipitation within a given period [3, 11, 12]. In the case of temperature prediction, most studies used daily/monthly temperature values as output [16, 18, 65]. For classification tasks, a class label was used to indicate whether the data point was anomalous or normal [8]. On average, 26 years of training data were used for model training but ranged from a minimum of four years to a maximum of 111 years.

Few examples where spatial attributes were incorporated during training process were found in the work reviewed. A study by Cho et al. (2020) investigated machine learning-based bias correction methods for the Numerical Weather Prediction (NWP) model for extreme air temperatures. The authors used geometric and topological variables recorded during summer as input values. The latitude and longitude variables were used to indicate location, while elevation, slope, and solar radiation were used as the topological variables. The output variables were the maximum and minimum next day in-situ air temperatures.

2.4.2 Approach for Weather Prediction and Anomaly Detection

In studies dealing with the prediction of anomalies or extreme weather events. The classification criteria for anomalies depended on the region. In Mumbai, India, a precipitation anomaly was defined as a deviation of more than 1.25 standard deviations from the climatological mean and precipitation of more than 75 mm within a 6-hour window was classified as extreme [9]. In a similar study conducted in South Korea, the authors used two classes for heavy precipitation. The first class is precipitation of more than 70 mm in 6 hours and the second class is precipitation of more than 110 mm in 12 hours [8]. Regarding extreme or anomalous temperature prediction, the related studies reviewed did not directly focus on the prediction of anomalies, therefore no criteria were specified. In one of the two studies related to temperature predictions the authors applied machine learning to correct the bias of NWP models for extreme temperature predictions [17]. Another study used the results of the machine learning models to reveal anomalies in the data caused by pollution and changing weather patterns [15]. A semi-supervised anomaly prediction procedure was used by [8] which involved excluded extreme instances during training. Thus, the machine learning model learned what is considered normal resulting in data points being classified as anomalous if they did not conform to this behaviour.

2.4.3 Evaluation Protocols Used

Validation techniques such as the hold-out validation and k-fold cross-validation were used to evaluate the performance of the models developed. The hold-out validation method was the most used technique in the papers reviewed [9-11, 13-15]. The most common approach was to use about three quarters of the data for training and the rest for validation. The respective training/testing sizes are documented in summary included in Appendix A1. Only five of the studies reviewed used k-fold cross-validation for training [3, 8, 12, 19]. Different fold sizes of 3 [8], 5 [19] and 10 [3, 12] were used. [17], used two different techniques from other studies, namely hindcast validation and leave-one-station-out-cross-validation (LOSOCV). Hindcast validation is a process in which time series models are trained using all available historical data. For example, if the study period is from 1999 to

2018 and temperatures are to be predicted for January 1, 2004, the model is trained with data from the first date in the dataset to December 31, 2003. LOSOCV is the process of excluding certain stations during training and then using them during validation to compare the generalization ability of the techniques on different locations.

2.4.4 Evaluation Metrics Used

Various metrics have been used to evaluate regressors and classifiers. The most common metrics used to evaluate regression problems are MAE [3, 12, 13], MSE [3, 10, 15] and RMSE [3, 13]. Other metrics include the coefficient of variation of the RMSE CV [11], the median absolute distribution (MAD) [3] and the F1-score [14]. In addition to these metrics, the correlation coefficient (CC) was used to measure the strength between the actual and predicted output [10, 12-14]. Precision and recall were used to evaluate classifiers [8, 9], but [8] also considered the f-measure and Equitable Threat score (ETS).

2.4.5 Data Preprocessing

2.4.5.1 Methods Used for Missing Values

In the studies reviewed, various techniques were used to deal with the problem of missing data, including deletion and interpolation methods. Moon et al. (2019) [8] chose to completely remove instances with three or more missing attributes from the dataset. Interpolation methods include spline interpolation [15], linear interpolation [19], mean interpolation [18], and cokriging [17]. Tyagi et al. (2016) [15] found that Spline Interpolation was the more reliable method for dealing with missing temperature values compared to Polynomial Interpolation. Spline interpolation is a form of interpolation that uses a piecewise polynomial to estimate missing values. In addition, this technique overcomes the problems associated with fluctuations that occur between points when high degree polynomials are used. For mean interpolation, Radhika and Shashi (2009) [18] used the mean maximum temperature of each month to replace the missing instances in that month.

2.4.5.2 Feature Extraction Techniques Applied

The feature extraction process is an important step because meaningless input features will result in meaningless output. A large number of input variables can degrade the performance of machine learning techniques and significantly increase computation time and memory, leading to possible overfitting [8]. There are two types of dimension reduction techniques applied in related work to address this problem: Feature Extraction and Feature Selection. In feature extraction, high-dimensional attributes are projected into a low-dimensional space, while in feature selection, a subset of attributes is selected [8].

The most used feature extraction technique is Principal Component Analysis (PCA) [3, 8, 12]. Another technique used by [13] is minimal Redundancy Maximum Relevancy (mRMR) method. The mRMR selects the features that are most relevant and least redundant. Anjali et al. (2019) selected only features that were highly correlated with the outcome variable. Richman & Leslie (2018) used both linear (correlation-based feature selection) and non-linear feature selection methods (Radial Basis Function Support Vector Regression feature selection). The linear methods considered the individual predictability of each feature and the degree of redundancy of each feature, while the non-linear selection method used a generic search technique to find the features with the best performance.

2.4.5.3 Sliding Window Transformation

The sliding window technique is used to convert a dataset into a machine-readable format by linking input and output variables. The input record usually consists of a set of historical observations and the output value consists of a future value. This technique was also applied by [9, 11] to calculate the cumulative rolling rainfall. As noted by Cramer et al. (2017) this technique helped to smooth the volatile characteristics of the precipitation data. To test the effectiveness of the technique, the results of the technique trained on normal precipitation data and accumulated precipitation were compared. A 70% improvement in performance was found. Radhika et al. applied the technique by dividing the data set into five equal windows of 193 days each to train the model during the cross-validation process.

2.4.6 Model Development

This section presents the methods used to conduct the experiments in the literature discussed. This serves as the basis for conducting the experiments in this study. This section discusses the methods used for data pre-processing, modelling and evaluation of the SVR, ANN and linear regression techniques and the results obtained from the reviewed studies.

2.4.6.1 ANN for Weather Prediction

Artificial neural networks were the most used technique in the literature reviewed because of their ability to model complex nonlinear relationships between variables. The technique has been used to predict various atmospheric conditions, including normal precipitation [3, 10-12] and short-term extreme precipitation within a 48-hour period [8, 9] as well as temperatures [15-19]. Several variants of neural networks have been used in the literature discussed, including the multilayer perceptron (MLP) [3, 8, 13, 15], Convolutional Neural Networks (CNN) [14], and Backpropagation Neural Networks (BPNN) [10].

The most widely used feed-forward neural network architecture consists of three layers, including an input layer, a hidden layer, and an output layer. The number of nodes in the hidden layer varied from three [3] to four units [13, 15]. [18], on the other hand, used a formula of ($2i$ +1) to determine the number of nodes, where i is the number of input features. The set hyperparameters varied between studies as different applications were used during training: In MATLAB/Keras, the activation function , optimizer and dropout rate were set [14, 17]. The value ranges for these parameters are as follows:

- Activation function: The Rectified Linear Units (ReLU) [14, 18] and Sigmoid function [18].
- Optimizer: Adam [17], and
- Dropout rate: 0.1 [14].

The authors used cross-validation [3] or the hit-and-search method [15] to find the optimal architecture and hyperparameter configuration. The most used learning technique for the feedforward neural network was the backpropagation technique [8, 13, 18, 19]. However, [16] compared eight training techniques to determine which offered the best performance for temperature prediction. To test the robustness, some authors [3, 14] repeated the training process of the ANN.

It was found that ANN outperformed other methods in the majority of studies [3, 10, 14], and was computationally efficient in weather prediction, but there were some cases where other techniques such as SVM [18], M5P model tree - The M5P model tree combines a decision tree learner with linear regression functions at the nodes - [13], and Multiple Linear Regression (MLR) [19] performed better than the ANN. A comparison of different learning techniques in a study by [16] revealed that the best training techniques for temperature prediction are BFGS Quasi Newton (BFG), Levenberg-Marquardt (LM) and Scaled Conjugate Gradient (SCG) with correlation coefficients of 0.95, 0.94 and 0.93 respectively. The BFG technique approximates a Hessian matrix in each iteration instead of computing 2nd derivatives [16]. However, the technique requires more computations and memory in each iteration performed. LM is a powerful optimization technique for computing the Jacobian matrix which is less complex than computing the Hessian matrix [16]. The SCG technique developed by Moller uses a region approach used by the LM technique, which is combined with a conjugate gradient approach at each iteration, rather than a line search normally used by conjugate gradient techniques [16]. In addition, Tyagi [15] found that ANN works more efficiently on longer-term data.

2.4.6.2 SVM for Weather Prediction

The SVM was the second most used technique in related work after ANN. It has been used to predict normal [11, 12] and extreme precipitation [8, 9], and daily temperatures [17-19]. In most studies, the authors compared SVM with other techniques such as ANN and Multiple Linear Regression (MLR). One of the interesting applications of SVM was the development of a two-stage solution for predicting extreme rainfall [9]. The authors found that the patterns before extreme events differ depending on the time of day, so they constructed two models to fit both patterns. The first model was trained for the night-time pattern, and all non-extreme events predicted by the first model were fed into the second model to determine if they fell within the daytime pattern. This approach ensured that any events missed by the first model were detected by the second model. Another study by Cho et al. (2020) developed an ensemble model using the SVM, ANN, and MLP to correct for the biases of

Numerical Weather Prediction (NWP) models. The results of this study showed that using more than one technique in the form of an ensemble performed better than a single technique.

In these studies, the authors experimented with using different kernels (polynomial kernel and radial basis function (RBF)) [12, 18, 19]. The results show that the RBF kernel provided higher accuracy compared to the polynomial kernel. The type of hyperparameters that are set during the training process depends on the type of kernel chosen. For the RBF kernel function, the regularisation parameters C and $\in$ are required. For the parameters C and $\in$ [18] first used a mixture of larger values and then performed a second iteration of evaluation for fine tuning of the values as per the first iteration. In the first iteration of hyperparameter tuning the values for C was 1, 10, and 100, in the second iteration lower values of 0.5, 0.7 and 0.9 were considered. For $\in$, 1 and 2 was used during the first iteration, the second iteration was performed with $\in$ equal to 0.001, 0.01, 0.1 and 1. The hit-and-search method [9], grid search [18] and heuristic methods [17] were used to determine the optimal hyperparameter values.

The SVM was found to perform satisfactorily in most cases and outperformed other methods including ANN [9, 11, 12, 18]. Radhika & Shashi (2009) concluded that SVM could replace neural network-based models for weather prediction with appropriate selection of hyperparameters, based on its performance in temperature prediction. The success of SVM is due to the principle Structural Risk Minimization (SRM) that minimizes the upper bound of the generalization error rather than the training error [18]. It should also be noted that SVM has a high false positive rate when classifying the data. Nayak & Ghosh (2013) proposed the use of fine resolution data for extreme rainfall prediction to address this issue.

2.4.6.3 Regression for Weather Prediction

Linear regression techniques have been used in related studies for predicting seasonal precipitation anomalies [3], temperature prediction [19], and early warning systems for extreme short-term precipitation [8]. The range of linear regression techniques used included general linear regression

(GLM), generalised additive models (GAM), multivariate adaptive regression spline (MARS) [3], multiple linear regression (MLR) [19] and logistic regression [8].

GLM and GAM are likelihood-based regression models. GLM uses a link function that allows the technique to establish a linear relationship between the response and the predictors, even if the underlying relationship is not linear. GAM is an extension of GLM. GAM uses a nonparametric smoothing function that allows it to model nonlinear relationships. MARS is a form of regression analysis introduced by Friedman, which is technically an ensemble of linear functions connected by hinge functions. The technique consists of two stages, the forward stage, and the backward stage. In the study in question, the MARS models were built using recursive partitioning trees with an option for a forward and backward pass. MLR is a statistical technique that uses multiple explanatory variables to predict the outcome of a response variable. Logistic regression is a type of regression analysis for binary classification that uses a logistic function to predict binary outcomes from continuous values. The equation used to build the predictive models for regression analysis are defined in Eq. 1, Section 2.1.1.

In predicting seasonal precipitation, the best performing models were ANN and GLM [3]. In this study, the GLM was one of ten techniques evaluated. The GLM achieved a CC of 0.71 and MAE of 0.48 compared to ANN, which achieved 0.81 and 0.48 respectively [3]. In the case of temperature prediction, MLR proved to be more accurate than ANN and SVR [19]. The average error (AE) was 1.0782 compared to the values of ANN and SVR of 1.2958 and 1.1371, respectively. In the case of extreme rainfall classification for EWS, the logistic regression function showed promising results for the prediction of very short-term heavy precipitation [8]. The method proposed in this study used selective discretization to selectively discretize input variables with a nonlinear relationship with short-term heavy rainfall events, and PCA to reduce the dimensionality of input variables to obtain an informative view of the data. The results obtained in above mentioned studies shows that linear regression techniques are suitable to predict weather events.

2.4.7 Methods Applied to Scale Up

The model scaling step ensures that the machine learning technique is statistically powerful to model the problem. This can be done by comparing the technique to a baseline model. In some of the related studies, techniques such as the average, null, and the memory [3] were used as the baseline.

- Average model: The average model takes the mean of all predictions from models as the expected value.
- Null: The null model uses the mean of the feature being predicted in the training data as the expected value.
- Memory: The model uses the previous year's values as the expected value.

Other studies [14], used simpler machine learning techniques which include the SVR, linear regression, and random forest to evaluate more complex techniques such as the CNN. If the technique was found to computationally outperform a baseline it can be considered for further refinement, else it might imply that the technique or approach is not suitable to model the problem and should therefore be reconsidered. Some of the steps that can be taken to improve the performance include using different architectures and/or different model inputs [15].

2.4.8 Methods Applied to Obtain Optimal Hyperparameters

The following techniques were applied in related work to ensure that the best possible configuration settings were used:

- Testing different data partitions: [13] evaluated the performance of the techniques when trained on different training sizes, the training data used in this evaluation consisted of 66%, 75%, and 80% of the available data.
- Testing different window sizes: [18] tested five different window sizes (2, 3, 5, 7, and 10) during the training process and concluded that a window size of five (each window consists of 193 days) produced the best performance.
- Feature selection: [11] tested the effects of feature type on model performance by training the model on daily precipitation rates and accumulated precipitation. The authors found a significant improvement in model performance when using accumulated precipitation.

- Hyperparameter tuning: Parameter tuning was performed using the hit-and-trial method [9, 14], grid search [18], and automatic scaling techniques such as iRace which is an offline parameter tuning tool [11] and a Sequential Minimal Optimization (SMO) [17]. Automatic scaling has the advantage of faster processing compared to grid search. The techniques work by iteratively trying different parameter configurations and evaluating them across multiple instances to find the optimal configuration that fits the climate of a given dataset.

2.5. Summary of Key Findings

2.5.1 Anomaly Prediction

An anomaly is defined as the difference between a temperature measurement on a given day and the baseline value for a location. Temperature anomalies are disturbances in the atmosphere that cause temperatures at a station to be warmer or cooler than expected conditions for that time of year. Extreme temperature events such as heat waves, warm spells, cold waves, or cool spells occur when temperatures exceed certain thresholds and persist for an extended period. There is no standard reference for the anomaly threshold, as what is considered anomalous depends on the location. Statistical measures such as the z-score can be used to quantify how much an observation deviates from expected conditions to determine if an anomaly exists.

The review of related work revealed that the ANN and SVM were predominately used for daily or monthly temperature prediction in five out of thirteen studies, the remaining studies focused on the prediction of rainfall. Out of the five studies focusing on temperatures, two focused on some element of anomalous or extreme temperatures. One of the studies used machine learning techniques to correct for NWP model biases in predicting extreme air temperatures, while another used model outputs to examine the effects of pollution on anomalies and/or weather pattern shifts.

2.5.2 Machine Learning for Weather Prediction

Developing a robust methodology for machine learning projects is critical to the success of the project. This requires ensuring that the goal is clear, that appropriate measures are selected to evaluate and validate the machine learning models, that reliable data is available and processed appropriately, and that the appropriate techniques are trained and optimized. Widely used validation techniques are hold-out validation and k-fold cross-validation. It should be noted that the walk-forward approach identified through the literature review was not used in any of the papers reviewed. As noted from the review of available literature, the prediction error of regression-based models can be measured using MSE, MAE and RMSE, while for classification-based models' accuracy, precision, recall and F1 can be used.

A common problem in related works was the problem of missing data. Various interpolation techniques such as spline interpolation, linear interpolation, cokriging and mean interpolation were used to fill in missing values. When working with time series data, issues such as stationarity must be considered. Differencing techniques such as the moving averages or exponential smoothing can be used to make a non-stationary time series stationary. The sliding window technique has been useful in smoothing precipitation variability and led to a 70% improvement in results [11]. When processing data with different scales, it is useful to perform normalization to transform the data to the same scale. As reported in [9], feature weighting and scaling led SVM to better classify extreme and non-extreme values.

The most popular ANN was a three-layer feed-forward network, where the number of nodes in the hidden layer varied between three and four. For the SVM, the RBF kernel performed better than the polynomial kernel, but not much was found for the other hyperparameters. The hyperparameters were tuned using the trial-and-error, grid search and automatic scaling techniques. Several studies [3, 11, 18] emphasized the need to match the hyperparameter settings to the climate of the dataset to ensure that the optimal parameters are used.

The results of the studies show that weather prediction can be reliably performed using machine learning techniques. Furthermore, the techniques were found to be computationally inexpensive compared to current methods [14, 17]. Moreover, the performance of ANN was found to be equivalent to the systems currently in use [3, 15]. The SVM showed promising results and has been cited in some cases as a suitable replacement for the current ANN weather prediction models [9, 11, 18]. Several studies have tested the solution in different locations and found that it is transferable to other locations with different geographical climates [10, 11], which is promising for universal application. However, Cramer et al. (2017) found that there is a correlation between climatic conditions and prediction error, as certain climates were predicted better than others.

The results of the studies show that atmospheric temperature can be reliably predicted using the selected techniques (SVM, ANN and regression). The SVM had an MSE of 7.5 [18] and an average RMSE of 1.72535 [17, 19]. The ANN had an MSE of 8.07 [18] and an average RMSE of 1.795 [17, 19]. The MLR had an RMSE of 1.8568 [19].

Chapter 3

3. Experimental Design

This chapter presents the methodology and design of the experiments conducted as part of this thesis. The chapter is organised as follows. It begins with an overview and analysis of the data. This is followed with details of the implementation procedures for prediction and classification tasks. The third section provides an overview of the methodology to implement and evaluate the machine learning techniques. An overview of how the experiments conducted in this study was implemented is provided in Section four. The Chapter concludes with an overview of the implementation workflow.

3.1 Datasets

3.1.1 Overview

Two datasets containing meteorological observation were obtained. The first dataset obtained from the NCEI contained historical meteorological observations which was used to calculate the climate baseline. The dataset contained observations taken between 1981 and 2010 as described in Section 2.3.1. More recent meteorological observations (between 01 January 1999 and 31 December 2018) obtained from the SAWS was used to train and validate the machine learning techniques. The datasets were collected from 17 weather stations in Gauteng and Northern Cape. Each of the stations is shown on the map in Figure 9, with those in Gauteng highlighted in blue and those in the Northern Cape highlighted in red. An overview of each dataset is provided in Table 3.

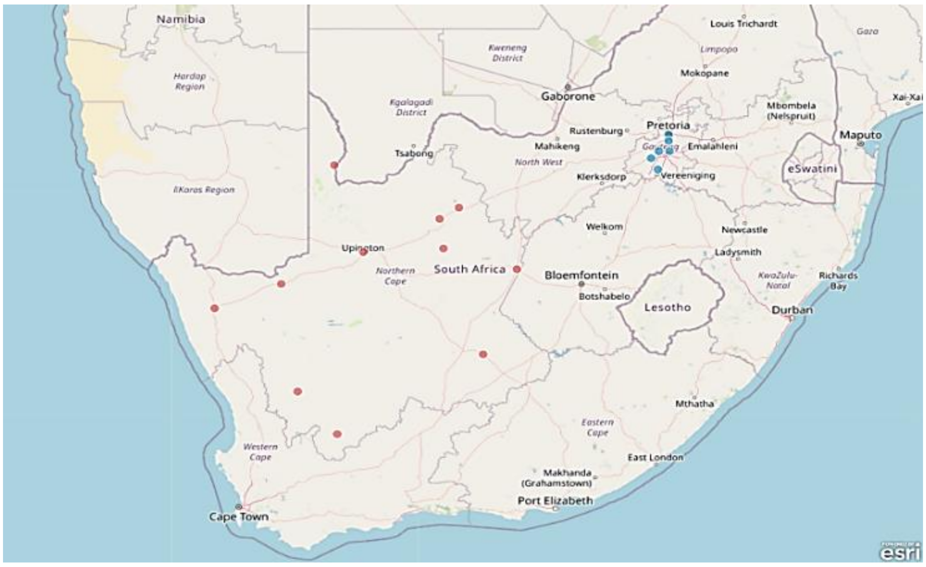

Figure 9 Map of South Africa showing the stations in Gauteng plotted in blue and the Northern Cape in red

Table 3 Dataset description

Dataset	Source	Description
Global historical Climatology Network - Daily (GHCN-Daily) Version 3	NCEI	This dataset contains daily historical climate values including the minimum, maximum, and average temperatures. Period: 1981-2010
Weather stations	SAWS	This dataset contains the spatial attributes of each weather stations such as the station name, province, and geographic coordinates.
Temperatures and Rainfall	SAWS	This dataset contains daily meteorological observations, including maximum, and minimum temperature and total precipitation. Period: 1999-2018
Humidity, pressure, and wind	SAWS	The dataset contains humidity, barometric pressure, wind speed, and wind direction recorded at three different times per day, 8:00 am, 2:00 pm, and 8:00 pm. Period: 1999-2018

Three stations located in Johannesburg, Calvinia and Upington were selected in the experiments, details of each station are summarized in Table 4.

Table 4 Details of selected weather stations

Station Name	Latitude	Longitude	Height (m)	Province
JOHANNESBURG INT WO	-26.14	28.23	1,695	Gauteng
CALVINIA WO	-31.48	19.76	975	Northern Cape
UPINGTON	-28.41	21.26	835	Northern Cape

3.1.2 Maximum Temperature Anomalies

Anomalies are defined in this study as observations that are above (positive) or below (negative) two standard deviations from the daily climate baseline. The process for identifying anomalies was conducted in four steps, which are described below:

1. Calculate daily climate baseline: This step involved calculating daily average temperatures and standard deviation over the reference period (1981-2010) using NCEI's historical dataset.
2. Get daily climate baseline for each observation: In this step, the climate baseline value for each day of the year was combined to corresponding observations.
3. Calculate z-score: In this step, the z-score was calculated for each observation in the dataset based on the climate baseline obtained as described in point 2.
   The z-score, as defined in Eq. (15) in Section 2.3.4 was used to measure the relative location of observations from the daily climatology baseline [9].
4. Classify observations: In this step, anomalous points were identified that was two or more standard deviations from the mean. Therefore, the positive and negative limits for anomalies can be defined as follows [9]:

$$\sigma^{+} = \bar{X} + 2\overline{SD} \tag{16}$$
$$\sigma^{-} = \bar{X} - 2\overline{SD}$$

   Where:
   σ^{+} and σ^{-} are the positive and negative boundaries, respectively,
   $\bar{X}$ is the climate baseline, and
   $\overline{SD}$ is the standard deviation.

The workflow followed to identify anomalies is presented in Figure 10.

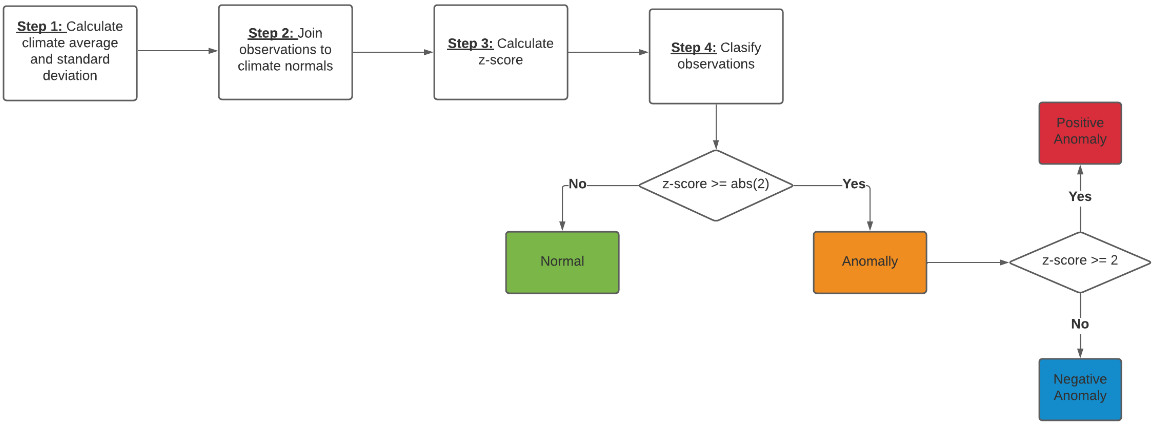

Figure 10 Anomaly identification workflow

The anomaly limits were determined by analysing the distribution of data at each station. A threshold of two standard deviations was chosen because 95% of the maximum temperatures at the selected stations were within 2 standard deviations from the mean, as shown in Figure 11. A summary of maximum temperatures between the observation period (1999-2018) is provided in Table 5.

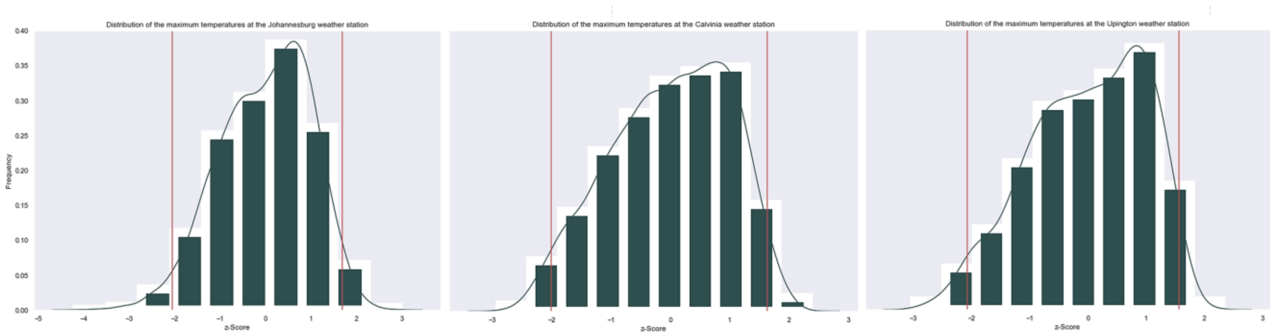

Figure 11 Distribution of maximum temperatures at selected stations (Johannesburg (a), Calvinia (b) and Upington (c))

Table 5 Statistical summary of the maximum temperatures between 1999 and 2018 at Johannesburg, Calvinia and Upington

	Johannesburg		Calvinia		Upington	
	Observed Temperature (°C)	Climate Baseline	Observed Temperature (°C)	Climate Baseline	Observed Temperature (°C)	Climate Baseline
count	7305	7305	7305	7305	7305	7305
mean	22.56	22.86	24.93	25.01	29.82	29.59
std	4.48	3.2	7.29	4.76	6.79	5.35
min	4	6.32	4	6.93	9.6	8.47
25%	19.4	20.07	19.6	20.34	24.8	24.77
50%	23	24.32	25.6	25.4	30.4	30.56
75%	26	25.56	30.8	29.46	35.4	34.58
max	35.6	27.06	41.8	32.99	45.3	37.6

Table 6 shows the number of anomalies in the observations. Refer to Appendix B for a detailed description of the maximum temperatures per station and the number anomalies per year.

Table 6 Number of negative and positive anomalies between 1999 and 2018 for selected stations.

Stations	Negative Anomaly	Positive Anomaly	Total Per Year	Average per Year
Johannesburg	325	39	364	5%
Calvinia	225	46	271	4%
Upington	268	52	320	4%
Total	**818**	**137**	**955**	**4%**

3.2 Problem Definition

A supervised machine learning approach was undertaken using three machine learning techniques, namely ANN, SVM and Huber regressor, based on their performance in related work as described in Section 2.4.6 [3, 8, 9, 11-13, 15]. Both regression and classification-based techniques for anomaly detection were investigated and compared. The comparison between regression and classification-based techniques were performed based on the actual number of anomalies predicted.

Three experiments were conducted to obtain the optimal solution for predicting the maximum temperatures, as shown in the workflow in Figure 12. An assessment and comparison of validation techniques was performed in Experiment 1. The spatial relationships of the data were explored in

Experiment 2. Experiment 3 involved the comparison of regression and classification-based techniques.

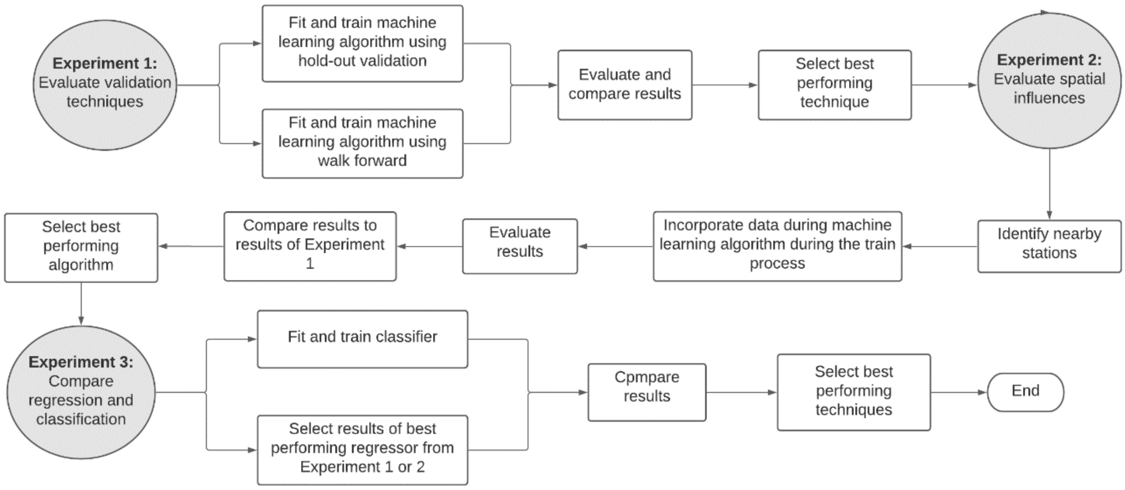

Figure 12 Weather prediction implementation workflow

3.2.1 Prediction Task

The prediction task was approached in two stages for regression-based techniques. First, the machine learning models were trained using the previous three day's weather observations to predict the maximum temperatures for the next day. The input and output features are listed below:

- Input features:
    - Maximum temperature (°C),
    - Minimum temperature (°C) and,
    - Rainfall (mm).
- Output features:
    - Maximum temperature (°C) (next day).

The second stage involved classifying the predicted values labelled as normal or anomalous based on the z-score [9]. The performance of the techniques was evaluated based on:

1. How well next day maximum temperatures were predicted compared to actual temperatures, and
2. How well anomalies were detected in the predictions compared to actual anomalies present in the dataset.

3.2.2 Classification Task

The best performing technique from the regression-based solution was selected. The classification task involved training the machine learning technique with the previous three day's weather observations to categorise whether the next day would be abnormal (label = 0) or normal (label = 1). The input and output features used are listed below:

- Input features:
    - Maximum temperature (°C),
    - Minimum temperature (°C),
    - Rainfall (mm), and
- Output features:
    - Anomaly indicator (Refer to Section 3.1.2 for calculation overview)

The performance of the classifier was evaluated in terms of the number of anomalies predicted and compared to the regression-based technique's results to determine which solution is more suitable for temperature anomaly detection.

3.3 Model development and evaluation

3.3.1 Model Evaluation

The hold-out validation and walk-forward validation techniques were used to evaluate the performance of machine learning techniques in this study. Hold-out validation is one of the most used validation techniques in related work as identified in Section 2.4.3 [3, 9-11, 13-15]. With this technique, the data is split into two parts, the first part is used for training and the remaining part is used for validation [23]. In walk-forward validation, the performance of the technique is evaluated in successive and overlapping training and validation partitions [21, 23, 34]. Refer to Section 2.2.2 for a detailed overview of each technique.

An ideal model will be one that predicts the next day's values with high accuracy and has a high anomaly detection rate. To determine how well the model performed in predicting the next day's values, a combination of three metrics was used. The MAPE was used to express the accuracy of the models' predictions for comparison to other techniques and across stations. A model with a lower MAPE was preferred because it meant that the technique was more accurate. The MAE was used to measure the error in terms of the average difference in degrees Celsius. The MSE was used to determine if there were large outliers in the predictions. In most cases, a model with a high MSE did not fit the data well.

The performance of the technique in terms of anomalies was evaluated using accuracy, precision, recall and F1 score. The main metric used to evaluate the techniques' ability to predict anomalies was the F1 score. The F1 score was used to measure the quality of the predictions by providing a means to determine how accurate the positive predictions were. In addition to the F1 score, precision was used to measure how many anomalies were predicted relative to the total anomalies present in the data and the recall to measure how many of the predicted anomalies were correct. A definition of each metrics is given in Section 2.2.3.

3.3.2 Data Preprocessing Tasks Applied

Data transformations allow machine learning techniques to gain better insight into patterns and trends in the data, leading to better quality results. This section describes the preprocessing activities that were identified and applied to perform the experiments conducted in this study. The preprocessing activities discussed in this section apply to all experiments. Those that are experiment-specific are discussed in their respective sections.

The first task involved filling missing values for each station using linear interpolation. As defined in Section 2.2.4.1, this method involves estimating the value of the missing instance by calculating the average based on the temperatures the day before and after the missing instance [37]. The second task was to create derived features from the data. This included creating the anomaly indicator for classification and calculating the cumulative rainfall. Cumulative rainfall was selected as it resulted in better performance compared to daily values in a study performed by Cramer et al. (2017) [11]. Cumulative rainfall was calculated based on a rolling period of 21 days. The work of Nayak & Ghosh (2013) [9] was leveraged in selecting the window size. In this study, the authors used a centered 21-day mean to predict extreme rainfall.

The sliding window transformation introduced in Section 2.2.4.5 was applied to frame the time series as a supervised learning problem [1, 11]. A lag operator was applied to each observation to obtain its respective predecessor. Each input window consisted of the observations of the previous *n*th day ($y_n = y_{t-1}, y_{t-2}, \ldots, y_{t-n}$) [1, 11, 66]. During the exploratory phase, seven different configurations of the window size were tested. The results of the analysis are presented in Section 4.1. After the windowing was performed, the dataset was first split into a train and validation dataset before normalisation of the data was performed. The data was normalised between 0 and 1 [1, 41].

3.3.3 Machine Learning Model Development

This section provides an overview of how the machine learning techniques selected in this study was implemented. The techniques include the SVR, ANN and Huber Regressor.

3.3.3.1 Artificial Neural Network

The reviewed literature was used as a basis to select the parameter values considered in the experiments performed. A summary of the hyperparameters considered for the ANN is summarized in Table 7. In the literature reviewed, as discussed in Section 2.4.6.1, the hidden layer typically consisted of three nodes [3, 17] or four nodes [13, 15] therefore three and four nodes were considered during the parameter tuning process. The ReLU and logistic activation functions were selected based on the work by [18] and [14, 17]. Adam was selected as the solver as it was found to perform satisfactory by [14, 17]. In total eight combination of hyperparameter settings were evaluated.

Table 7 Hyperparameters used for ANN tuning

Parameter	Range	Increment
Number of Hidden Layers	[1, 2]	1
Number of Nodes	[3, 4]	1
Epochs	[500]	-
Activation	[ReLU, logistic]	-
Solver	[Adam]	-

The network was re-trained ten times to verify that the results were robust [3]. This step was only required for the ANN as the results are arbitrary and may vary across different training runs. This is because the weights and biases of the network are initiated at random and therefore lead to different solutions each time the network is trained. A model that exhibits stability will have small variations in the results across training runs. A detailed discussion of the how the ANN works is provided in Section 2.2.5.1.

3.3.3.2 Support Vector Machine

The hyperparameters configured for SVM were ε and C. These parameters control the model's tolerance to error and its decision function [14, 15]. Lower values of ε are suggested by [14, 15] to avoid large errors and possible underfitting. The results of the hyperparameter tuning performed by [18] showed that lower values ranging between 0.5 and 0.9 for C and 0.001 to 1 for ε resulted in better performance as discussed in Section 2.4.6.2. This study leveraged on the outcomes as per above study and selected a range of lower values for both hyperparameters C and ε. For classification a combination of low and high values of C was evaluated. In total nine combination of hyperparameter settings were evaluated.

Details of the hyperparameters used for regression are given in Table 8 and classification in Table 9. The RBF Kernel was used, as it was found by [12, 18, 19] that it provided superior performance to that of the polynomial kernel. An overview of the SVR is provided in Section 2.2.5.2.

Table 8 Hyperparameters used for the regression based SVM tuning

Parameter	Range	Increment
C	[0.5, 1, 1.5]	0.5
ε	[0.001, 0.01, 1]	0.009

Table 9 Hyperparameters used for the classification based SVM tuning

Parameter	Range	Increment
C	[0.5, 1, 1.5], [10, 15, 20]	[0.5], [5]

3.3.3.3 Huber Regressor

Due to the limitation of studies using Huber Regression, none of the reviewed work could be used to guide the hyperparameter decision making process. Therefore, various hyperparameters were experimented with during the exploratory phase. The default values in Sckit-Learn was used as a basis to start off with. From the analysis it was found that alpha and epsilon had the greatest influence on the model results. Alpha is the regularization parameter that controls the intercept, and the value of Epsilon controls the tolerance of the model to outliers [67].

For this reason, only these two hyperparameters were adjusted. In total, a combination of nine hyperparameter settings were evaluated. Details of the hyperparameters evaluated are summarized in Table 10. A discussion of Huber regression is provided in Section 2.2.5.3.

Table 10 Hyperparameters used for Huber Regressor tuning

Parameter	Range	Increment
Epsilon	[1, 1.5, 2]	0.5
Alpha	[0.00005, 0.0001, 0.00015]	0.00005

3.3.4 Hyperparameter Tuning

As defined in Section 2.2.6, model scaling involves finding the optimal balance between optimization and generalization. This step is required to ensure that the model is statistically powerful to model the problem. This involved iteratively adjusting various configurations until a solution is found that has the right balance between optimization and generalization. The following activities as listed below were investigated in this research [32]:

- Hyperparameter tuning was performed for each station. The advantage of hyperparameter tuning per station ensures that the optimal configuration were used per dataset [11]. The optimal hyperparameters was found through manual adjustments of the different variables as described in Section 3.3.3.
- During the exploration phase, different sets of input and output features were evaluated.
- Different configuration settings such as the window size and fold size were evaluated.
- For ANN different architectures were explored in this study which included evaluating different hidden layers (1 and 2) and hidden layer sizes (3 and 4).

The optimal configuration was selected by monitoring the model error and the impact on the anomaly predictions.

3.4 Experiments

3.4.1 Experiment 1 – Evaluating the Effect of the Validation Technique on Model Performance

The purpose of this experiment was to evaluate the suitability of walk-forward validation for time series data by comparing it to the hold-out method. The machine learning techniques selected in this study (SVR, ANN, and Huber Regressor) were evaluated with each of the validation techniques. The results obtained was compared to determine which provided better performance. The performance was evaluated for each of the selected stations (Johannesburg, Calvinia, and Upington).

A total of 78 machine learning models were evaluated for each validation technique. Thus, a total of 156 machine learning models were developed and evaluated in this experiment. The reported number of models is based on the total number of hyperparameter combinations considered for each technique, as defined in the corresponding subsection in Section 3.3.3. Figure 13 shows a breakdown of the number of machine learning models developed (between square brackets) per validation method, machine learning technique, and station.

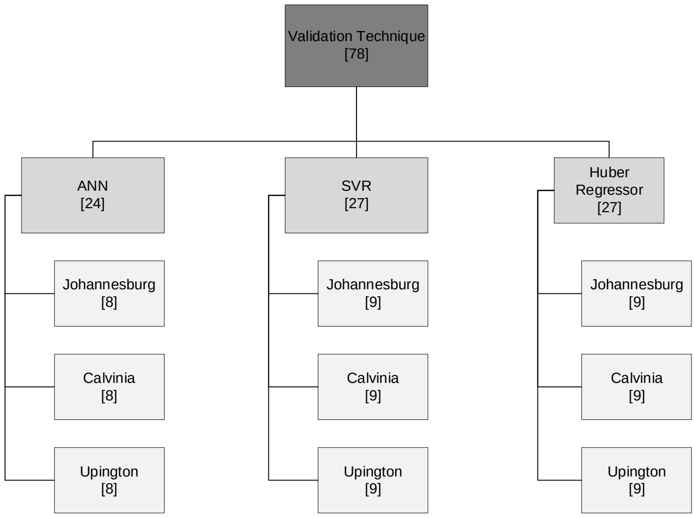

Figure 13 Breakdown of the total number of machine learning models evaluated during the parameter tuning process technique and station

The input features used to predict maximum temperatures consisted of the previous three day's variables as listed below:

- Maximum temperature (°C),
- Minimum temperature (°C) and,
- Accumulated rainfall (mm).

The model output was configured to predict maximum temperature (°C) one day ahead. The predictions were analyzed to identify anomalies detected in the predictions using the process described in Section 3.1.2.

Predictions were compared to actual values to assess the quality of predictions and how well anomalies were detected. Details of the metrics used to measure performance can be found in Section 3.3.1. Implementation details for each validation technique examined in this experiment are discussed in the respective subsections below.

Hold-out Validation

The first step in conducting the experiment was to apply the preprocessing changes discussed in Section 3.3.2. This included filling in missing values, creating derived features, applying the sliding window transformation, and normalizing the data. The data was partitioned into a training and validation sets. The data were split based on a ratio of 80:20, i.e., the first 16 years between 1999 and 2014 were used for training and the remaining data for validation. The testing data consisted of four years of data between 2015 and 2018. Machine learning models were built using the different techniques and tuned manually based on the hyperparameter settings set out in Section 3.3.3 for each of the stations selected. The results were exported to csv files for further analysis.

Walk-Forward Validation

The process for walk-forward validation was the same as with the hold-out validation technique, except for the data partitioning method. For the partition process, a function was constructed that performed stepwise data partitioning during the training process.

The dataset was partitioned into 20 folds consisting of 365 days per fold. Thus, resulting in 19 years of data between 2000 and 2018 being available for validation as the first fold (1999) was used for the first training iteration. A fold size of 20 was selected as it performed computationally efficient and resulted in improved performance during the exploratory phase, refer to Section 4.1.2 for the results of the fold size comparison. The training processed involved iteratively training the technique on each fold. To compare the results to the hold-out validation only the last four folds between 2015 and 2018 was considered during the evaluation and reported on in Chapter Four. The results were exported to csv files for further analysis.

3.4.2 Experiment 2 – Assessing the Spatial Temporal Analysis

In the first experiment, only the temporal attribute of the data was assessed by examining how each data point relates to others over time. In this experiment, both the spatial and temporal attributes were evaluated. To do this, data from nearby stations were included during the training process and examined to see if this had any effect on how well anomalies could be predicted.

To identify nearby stations, the distance between the selected station (JHB Int WO, Calvinia WO, and Upington WO) and surrounding stations is calculated. The station with the least number of kilometres (KM) from the selected station was selected. Table 11 provides an overview of the distance between the stations in KM.

Table 11 Distance in KM between selected station and their respective, the closest station are highlighted in bold

Stations	Johannesburg Int WO		Calvinia WO		Upington WO	
Nearby stations	**JHB BOT TUINE**	**23.55**	**SUTHERLAND**	**132.83**	**POSTMASBURG**	**178.51**
	IRENE WO	25.96	POFADDER	264.66	KATHU	190.61
	PRETORIA UNISA	42.03	SPRINGBOK WO	270.38	POFADDER	198.85
	ZUURBEKOM AWS	45.51	UPINGTON WO	370.86	TWEE RIVIEREN	224.37
	VEREENIGING	54.88	DE AAR WO	413.04	KURUMAN	241.38
			POSTMASBURG	473.08	KIMBERLEY WO	346.12
			KATHU	527.27	SPRINGBOK WO	357.59
			TWEE RIVIEREN	563.19	DE AAR	364.35
			KIMBERLEY WO	565.96	CALVINIA WO	369.83
			KURUMAN	574.38	SUTHERLAND	446.07

The same features used for training were extracted from the dataset of the nearest station and combined with the test stations data used for training. As in experiment one, the following set of input and output features was used:

- Input:
    - Maximum temperature (°C),
    - Minimum temperature (°C), and
    - Accumulated rainfall (mm).
- Output:

- Maximum temperature (°C)

Before combining data from nearby stations, the data were first processed, e.g., by filling in missing data. The optimal hyperparameter settings and validation techniques determined in Experiment 1 were used in this experiment. The experiment was performed using the selected techniques for each of the stations.

A total of nine machine learning models were evaluated. As with Experiment 1, the output of the model (next day maximum temperature) was classified as anomalous or normal using the anomaly classification process described in Section 3.1.2. The results of each technique per station were compared to the corresponding results from Experiment 1 to determine if there was any improvement in the anomaly detection capabilities of the techniques. The results were exported to csv files for further analysis.

3.4.3 Experiment 3 – Assessing the Performance of Classification-based techniques

In experiment three, a comparative analysis of regression- and classification-based techniques was performed. The goal of this experiment was to evaluate and compare different types of techniques based on their ability to detect anomalies to ensure that the most optimal method is selected. The best performing technique identified in Experiments 1 and 2 was selected for this comparison.

A total of 18 classifiers were evaluated for the selected stations, based on the total number of hyperparameter configurations defined in Section 3.3.3. The classification technique was trained to predict whether the maximum temperatures the next day would be normal or abnormal based on the past three days.

The input and output fields used are listed below, for an overview of the anomaly indicator calculation refer to Section 3.1.2.

- Input variable:

    - Maximum temperature (°C),
    - Minimum temperature (°C),
    - Accumulated rainfall (mm), and
    - Anomaly indicator.
- Output variable:
    - Anomaly indicator

For this comparison, the results of the anomaly prediction analysis from Experiment 1 were used. The implementation details are described in Section 3.4.1. The performance of the classifier and the regressor was compared to determine which resulted in better anomaly detection. The results were exported to a csv file for further analysis.

3.5 Implementation

This section describes the tools used and the steps taken to perform the experiments. Python (version 3), Jupyter Notebook and Excel were the main tools used in this study. Various libraries available in Python were used to perform the machine learning, data analysis and visualisation tasks. Among the most important ones are Sckit-Learn, Pandas and Plotly. The sequence of tasks performed during the implementation process is shown in Figure 14 and discussed further below.

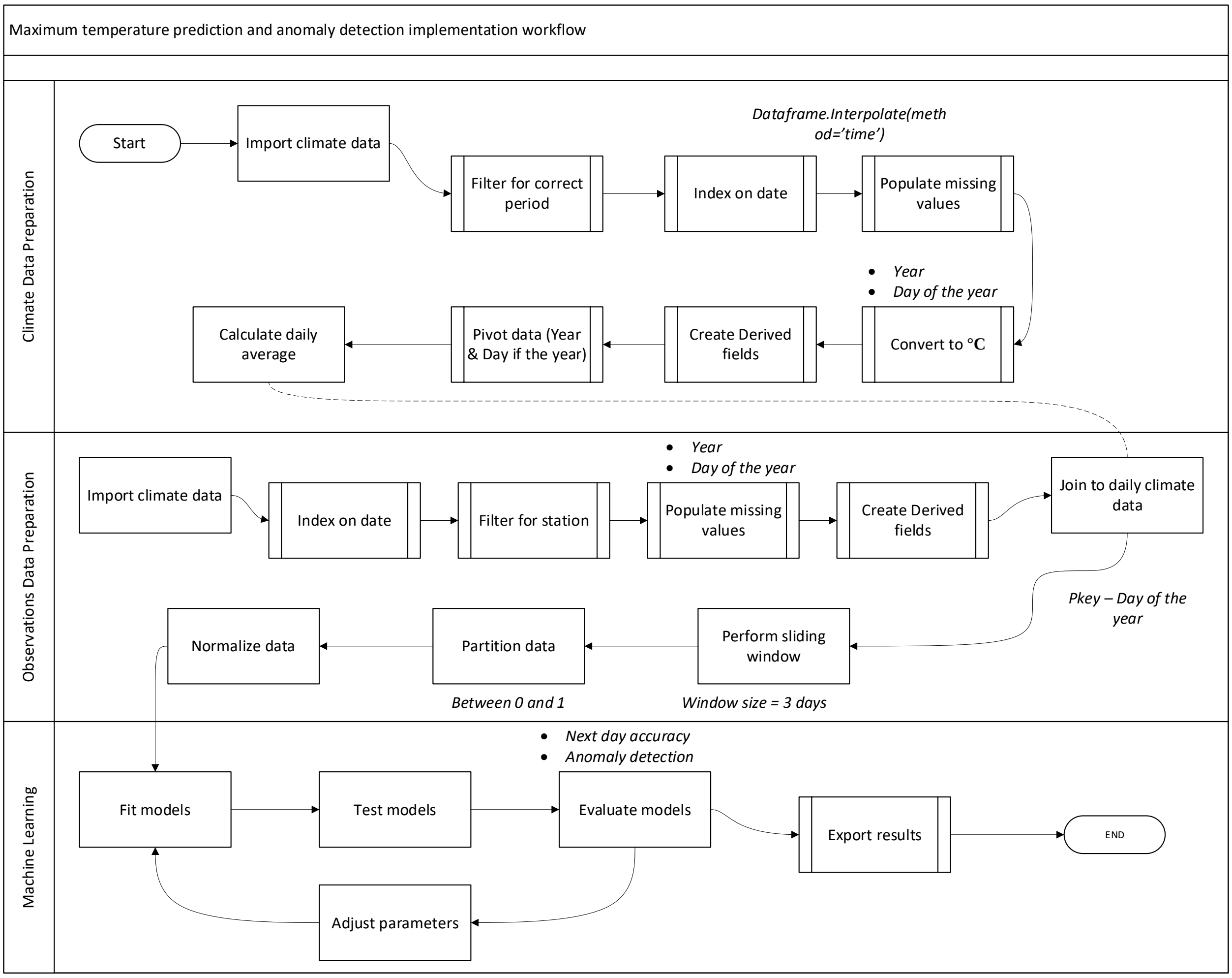

Figure 14 Maximum temperature prediction and anomaly detection implementation workflow

The task was divided into two phases, the first involved classifying anomalies and the second, the machine learning process. The first phase was performed only once for each station to prepare the data files used. The second phase was executed for each of the experiments performed.

The first step was to import the climate data into a Pandas DataFrame. A filter was applied to ensure that only observations between 1981 and 2010 were included. The dataset was indexed by date to allow for time-based interpolation. The temperature values in the dataset were in tenths of a degree Celsius, therefore the values had to be converted to degrees Celsius (maximum temperature/10). The day of the year and the year were derived from the date column. These fields were created to pivot the dataset. The climate DataFrame was pivoted by the day of year and year. The average for each day of the year was calculated to obtain the climate baseline for each day.

Daily weather observations were imported into a DataFrame and indexed by date. A filter was applied to extract the appropriate station data. Missing data were filled using time-based interpolation. The day of the year was derived from the dataset. This field was used to join to the climate values so that each day in the observations had a corresponding climate baseline value. The z-score, as defined in Eq. (15), was used to calculate how much the observed value differs from the climate norm. Based on the z-score, each datapoint was labelled as anomalous or normal based on the limits defined in Eq. (16). The anomalies were further classified as positive or negative anomalies based on the movement of the deviation. The final prepared DataFrame was exported to a csv file.

The next phase involved training the models. The first task was to read the data for each station into a DataFrame and index it on the date. A custom function was developed to perform the window operation to obtain the previous three days values. Feature extraction and data partitioning was performed using Panda's data partitioning method, which specifies the row and column index. Normalisation of the data was performed using the MinMaxScaler function. The lower and upper limits were set to 0 and 1, respectively.

The SVR, MLPRegressor, and HuberRegressor techniques, available through the Scikit-Learn library, were used to fit the models. For walk-forward validation, a function was developed to perform stepwise partitioning to obtain the training and test data sets. The result of each fold was appended to a DataFrame. The results were calculated using the built-in functions for MAE and MSE, as well as user-defined functions for MAPE, RMSE, accuracy, precision, recall, and F1 score. Results were exported to a csv file for further analysis.

3.6 Conclusion

The maximum temperature prediction problem was formulated as a supervised machine learning problem. The techniques considered in the experiments performed were SVM, ANN and Huber

Regression, since they have been shown to be satisfactory in related work. The goal of the experiments conducted was to determine the best approach for predicting temperature anomalies using machine learning. Therefore, the following aspects were evaluated: the suitability of the validation techniques, the influence of spatial relationships, and the type of techniques used.

The performance was evaluated using MAPE, MAE, MSE, and RMSE for the predictions and the accuracy, precision, recall, and F1 score for classification. The results of each technique were evaluated based on their ability to predict the next day's values and detect anomalies.

Chapter 4

4. Results

Chapter 3 discussed how the experiments were conducted. In this section, the results of the experiments are presented. Three experiments were conducted to find the ideal solution for detecting maximum temperature anomalies. The first experiment compared the hold-out and walk-forward validation technique. The second experiment dealt with the spatial relationships between stations. The third and final experiment compared regression and classification techniques for anomaly detection. The results of each experiment are presented in the respective subsections.

4.1 Experiment 1: Results of the Different Validation Techniques

The first experiment compared the performance of machine learning techniques evaluated based on two different validation techniques. The results of hold-out validation are presented first, followed by the results of walk-forward validation. The techniques were trained to predict the maximum temperatures for the next day based on the conditions of the previous three days. The predictions of the models were evaluated to assess the anomaly detection capabilities.

The optimal window size was determined during the exploratory phase by testing seven different window sizes based on the default hyperparameter values given in Scikit-Learn. The results obtained are shown in Table 12, the best performing window size are highlighted in bold. A window size of three resulted in higher anomaly detection rates and was therefore adopted throughout the study.

Table 12 Average next day error (MAPE) and anomaly prediction rate (F1-score) across the testing stations for each window sizes tested using the SVR, Huber regressor and ANN.

Average MAPE							
Window Size	**1**	**3**	**5**	**7**	**10**	**14**	**21**
ANN	12.14	**11.83**	11.66	11.67	11.58	11.62	11.72
Huber Regressor	12.19	**11.96**	11.73	11.72	11.64	11.60	11.58
SVR	12.19	**11.89**	11.70	11.69	11.65	11.69	11.72
Overall Average	**12.17**	**11.89**	**11.70**	**11.69**	**11.63**	**11.64**	**11.67**
Average F1-score							
ANN	0.11	**0.12**	0.10	0.06	0.07	0.07	0.07
Huber Regressor	0.13	**0.12**	0.07	0.05	0.06	0.07	0.08
SVR	0.09	**0.12**	0.08	0.07	0.07	0.06	0.05
Overall Average	**0.11**	**0.12**	**0.08**	**0.06**	**0.06**	**0.07**	**0.07**

Hyperparameter tuning was performed to determine which settings were most appropriate for weather data. The MAPE was used to evaluate predictions and the F1 score was used to evaluate classification. Hyperparameter tuning was performed using the process described in Section 3.3.4. Based on the ranges of hyperparameters discussed in Section 3.3.3, nine different configurations were evaluated for the SVR and Huber regressor and eight for the ANN.

4.1.1 Hold-out Validation

A partitioning ratio of 80:20 was applied to the dataset. Thus, the validation data consisted of 1461 observations between 2015 and 2018 (4 years). Table 13 shows the number of normal and anomalous instances by station and year; an overview of the entire dataset can be found in Appendix B.

Table 13 Total number of anomalies by station and year during the test period (2015-2018)

Station	Anomaly Type	2015	2016	2017	2018	Total
JHB	**Negative Anomalies**	9	11	16	8	**44**
	Positive Anomalies	12	4	1	9	**26**
	Normal Instances	344	351	348	348	**1391**
CAL	**Negative Anomalies**	10	7	9	12	**38**
	Positive Anomalies	3	5	1	7	**16**
	Normal Instances	352	354	355	346	**1407**
UPT	**Negative Anomalies**	16	7	10	13	**46**
	Positive Anomalies	10	6	4	5	**25**
	Normal Instances	339	353	351	347	**1390**

Tables 14 to 16 show the results of hyperparameter tuning performed for the SVR for each of the stations tested, with the best combinations highlighted in bold. Figures 15 to 17 show the prediction and classification performance for each of the combinations. The black solid line shows the MAPE, the red dashed line shows the F1-score for positive anomalies and the blue dotted line shows the F1 score for negative anomalies.

Smaller values of ε resulted in an increase in prediction and classification accuracy. The hyperparameter C, on the other hand, had an insignificant effect on the results. The following parameters were found to be optimal for SVR: Johannesburg: (ε :0.001, C:1.5), Calvinia: (ε :0.001, C:1), Upington: (ε :0.01, C:1).

Table 14 Results of the SVR hold-out hyperparameter tuning for Johannesburg

Config	E	C	MAPE	F1(+)	F1(-)
1	0.001	0.5	8.34	0.14	0.32
2	0.001	1	8.34	0.14	0.32
3	**0.001**	**1.5**	**8.35**	**0.14**	**0.34**
4	0.01	0.5	8.36	0.09	0.34
5	0.01	1	8.34	0.09	0.34
6	0.01	1.5	8.34	0.09	0.33
7	1	0.5	21.63	0.12	0.15
8	1	1	21.63	0.12	0.15
9	1	1.5	21.63	0.12	0.15

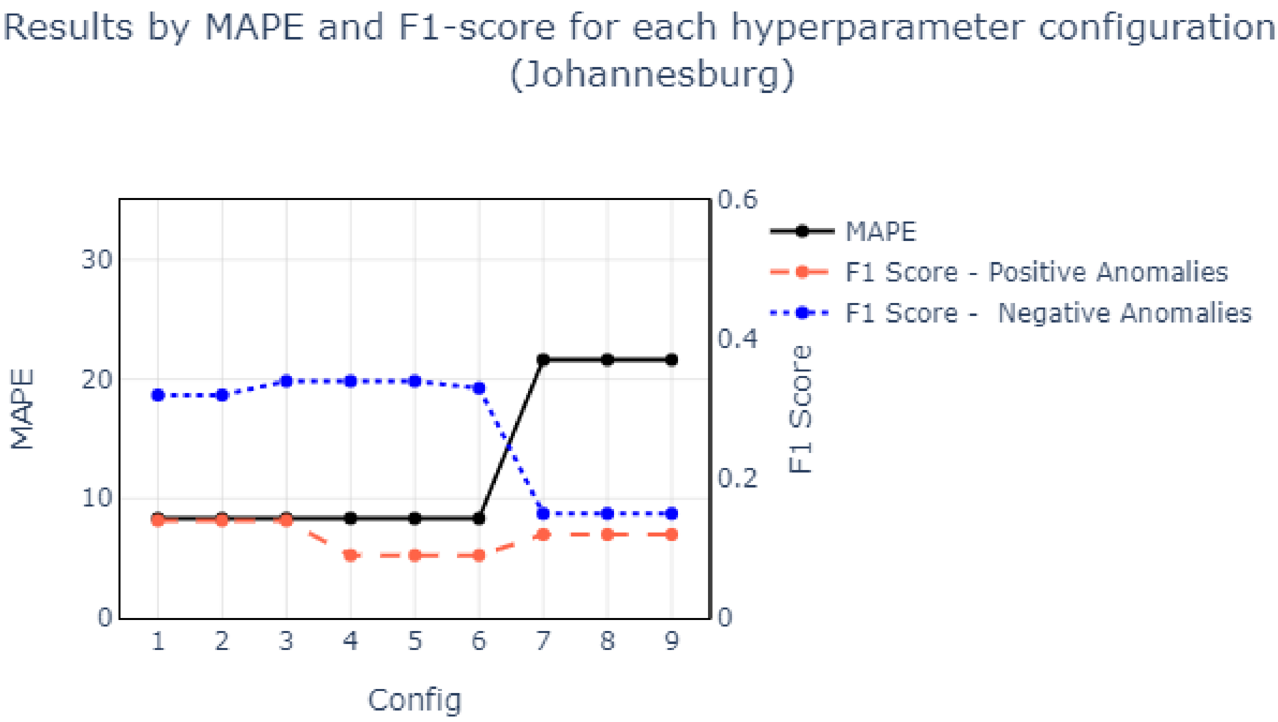

Figure 15 Results of the SVR hold-out hyperparameter tuning for Johannesburg

Table 15 Results of the SVR hold-out hyperparameter tuning for Calvinia

Config	E	C	MAPE	F1(+)	F1(-)
1	0.001	0.5	15.28	0.08	0.12
2	**0.001**	**1**	**15.27**	**0.09**	**0.17**
3	0.001	1.5	15.3	0.08	0.16
4	0.01	0.5	15.3	0.09	0.13
5	0.01	1	15.29	0.09	0.17
6	0.01	1.5	15.31	0.09	0.17
7	1	0.5	27.6	0	0.02
8	1	1	27.6	0.00	0.02
9	1	1.5	27.6	0	0.02

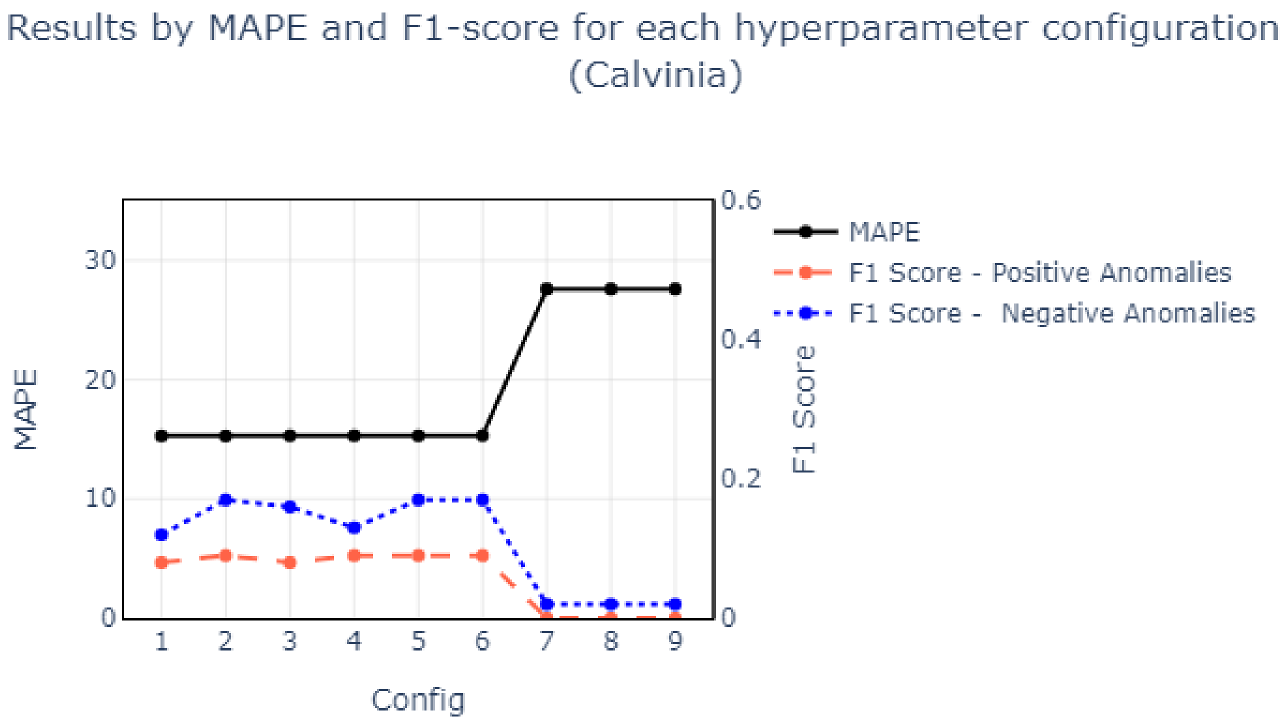

Figure 16 Results of the SVR hold-out hyperparameter tuning for Calvinia

Table 16 Results of the SVR hold-out hyperparameter tuning for Upington

Config	ε	C	MAPE	F1(+)	F1(-)
1	0.001	0.5	9.59	0.06	0.2
2	0.001	1	9.58	0.06	0.2
3	0.001	1.5	9.57	0.06	0.2
4	0.01	0.5	9.58	0.06	0.2
5	**0.01**	**1**	**9.56**	**0.06**	**0.21**
6	0.01	1.5	9.55	0.06	0.21
7	1	0.5	20.93	0.06	0.05
8	1	1	20.93	0.06	0.05
9	1	1.5	20.93	0.06	0.05

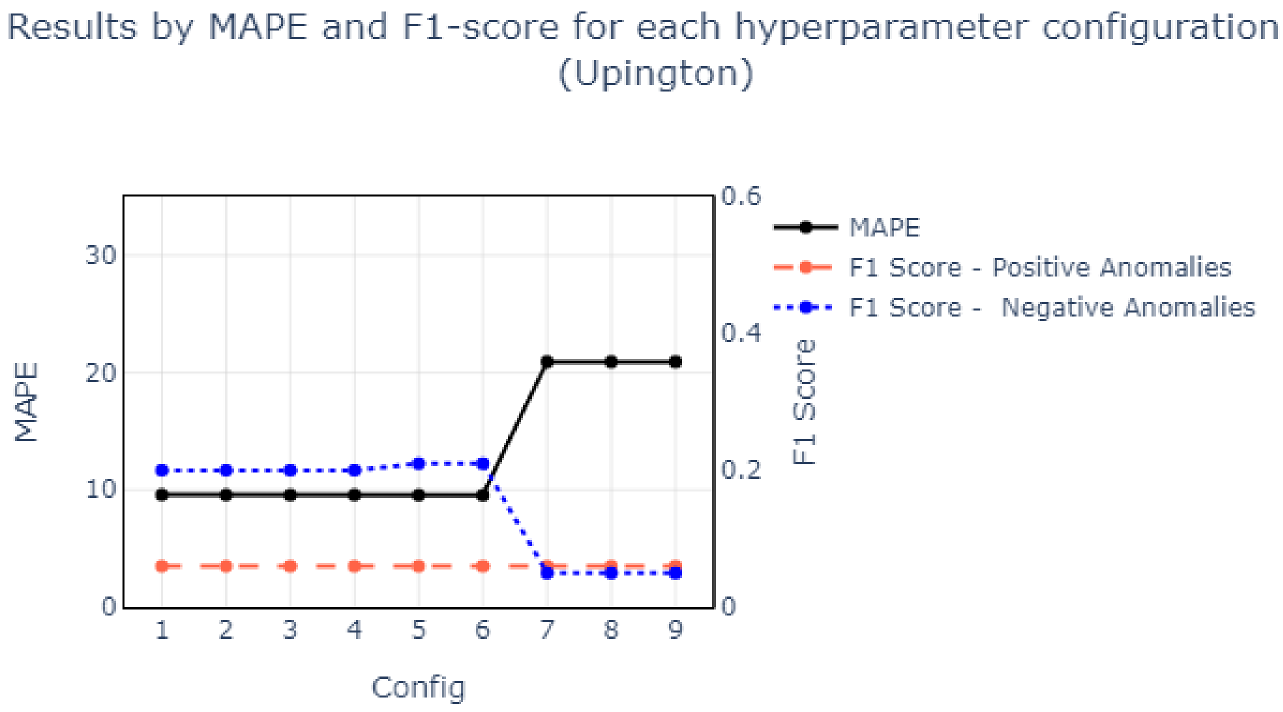

Figure 17 results of the SVR hold-out hyperparameter tuning for Upington

Table 17 to 19 show the results of hyperparameter tuning performed for the ANN for each of the stations tested, with the best combinations highlighted in bold. Figures 18 to 20 show the prediction and classification performance for each of the combinations. The optimal architecture consisted of a network with an input layer, two hidden layers with four nodes each, and an output layer. When using the Adam solver, the ReLU activation function provided the most optimal results.

Table 17 Results of the ANN hold-out hyperparameter tuning for Johannesburg

Config	Layer	Size	Activation	MAPE	F1(+)	F1(-)
1	1	3	ReLU	16.17	0.09	0.09
2	2	3	ReLU	16.43	0.09	0.09
3	1	4	ReLU	15.2	0.08	0.11
4	**2**	**4**	**ReLU**	**14.84**	**0.09**	**0.15**
5	1	3	logistic	17.47	0.09	0.1
6	2	3	logistic	17.31	0.08	0.09
7	1	4	logistic	17.39	0.09	0.09
8	2	4	logistic	17.32	0.09	0.08

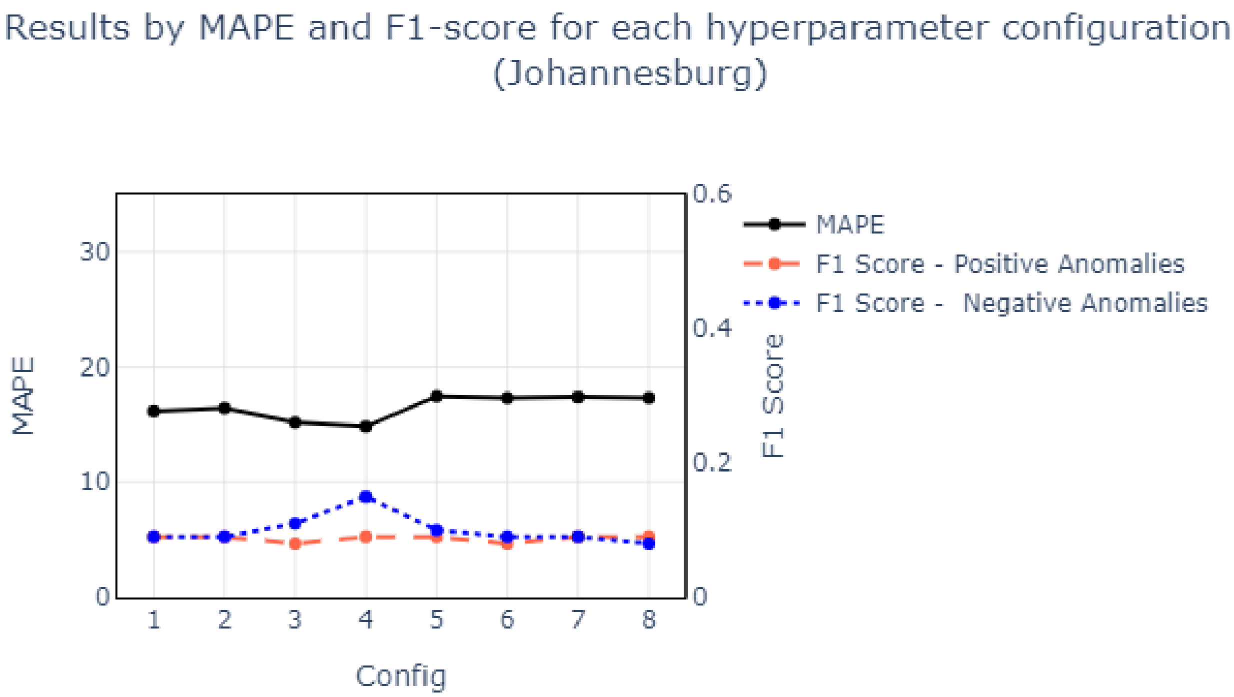

Figure 18 Results of the ANN hold-out hyperparameter tuning for Johannesburg

Table 18 Results of the ANN hold-out hyperparameter tuning for Calvinia

Config	Layer	Size	Activation	MAPE	F1(+)	F1(-)
1	1	3	ReLU	26.34	0.06	0.05
2	2	3	ReLU	24.78	0.1	0.07
3	1	4	ReLU	25.14	0.11	0.06
4	**2**	**4**	**ReLU**	**22.2**	**0.09**	**0.06**
5	1	3	logistic	28.08	0	0
6	2	3	logistic	27.58	0	0
7	1	4	logistic	28.04	0	0
8	2	4	logistic	27.57	0	0

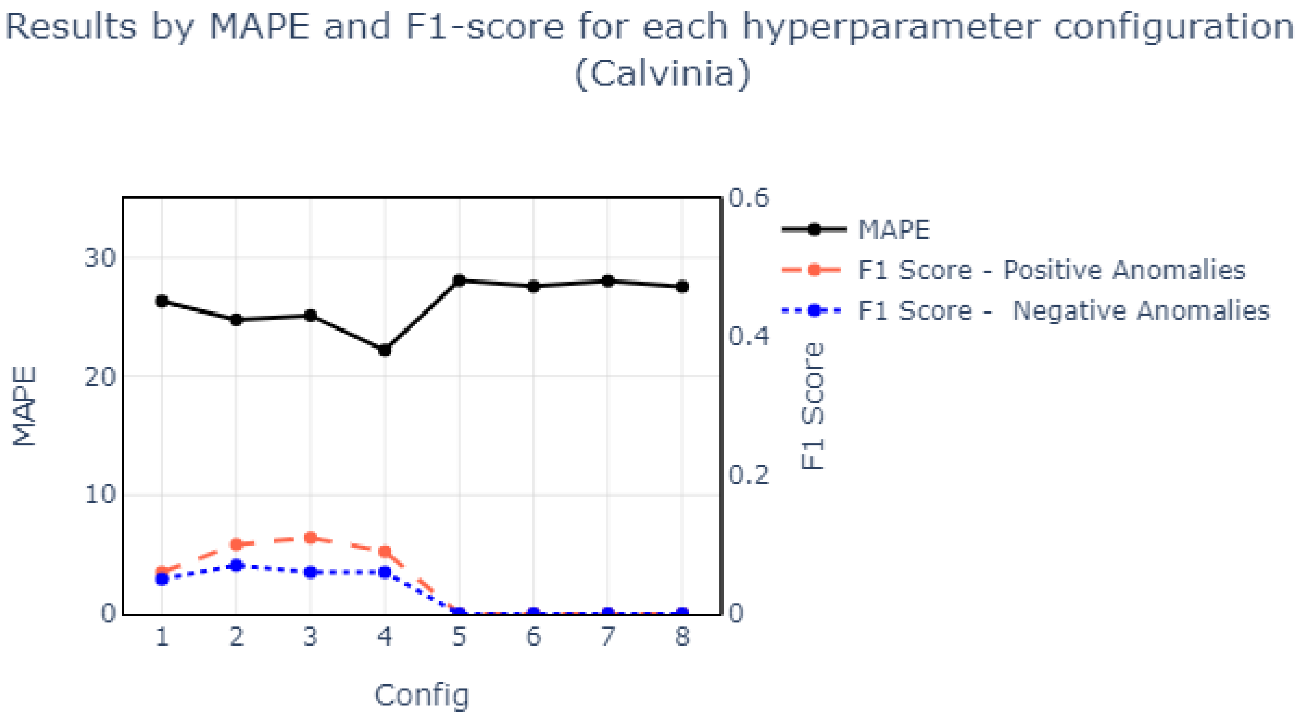

Figure 19 Results of the ANN hold-out hyperparameter tuning for Calvina

Table 19 Results of the ANN hold-out hyperparameter tuning for Upington

Config	Layer	Size	Activation	MAPE	F1(+)	F1(-)
1	1	3	ReLU	18.71	0.04	0.06
2	2	3	ReLU	16.67	0.05	0.09
3	1	4	ReLU	17.95	0.05	0.06
4	**2**	**4**	**ReLU**	**15.55**	**0.04**	**0.05**
5	1	3	logistic	20.42	0.05	0.06
6	2	3	logistic	20.14	0.06	0.06
7	1	4	logistic	20.39	0.05	0.06
8	2	4	logistic	20.07	0.06	0.06

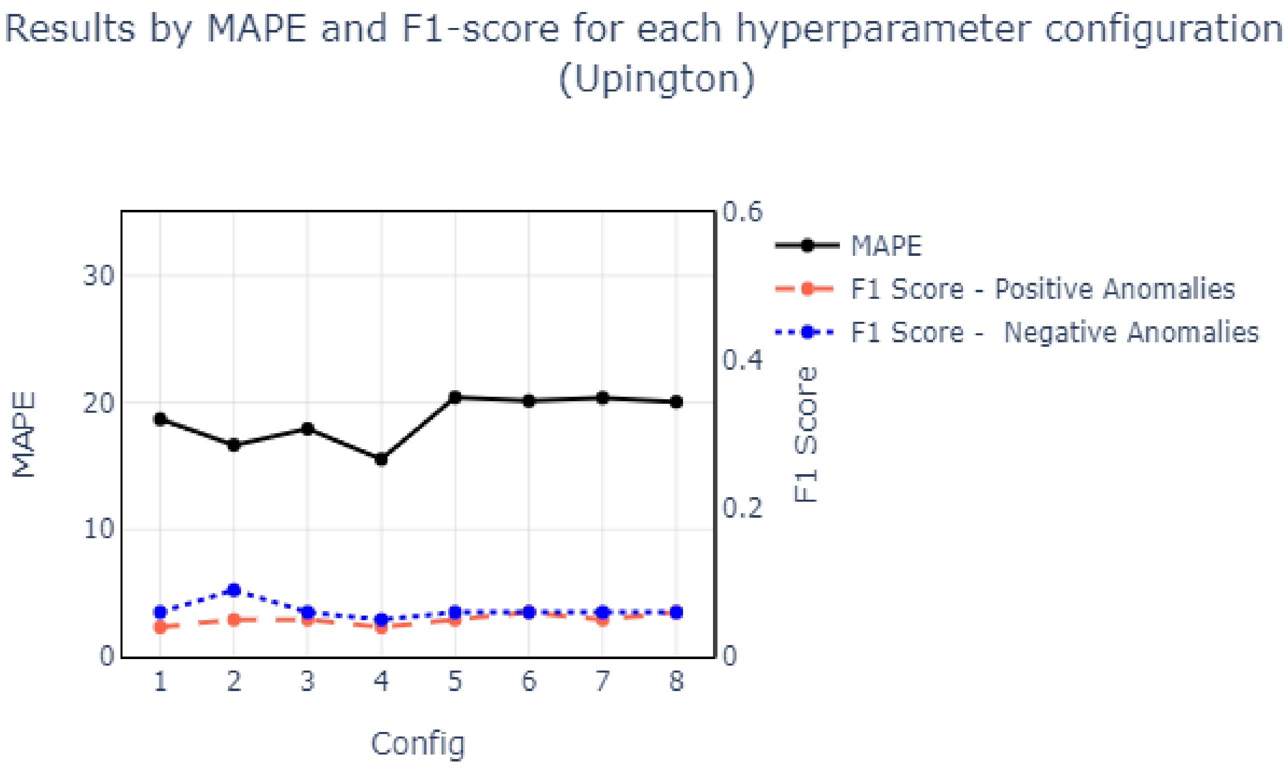

Figure 20 Results of the ANN hold-out hyperparameter tuning for Upington

Table 20 to 22 show the results of hyperparameter tuning performed for Huber regression for each of the stations tested, with the best combinations highlighted in bold. The classification and performance for each hyperparameter combination are shown in Figures 21 to 23. Lower values of epsilon resulted in better prediction and classification accuracy. Alpha had an insignificant effect on the results. The most optimal parameters for Huber regression were Johannesburg: (Epsilon: 2, Alpha: 0.00005), and Calvinia and Upington: (Epsilon: 1, Alpha: 0.0015).

Table 20 Results of the Huber Regressor hold-out hyperparameter tuning for Johannesburg

Config	Epsilon	Alpha	MAPE	F1(+)	F1(-)
1	1	0.00005	8.75	0.24	0.31
2	1	0.0001	8.75	0.24	0.31
3	1	0.00015	8.75	0.24	0.31
4	1.5	0.00005	8.74	0.24	0.31
5	1.5	0.0001	8.74	0.24	0.31
6	1.5	0.00015	8.74	0.24	0.31
7	**2**	**0.00005**	**8.74**	**0.24**	**0.32**
8	2	0.0001	8.74	0.24	0.32
9	2	0.00015	8.74	0.24	0.32

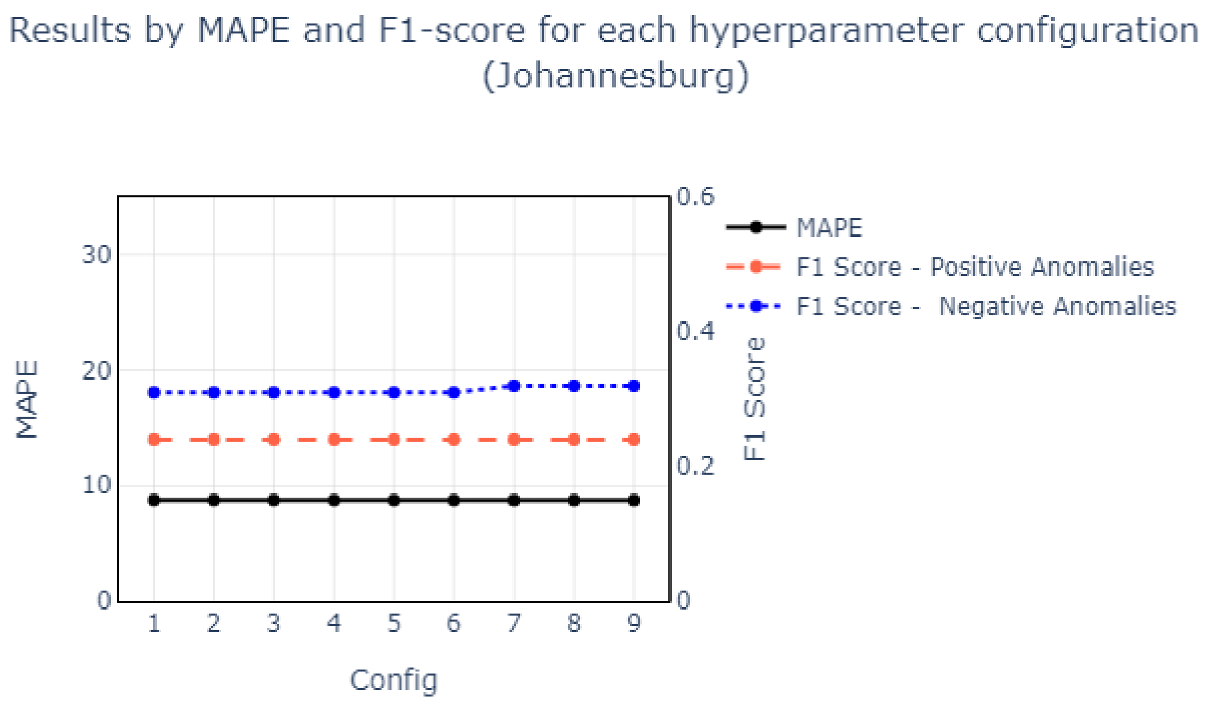

Figure 21 Results of the Huber Regressor hold-out hyperparameter tuning for Johannesburg

Table 21 Results of the Huber Regressor hold-out hyperparameter tuning for Calvinia

Config	Epsilon	Alpha	MAPE	F1(+)	F1(-)
1	1	0.00005	15.55	0.14	0.19
2	1	0.0001	15.55	0.14	0.19
3	**1**	**0.00015**	**15.55**	**0.14**	**0.19**
4	1.5	0.00005	15.41	0.08	0.19
5	1.5	0.0001	15.41	0.08	0.19
6	1.5	0.00015	15.41	0.08	0.19
7	2	0.00005	15.33	0.08	0.19
8	2	0.0001	15.33	0.08	0.19
9	2	0.00015	15.33	0.08	0.19

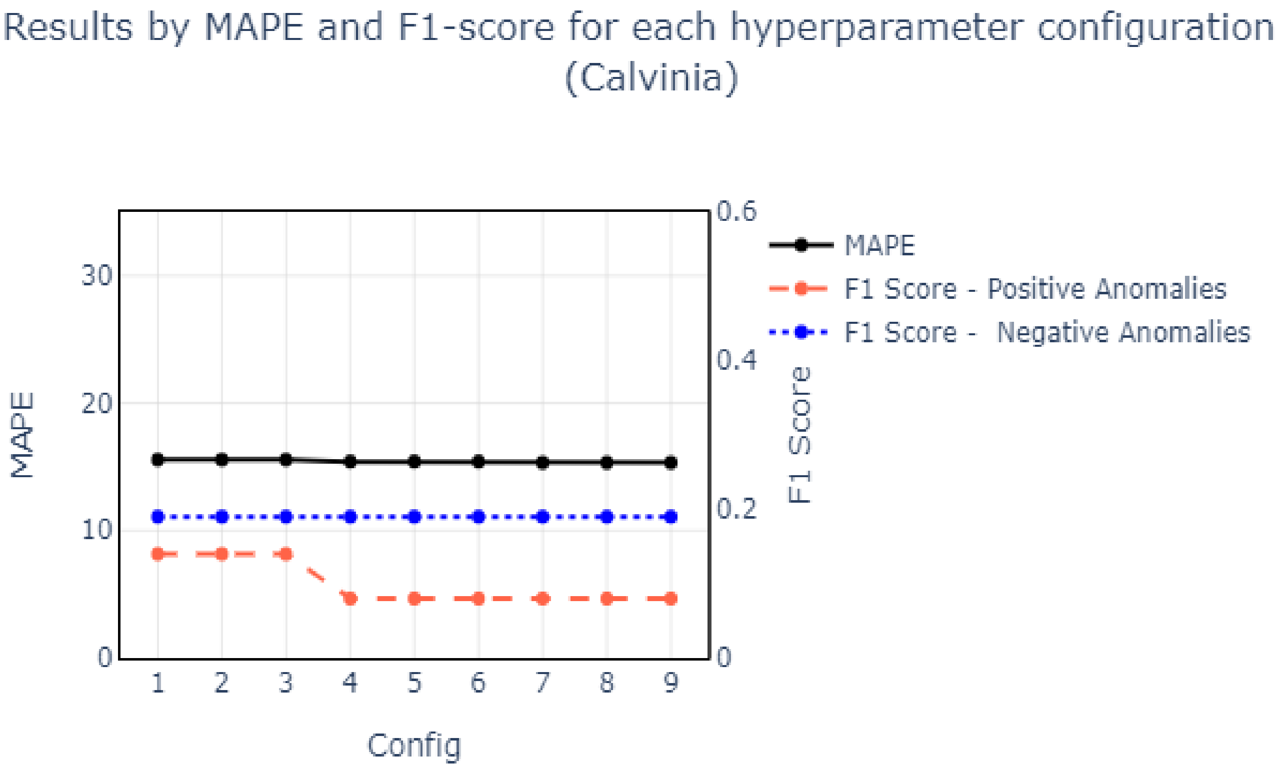

Figure 22 Results of the Huber Regressor hold-out hyperparameter tuning for Calvina

Table 22 Results of the Huber Regressor hold-out hyperparameter tuning for Upington

Config	Epsilon	Alpha	MAPE	F1(+)	F1(-)
1	1	0.00005	9.75	0.10	0.18
2	1	0.0001	9.75	0.10	0.18
3	**1**	**0.00015**	**9.75**	**0.10**	**0.18**
4	1.5	0.00005	9.73	0.06	0.19
5	1.5	0.0001	9.73	0.00	0.19
6	1.5	0.00015	9.76	0.00	0.20
7	2	0.00005	9.76	0.00	0.20
8	2	0.0001	9.76	0.00	0.20
9	2	0.00015	9.76	0.00	0.20

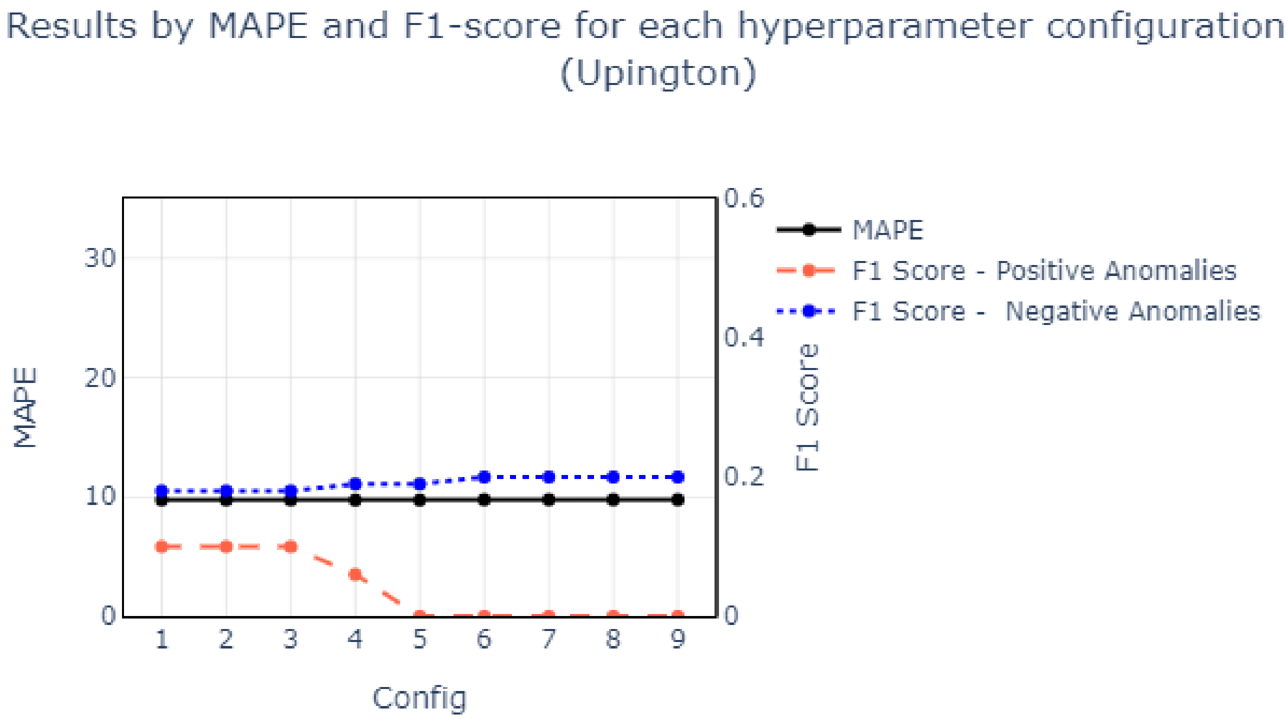

Figure 23 Results of the Huber Regressor hold-out hyperparameter tuning for Upington

The prediction results of the most optimal configuration are presented in Table 23. A lower score for the predictions and a high score for classification indicate better performance. The SVR performed most satisfactorily across all metrices, while ANN performed least satisfactorily. The results show that the predictions in degrees Celsius at certain stations (e.g., Johannesburg and Upington) are much closer to the actual temperature, as evidenced by a lower MAPE. However, based on the MSE, there are large outliers in the predictions, with the Calvinia station having the largest error among the stations tested.

Table 23 Prediction results for hold-out validation by technique and station - Johannesburg (JHB), Calvinia (CAL) and Upington (UPT).

		Next day prediction			
Technique	**Station**	**MAPE**	**MAE**	**MSE**	**RMSE**
SVR	JHB	8.35	1.78	5.89	2.43
	CAL	15.27	3.15	17.50	4.18
	UPT	9.56	2.53	11.87	3.44
Average		**11.06**	**2.49**	**11.75**	**3.35**
ANN	JHB	4.66±2.25	21.14±10.32	30.42±8.5	5.52±2.92
	CAL	6.01±3.72	28.23±17.96	51.59±22.18	7.18±4.71
	UPT	5.65±2.84	20.65±10.47	45.52±13.43	6.75±3.66
Average		**17.53**	**4.08**	**26.43**	**5.00**
Huber Regressor	JHB	8.74	1.87	6.19	2.49
	CAL	15.55	3.20	18.02	4.24
	UPT	9.75	2.59	12.27	3.50
Average		**11.35**	**2.55**	**12.16**	**3.41**

Table 24 shows the result of the anomaly classification and Table 25 shows the metric score by technique and station. It should however be noted that due to the imbalance of positive to negative anomalies in the datasets, the results of the accuracy metric are not a true reflection of the performance of the models. The results show that the techniques had a low anomaly detection rate. The techniques were better at modelling negative anomalies (conditions below the climate norm). In addition, the techniques had a high rate of incorrect predictions relative to the rate of correct predictions. The technique that performed best in both the number of positive and negative anomalies was the Huber regressor, followed by the SVR.

Table 24 Count of TP, TN, FN, and FP predicted using hold-out validation for each anomaly type by technique and station

		Positive anomalies			Negative anomalies			TN
Technique	**Station**	**TP**	**FN**	**FP**	**TP**	**FN**	**FP**	
SVR	**JHB**	3	23	2	11	33	12	1377
	CAL	1	15	3	4	34	3	1401
	UPT	1	24	1	6	40	4	1385
Average		**1**	**21**	**2**	**7**	**36**	**6**	**1387.67**
ANN	**JHB**	12 ± 1	25 ± 14	181 ± 3	15 ± 1	43 ± 29	73 ± 1	1366±1128
	CAL	2 ± 0	16 ± 14	17 ± 2	4 ± 0	38 ± 34	31 ± 1	1404±1356
	UPT	7 ± 0	25 ± 18	132 ± 1	10 ± 0	46 ± 36	131 ± 2	1386±1127
Max/min		**12 ± 0**	**25 ± 14**	**132 ± 1**	**15 ± 0**	**46 ± 29**	**131 ± 4**	**1404±1127**
Huber regressor	**JHB**	6	20	5	11	33	12	1374
	CAL	2	14	5	5	33	6	1396
	UPT	2	23	6	6	40	8	1376
Average		**3**	**19**	**5**	**7**	**35**	**9**	**1382.00**

Table 25 Classification results for hold-out validation by technique and station where (A) is accuracy, (P) precision, (R) recall and (F1) the F1-score.

		Positive anomalies				Negative anomalies			
Technique	**Station**	**A**	**P**	**R**	**F1**	**A**	**P**	**R**	**F1**
SVR	**JHB**	0.97	0.18	0.12	0.14	0.97	0.46	0.27	0.34
	CAL	0.99	0.14	0.06	0.09	0.97	0.4	0.11	0.17
	UPT	0.98	0.17	0.04	0.06	0.97	0.55	0.13	0.21
Average		**0.98**	**0.16**	**0.07**	**0.1**	**0.97**	**0.47**	**0.17**	**0.24**
ANN	**JHB**	0.97±0.81	0.14±0.05	0.46±0.04	0.17±0.05	0.96±0.8	0.39±0.03	0.34±0.02	0.34±0.03
	CAL	0.99±0.96	0.17±0.00	0.13±0.00	0.11±0.06	0.97±0.94	0.33±0.00	0.11±0.00	0.1±0.04
	UPT	0.98±0.8	0.09±0.00	0.28±0.00	0.06±0.02	0.97±0.79	0.13±0.00	0.22±0.00	0.07±0.02
Max/min		**0.99±0.8**	**0.17±0.00**	**0.46±0.00**	**0.17±0.02**	**0.97±0.79**	**0.39±0.00**	**0.34±0.00**	**0.34±0.02**
Huber Regressor	**JHB**	0.97	0.24	0.23	0.24	0.96	0.39	0.27	0.32
	CAL	0.98	0.15	0.13	0.14	0.97	0.31	0.13	0.19
	UPT	0.97	0.13	0.08	0.1	0.96	0.3	0.13	0.18
Average		**0.97**	**0.17**	**0.15**	**0.16**	**0.96**	**0.33**	**0.18**	**0.23**

4.1.2 Walk-Forward Validation

An exploratory analysis was performed to identify optimal number of folds for walk-forward validation. From the exploratory analysis, a partitioning of 20 folds consisting of 365 days per fold resulted in better performance and was therefore used throughout this study. As can be seen in Figure 24, a lower MAPE and higher F1 score is observed. The results can be found in Table 26. Based on 20 folds, 19 iterations of training (1999-2017) and 19 iterations of validation (2000-2018) were performed. For comparison with the hold-out validation results, as described in Section 4.1.1, only the last four training iterations (2015-2018) are reported in the results that follow.

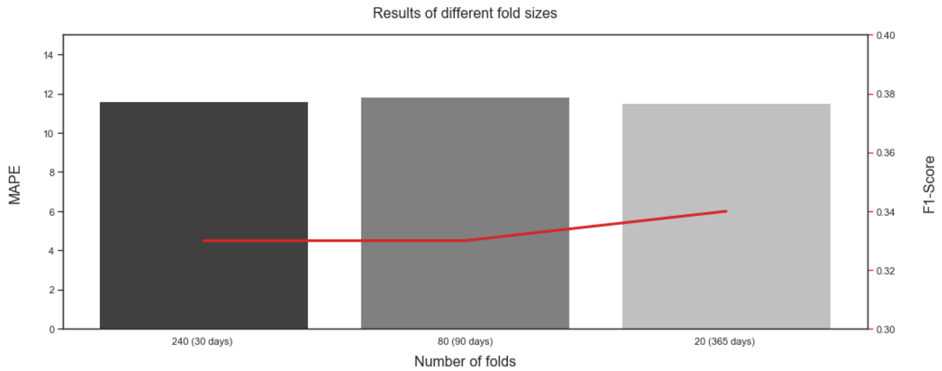

Figure 24 Comparison of walk-forward fold sizes for temperature prediction and anomaly detection

Table 26 Results of the walk-forward fold size comparison

Number of folds	Fold Size	MAPE	MAE	MSE	RMSE	Accuracy	Precision	Recall	F1score
240	**30**	11.61	2.53	9.42	2.99	0.8	0.34	0.33	0.33
80	**90**	11.83	2.57	9.79	3.07	0.8	0.33	0.33	0.33
20	**365**	**11.52**	**2.51**	**9.32**	**3.05**	**0.8**	**0.34**	**0.33**	**0.34**

Table 27 Results of the SVR walk-forward hyperparameter tuning for Johannesburg

Config	ε	C	MAPE	F1(+)	F1(-)
1	0.001	0.5	8.35	0.38	0.26
2	**0.001**	**1**	**8.36**	**0.38**	**0.27**
3	0.001	1.5	8.36	0.32	0.31
4	0.01	0.5	8.36	0.33	0.28
5	0.01	1	8.35	0.33	0.28
6	0.01	1.5	8.35	0.33	0.28
7	1	0.5	20.72	0.17	0.14
8	1	1	20.72	0.17	0.14
9	1	1.5	20.72	0.17	0.14

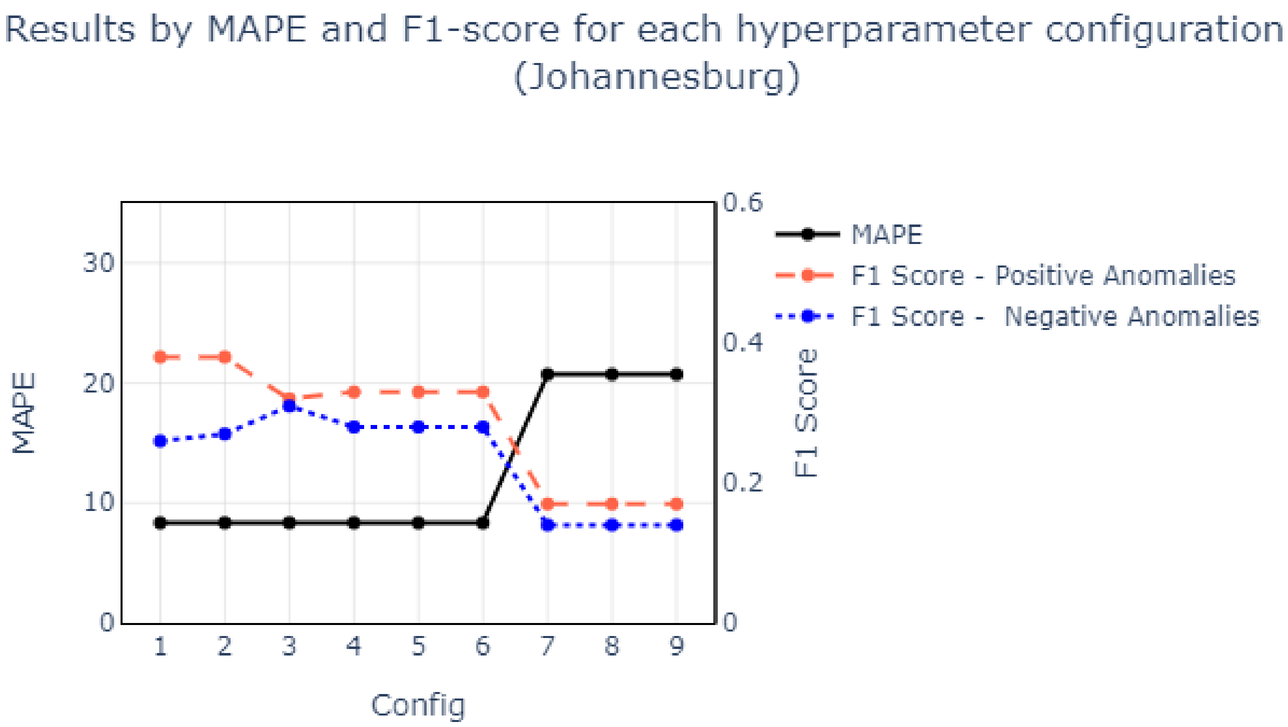

Figure 25 Results of the SVR walk-forward hyperparameter tuning for Johannesburg

Table 28 Results of the SVR walk-forward hyperparameter tuning for Calvinia

Config	E	C	MAPE	F1(+)	F1(-)
1	0.001	0.5	15.26	0.5	0.32
2	0.001	1	15.26	0.5	0.33
3	0.001	1.5	15.29	0.5	0.33
4	0.01	0.5	15.26	0.5	0.32
5	**0.01**	**1**	**15.27**	**0.5**	**0.33**
6	0.01	1.5	15.31	0.5	0.33
7	1	0.5	27.61	0.00	0.08
8	1	1	27.61	0.00	0.08
9	1	1.5	27.61	0.00	0.08

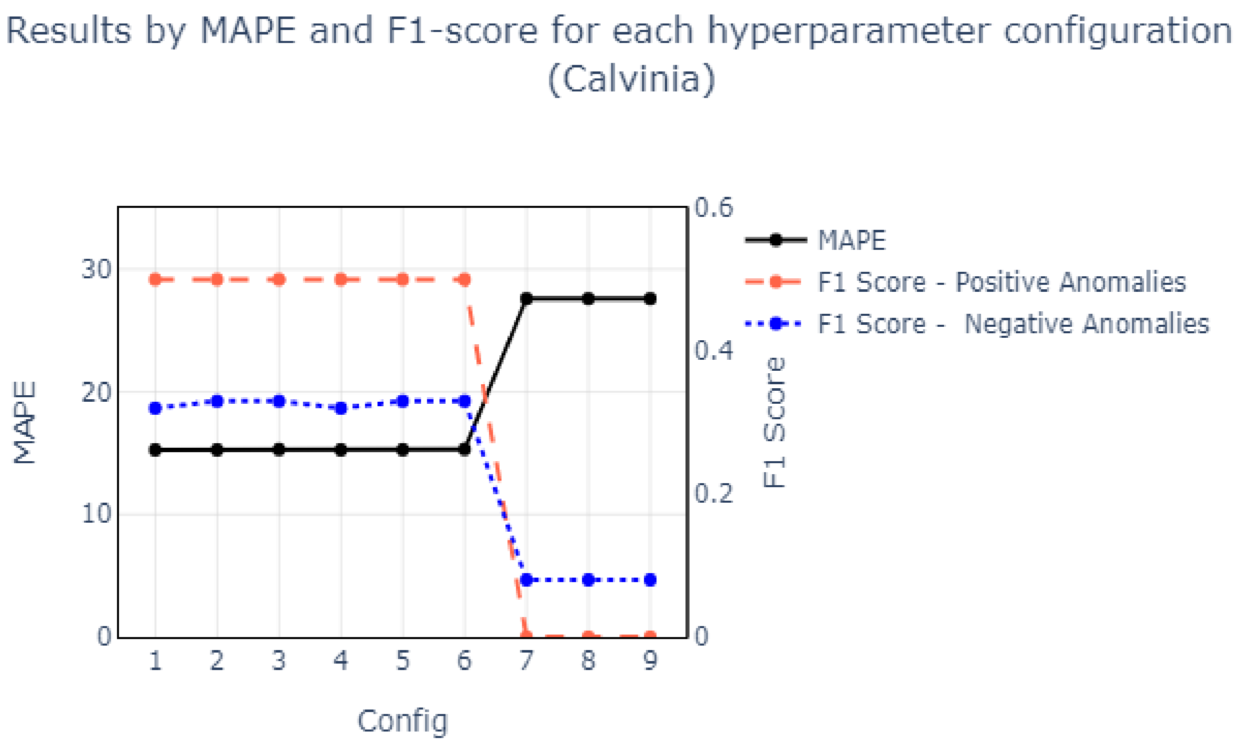

Figure 26 Results of the SVR walk-forward hyperparameter tuning for Calvinia

Table 29 Results of the SVR walk-forward hyperparameter tuning for Upington

Config	ε	C	MAPE	F1(+)	F1(-)
1	0.001	0.5	7.96	0.15	0.23
2	0.001	1	7.93	0.15	0.19
3	0.001	1.5	7.93	0.15	0.19
4	0.01	0.5	7.98	0.15	0.23
5	**0.01**	**1**	**7.95**	**0.17**	**0.23**
6	0.01	1.5	7.96	0.17	0.23
7	1	0.5	21.89	0.06	0.06
8	1	1	21.89	0.06	0.06
9	1	1.5	21.89	0.06	0.06

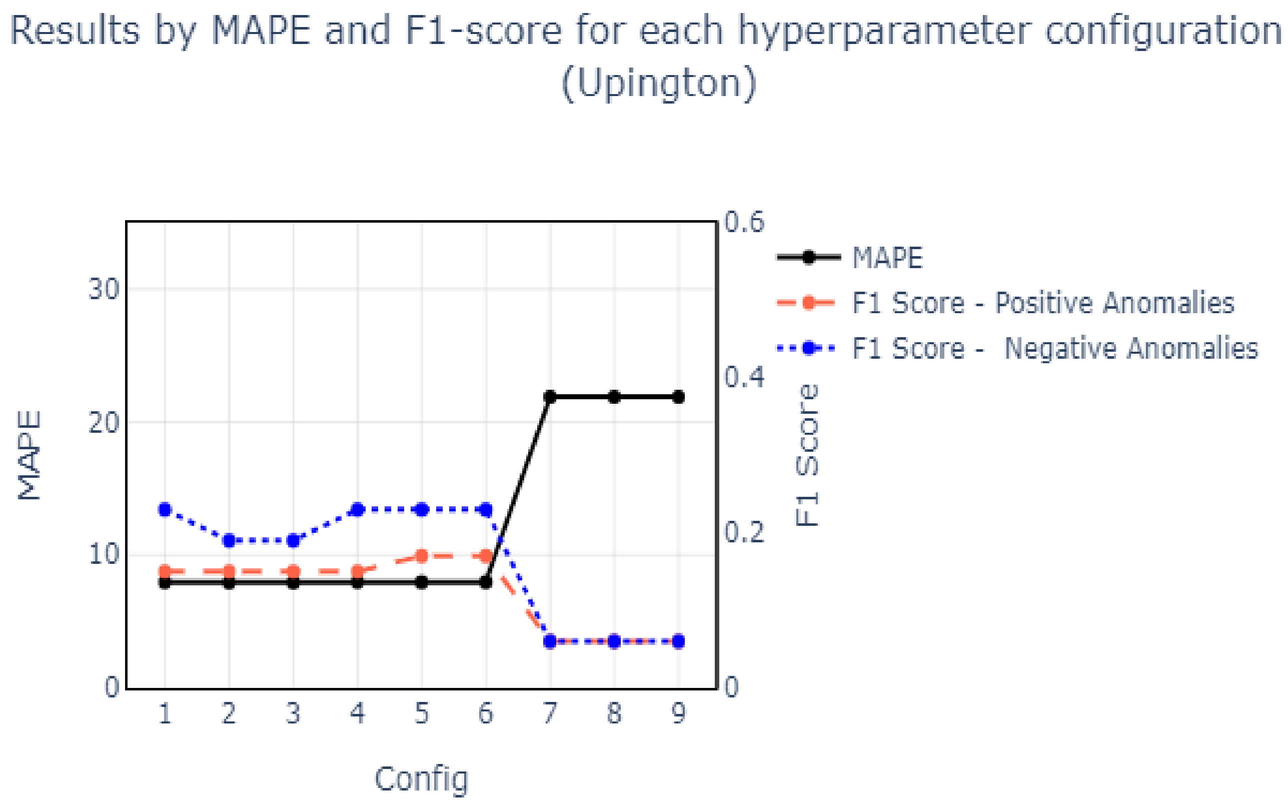

Figure 27 Results of the SVR walk-forward hyperparameter tuning for Upington

The results of the hyperparameter tuning performed for the ANN are shown in Table 30 to 33, with the best combinations highlighted in bold. The prediction and classification performance are shown in Figures 28 to 30. The optimal architecture consisted of an input layer, two hidden layers with four modes each using the Adam solver and the ReLU activation function.

Table 30 Results of the ANN walk-forward hyperparameter tuning for Johannesburg

Config	Layer	Size	Activation	MAPE	F1(+)	F1(-)
1	1	3	ReLU	16.04	0.11	0.14
2	2	3	ReLU	15.28	0.13	0.17
3	1	4	ReLU	15.99	0.1	0.15
4	**2**	**4**	**ReLU**	**13.67**	**0.14**	**0.15**
5	1	3	logistic	17.38	0.1	0.14
6	2	3	logistic	17.2	0.1	0.16
7	1	4	logistic	17.42	0.1	0.13
8	2	4	logistic	17.21	0.09	0.13

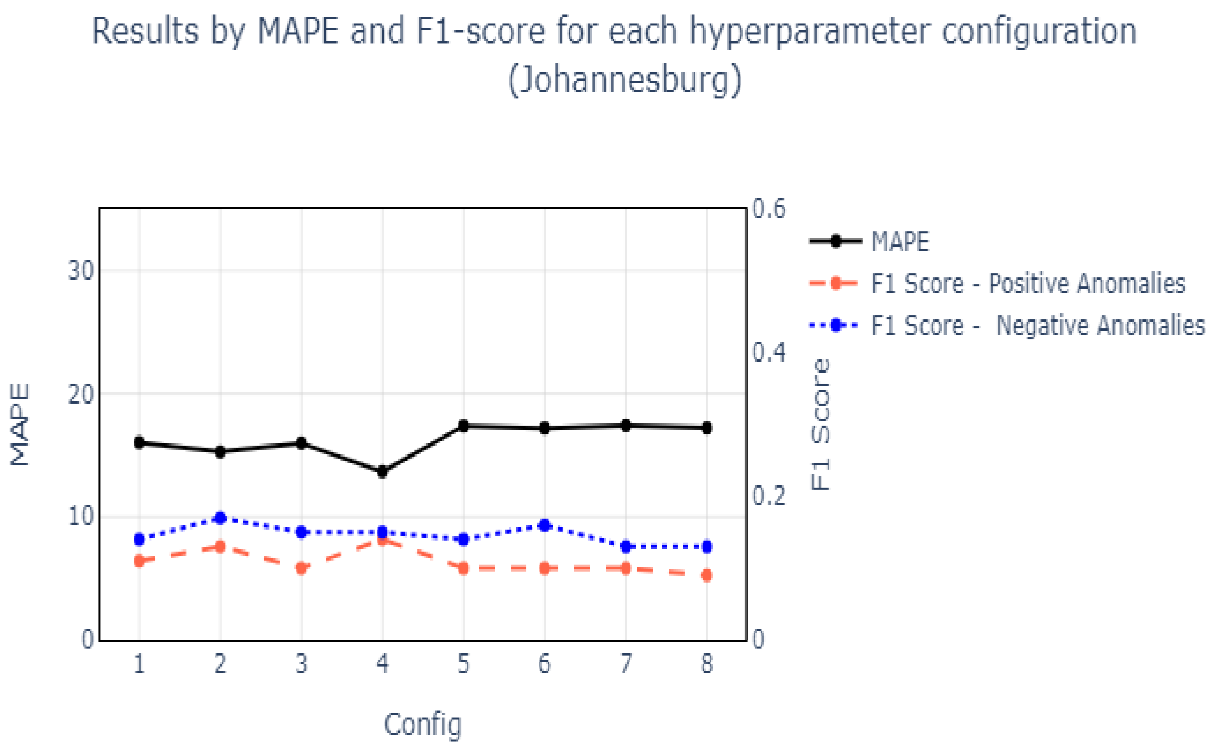

Figure 28 Results of the ANN walk-forward hyperparameter tuning for Johannesburg

Table 31 Results of the ANN walk-forward hyperparameter tuning for Calvinia

Config	Layer	Size	Activation	MAPE	F1(+)	F1(-)
1	1	3	ReLU	25.57	0.24	0.13
2	2	3	ReLU	23.83	0.15	0.17
3	1	4	ReLU	24.47	0	0.19
4	**2**	**4**	**ReLU**	**22.59**	**0.21**	**0.16**
5	1	3	logistic	27.93	0	0
6	2	3	logistic	27.74	0	0
7	1	4	logistic	27.99	0	0
8	2	4	logistic	27.67	0	0

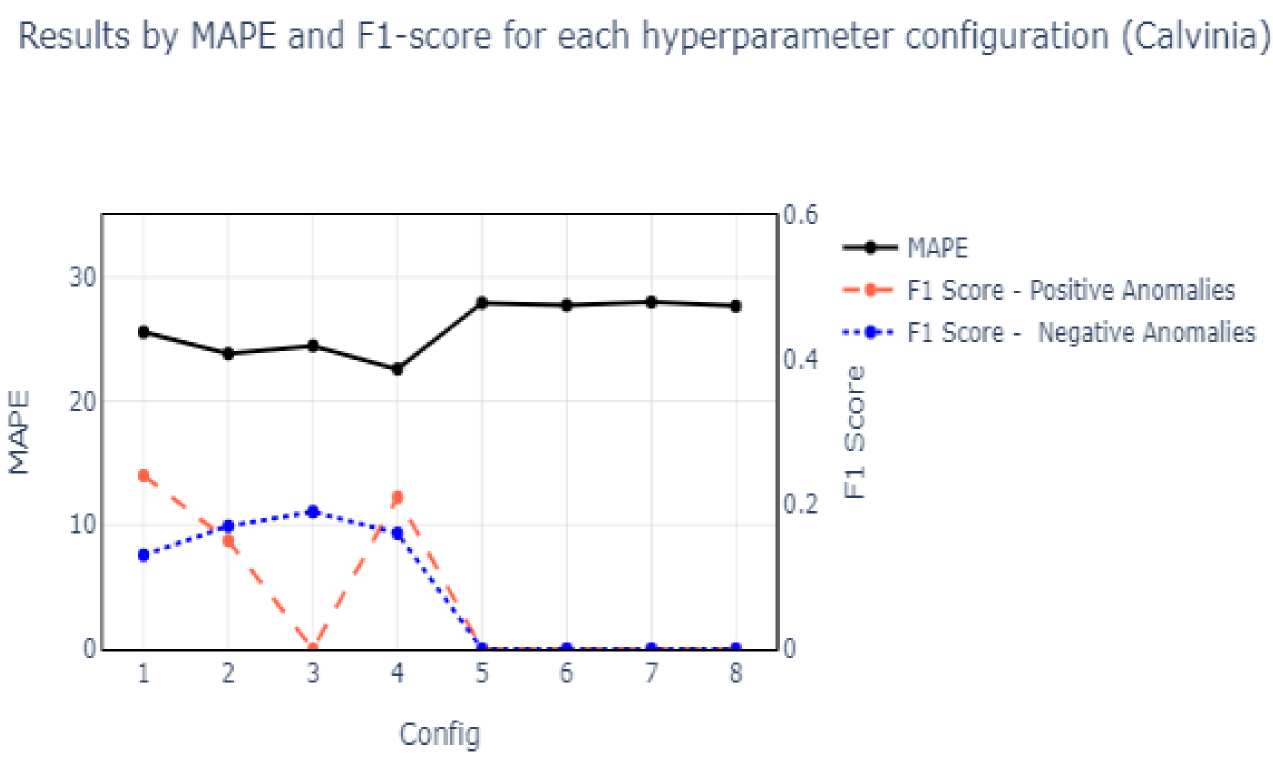

Figure 29 Results of the ANN walk-forward hyperparameter tuning for Calvinia

Table 32 Results of the ANN walk-forward hyperparameter tuning for Upington

Config	Layer	Size	Activation	MAPE	F1(+)	F1(-)
1	1	3	ReLU	17.71	0.05	0.1
2	2	3	ReLU	15.03	0.06	0.1
3	1	4	ReLU	16.42	0.06	0.11
4	**2**	**4**	**ReLU**	**14.75**	**0.06**	**0.11**
5	1	3	logistic	20.13	0.05	0.08
6	2	3	logistic	19.94	0.05	0.08
7	1	4	logistic	20.06	0.05	0.08
8	2	4	logistic	19.82	0.06	0.08

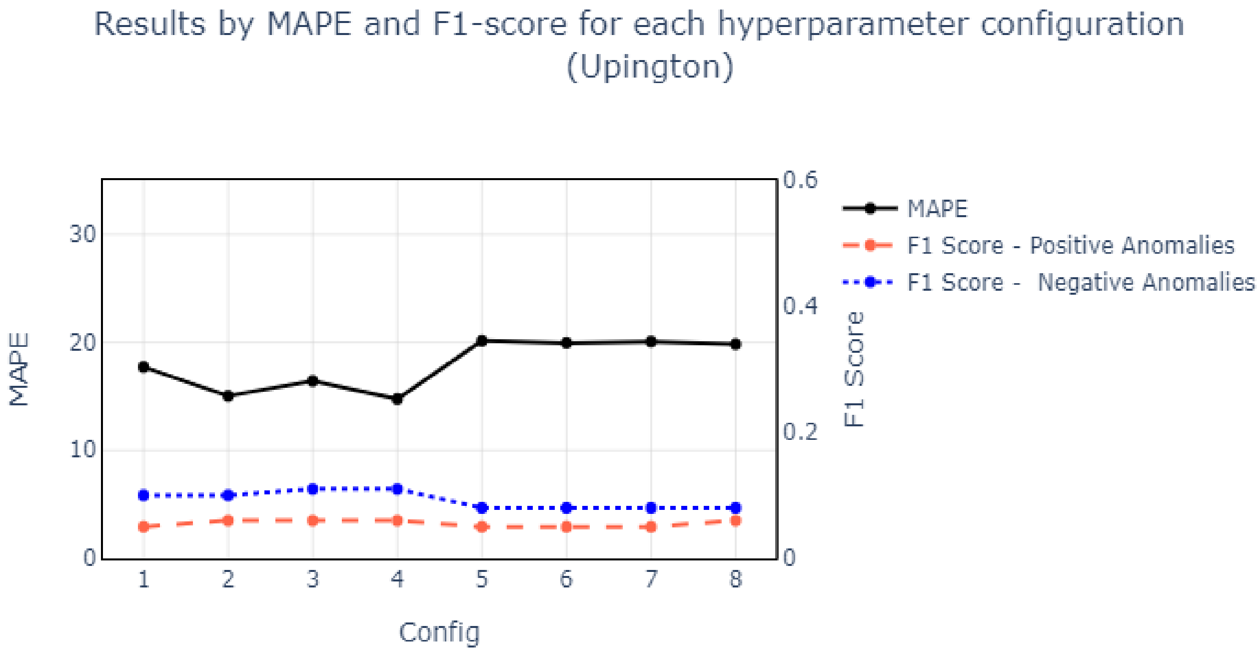

Figure 30 Results of the ANN walk-forward hyperparameter tuning for Upington

The results of hyperparameter tuning for the Huber regressor are shown in Tables 34 to 36, with the best combinations highlighted in bold. Figures 31 to 33 show the predictions and classification and performance for each hyperparameter combination. Larger values of Epsilon resulted in an increase in the number of positive anomalies observed at certain stations. The most optimal parameters for

Huber regression were Johannesburg: (epsilon:1.5, alpha :0.00005), Calvinia: (epsilon:2, alpha :0.00005), and Upington: (epsilon:1, alpha :0.00015).

Table 33 Results of the Huber Regressor walk-forward hyperparameter tuning for Johannesburg

Config	Epsilon	Alpha	MAPE	F1(+)	F1(-)
1	1	0.00005	8.75	0.34	0.28
2	1	0.0001	8.76	0.33	0.27
3	1	0.00015	8.76	0.33	0.27
4	**1.5**	**0.00005**	**8.74**	**0.34**	**0.28**
5	1.5	0.0001	8.74	0.34	0.28
6	1.5	0.00015	8.74	0.34	0.28
7	2	0.00005	8.74	0.31	0.27
8	2	0.0001	8.74	0.31	0.27
9	2	0.00015	8.74	0.31	0.27

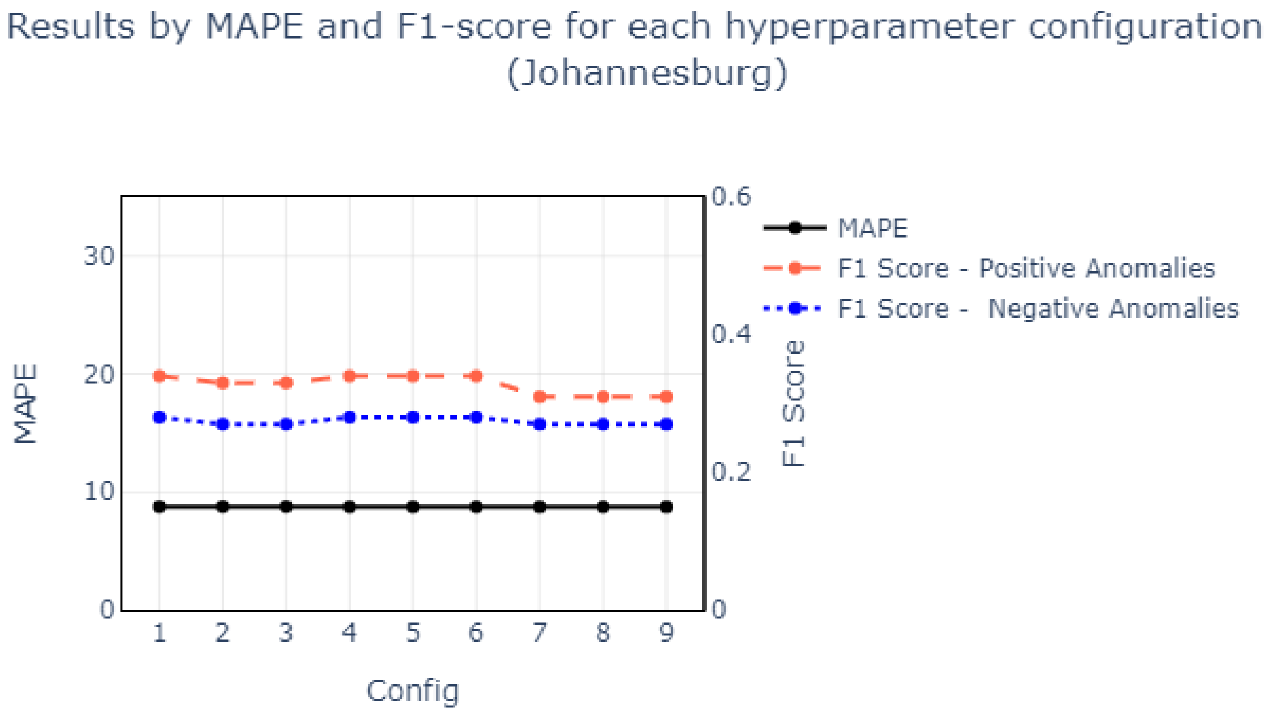

Figure 31 Results of the Huber Regressor walk-forward hyperparameter tuning for Johannesburg

Table 34 Results of the Huber Regressor walk-forward hyperparameter tuning for Calvinia

Config	Epsilon	Alpha	MAPE	F1(+)	F1(-)
1	1	0.00005	15.55	0.3	0.26
2	1	0.0001	15.55	0.3	0.26
3	1	0.00015	15.55	0.3	0.26
4	1.5	0.00005	15.41	0.4	0.26
5	1.5	0.0001	15.41	0.4	0.26
6	1.5	0.00015	15.41	0.4	0.26
7	**2**	**0.00005**	**15.34**	**0.4**	**0.26**
8	2	0.0001	15.34	0.4	0.26
9	2	0.00015	15.34	0.4	0.26

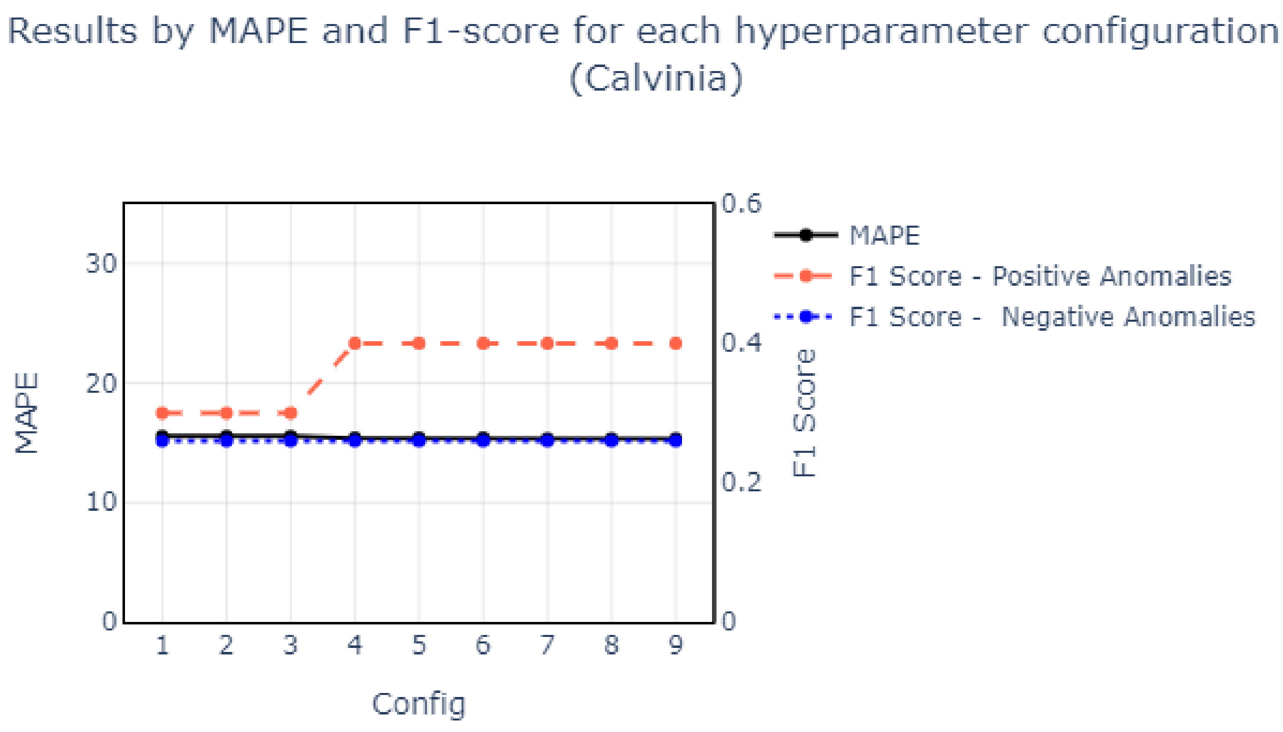

Figure 32 Results of the Huber Regressor walk-forward hyperparameter tuning for Calvinia

Table 35 Results of the Huber Regressor walk-forward hyperparameter tuning for Upington

Config	**Epsilon**	**Alpha**	**MAPE**	**F1(+)**	F1(-)
1	1	0.00005	8.65	0.27	0.21
2	1	0.0001	8.65	0.27	0.21
3	**1**	**0.00015**	**8.65**	**0.27**	**0.21**
4	1.5	0.00005	8.64	0	0.22
5	1.5	0.0001	8.64	0	0.22
6	1.5	0.00015	8.64	0	0.22
7	2	0.00005	8.67	0	0.23
8	2	0.0001	8.67	0	0.23
9	2	0.00015	8.67	0	0.23

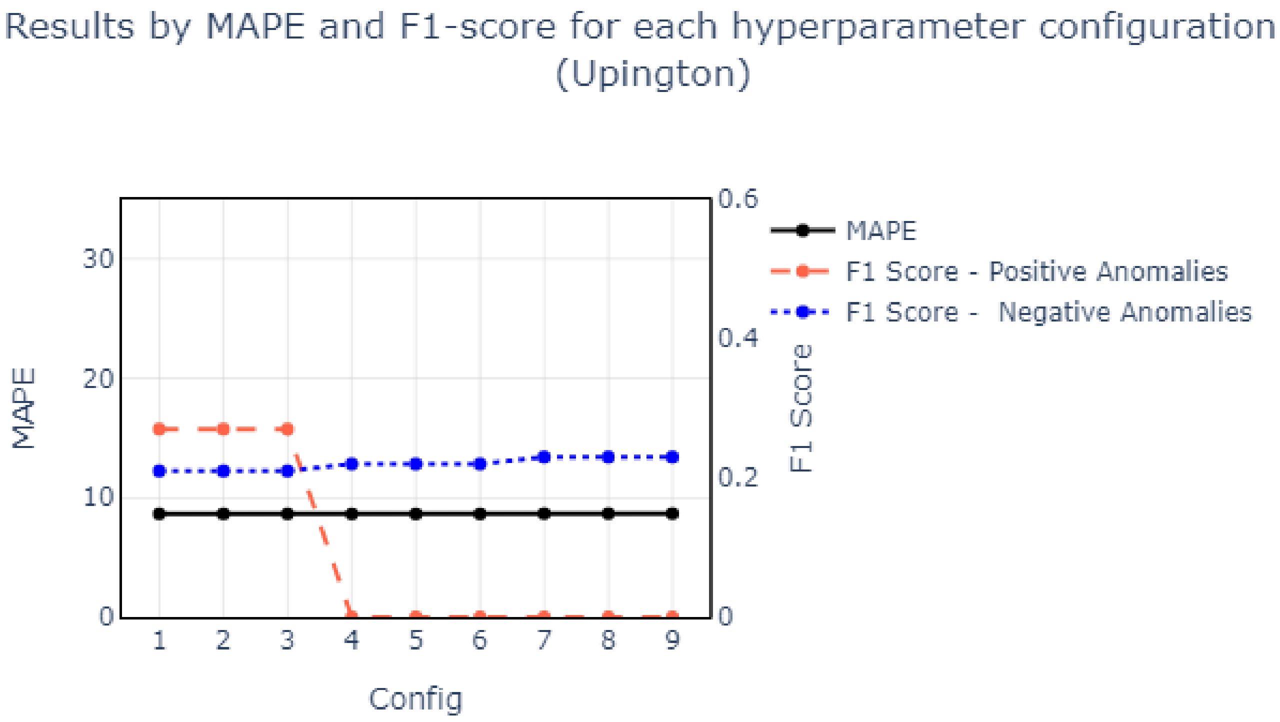

Figure 33 Results of the Huber Regressor walk-forward hyperparameter tuning for Upington

The prediction results of the most optimal configuration are shown in Table 36. The techniques were able to predict temperatures that are close to the actual values, as shown by MAPE. However, there are large outliers in the predictions, as shown by the MSE. The SVR performed better than the other techniques, followed by the Huber regressor. The ANN performed least satisfactorily. The results show that the climate at certain stations has an impact on how well the technique performs. This is evident in the variance of results between stations, with the Calvinia station having the highest prediction error compared to the other stations.

Table 36 Prediction results for walk-forward validation by technique and station based on the last four folds consisting of 365 days each between 2015 and 2018

		Next day prediction			
Technique	**Station**	**MAPE**	**MAE**	**MSE**	**RMSE**
SVR	**JHB**	8.36	1.77	5.88	2.42
	CAL	15.26	3.15	17.43	4.17
	UPT	7.95	2.13	9.56	2.86
Average		**10.52**	**2.35**	**10.96**	**3.15**
ANN	**JHB**	4.41±2.11	19.58±10.04	27.44±8.1	5.24±2.85
	CAL	6.29±3.44	30.97±16.85	54.96±19.43	7.41±4.41
	UPT	6.17±2.74	23.93±10.03	53.39±13.11	7.31±3.62
Max/min		**6.29±2.11**	**30.97±0.03**	**54.96±8.1**	**5.24±2.85**
Huber Regressor	**JHB**	8.74	1.86	6.20	2.49
	CAL	15.34	3.22	17.46	4.18
	UPT	8.65	2.39	10.33	3.12
Average		**10.91**	**2.49**	**11.33**	**3.26**

Table 37 shows the result of the anomaly classification and Table 38 shows the metric score by technique and station. The technique that performed best based on F1 score was the SVR, followed by Huber Regressor. The results show that the techniques had an overall low anomaly detection rate. A higher rate of negative anomalies is predicted than positive anomalies. A high rate of false positive predictions was also observed.

Table 37 Count of TP, TN, FN, and FP predicted using walk-forward validation for each anomaly type by technique and station

		Positive anomalies			Negative anomalies			TN
Technique	**Station**	**TP**	**FN**	**FP**	**TP**	**FN**	**FP**	
SVR	**JHB**	3	23	3	10	34	12	1376
	CAL	1	15	3	4	34	3	1401
	UPT	1	24	2	6	40	4	1384
Average		**1**	**21**	**3**	**6**	**37**	**6**	**1387**
ANN	**JHB**	6 ± 0	12 ± 0	86 ± 2	5 ± 0	16 ± 3	34 ± 10	1376±1222
	CAL	2 ± 0	7 ± 1	28 ± 1	3 ± 0	12 ± 4	18 ± 2	1405±1350
	UPT	3 ± 0	10 ± 1	73 ± 1	4 ± 0	16 ± 4	76 ± 3	1389±1121
Max/min		**6 ± 0**	**12 ± 0**	**86 ± 1**	**5 ± 0**	**38 ± 3**	**76 ± 2**	**1390±1231**
Huber regressor	**JHB**	6	20	5	11	33	12	1374
	CAL	2	14	3	5	33	6	1395
	UPT	2	23	6	6	40	7	1377
Average		**3**	**19**	**5**	**7**	**35**	**8**	**1382**

Table 38 Classification results for walk-forward validation by technique and station based on the average results of the last four folds consisting of 365 days between 2015 and 2018

		Positive anomalies				Negative anomalies			
Technique	**Station**	**A**	**P**	**R**	**F1**	**A**	**P**	**R**	**F1**
SVR	**JHB**	0.98	0.15	0.31	0.38	0.95	0.39	0.21	0.27
	CAL	0.99	0.13	0.13	0.50	0.97	0.27	0.13	0.33
	UPT	0.99	0.13	0.03	0.17	0.96	0.44	0.11	0.23
Average		**0.99**	**0.14**	**0.16**	**0.35**	**0.96**	**0.37**	**0.15**	**0.28**
ANN	**JHB**	0.99±0.85	0.25±0	1±0	0.40±0.04	0.97±0.84	0.5±0	0.63±0	0.40±0.05
	CAL	0.99±0.91	0.25±0	0.5±0	0.33±0.11	0.98±0.91	1±0	0.43±0	0.27±0.06
	UPT	0.99±0.79	1±0.00	0.75±0	0.18±0.03	0.98±0.78	1±0	0.43±0	0.29±0.07
Max/min		**0.99**±0.79	**1±0.00**	**0.75±0**	**0.40±0.04**	**0.98±0.78**	**1±0**	**0.63±0**	**0.40±0.05**
Huber Regressor	**JHB**	0.98	0.26	0.38	0.34	0.95	0.36	0.23	0.28
	CAL	0.99	0.08	0.13	0.40	0.97	0.29	0.15	0.26
	UPT	0.98	0.10	0.05	0.27	0.96	0.27	0.11	0.21
Average		**0.98**	**0.15**	**0.19**	**0.34**	**0.96**	**0.31**	**0.16**	**0.25**

The network found to be the most optimal, reported above, shows an improvement in error over time, particularly from the 14th training iteration onwards. In addition, the variance is more stable across the 10 training runs compared to the other two best performing neural networks. The results of the predictive error of the three best performing neural networks are shown in Appendix C.

4.1.3 Comparison of techniques

The results of the comparison of the prediction results are shown in Table 39. The values shown below represent the average per technique. The results show that an improvement in prediction accuracy is observed under walk-forward validation across all metrices used.

Table 39 Average performance comparison across all stations by technique for next day predictions.

Technique	Metric	Hold-out	Walk-forward	Difference	Movement
SVR	**MAPE**	11.06	10.52	0.54	↓
	MAE	2.49	2.35	0.14	↓
	MSE	11.75	10.96	0.79	↓
	RMSE	3.35	3.15	0.20	↓
ANN	**MAPE**	17.53	17.00	0.53	↓
	MAE	4.08	4.00	0.08	↓
	MSE	26.43	25.8	0.63	↓
	RMSE	5.00	4.87	0.13	↓
Huber Regressor	**MAPE**	11.35	10.91	0.44	↓
	MAE	2.55	2.49	0.06	↓
	MSE	12.16	11.33	0.83	↓
	RMSE	3.41	3.26	0.15	↓

The classification results are shown in Table 40. Through the comparison, it was found that the total percentage of detected anomalies increases with walk-forward validation. However, this increase is accompanied by an increase in the rate of false positives.

Table 40 Average performance comparison across all stations by technique for anomaly predictions

Technique	Metric	Anomaly movement	Hold-out	Walk-forward	Difference	Movement
SVR	**Precision**	Positive	0.16	0.14	0.02	↓
		Negative	0.47	0.37	0.1	↓
	Recall	Positive	0.07	0.16	-0.09	↑
		Negative	0.17	0.15	0.02	↓
	F1 Score	Positive	0.1	0.35	-0.25	↑
		Negative	0.24	0.28	-0.04	↑
ANN	**Precision**	Positive	0.04	0.04	0.00	-
		Negative	0.10	0.11	-0.01	↑
	Recall	Positive	0.10	0.17	-0.07	↑
		Negative	0.10	0.09	0.01	↓
	F1 Score	Positive	0.07	0.14	-0.07	↑
		Negative	0.09	0.14	-0.05	↑
Huber Regressor	**Precision**	Positive	0.17	0.15	0.02	↓
		Negative	0.33	0.31	0.02	↓
	Recall	Positive	0.15	0.19	-0.04	↑
		Negative	0.18	0.16	0.02	↓
	F1 Score	Positive	0.16	0.34	-0.18	↑
		Negative	0.23	0.25	-0.02	↑

The results were similar in both predicting the maximum temperatures and detecting anomalies between the validation techniques studied. However, the walk-forward technique provided better performance, as a reduction in the next-day prediction error and an increase in the anomaly detection rate were observed for both positive and negative anomalies. Aside from performance, the walk-forward technique offers additional advantages over hold-out validation. This includes a larger test set that allows for an evaluation over a longer period.

4.2 Experiment 2: Spatial Temporal Analysis

The goal of this experiment was to test whether accounting for possible spatial influences would improve the model's ability to detect anomalies. To accomplish this, data from nearby stations were included as model input. This experiment was conducted for the selected techniques and stations. The optimal hyperparameters and validation techniques determined in Experiment 1 were used in this experiment. Thus, the techniques were trained on 19 years of data between 1999 and 2017 and validated on 19 years of data between 2000 and 2018. The results reported below are based only on the last four iterations between 2015 and 2018.

The prediction results are shown in Table 41. The techniques do not perform well in predicting maximum temperatures as larger MAPE's are observed. In addition, the predictions have large deviations from the actual value, as shown by the MSE. In this experiment, the SVR performed better than the ANN and the Huber regressor.

Table 41 Prediction result of the spatial analysis by technique and station

		Next day predictions			
Technique	**Station**	**MAPE**	**MAE**	**MSE**	**RMSE**
SVR	**JHB**	28.51	5.95	49.47	7.01
	CAL	18.83	4.02	25.17	5.02
	UPT	27.72	5.75	47.42	6.85
Average		**25.02**	**5.24**	**40.69**	**6.29**
ANN	**JHB**	31.8±16.01	6.39±3.4	55.54±16.34	7.45±4.04
	CAL	31.06±9.51	6.25±2.13	54.53±7.01	7.38±2.65
	UPT	9.84±5.03	39.4±17.77	124.34±35.24	11.15±5.94
Max/min		**31.8±5.03**	**39.4±2.13**	**124.34±7.01**	**11.15±2.65**
Huber Regressor	**JHB**	28.15	5.88	48.40	6.93
	CAL	17.19	3.70	22.15	4.70
	UPT	33.01	8.78	106.94	10.27
Average		**26.12**	**6.12**	**59.16**	**7.30**

Table 42 shows the anomaly classification details and Table 43 shows the classification results. The techniques had a low anomaly detection rate and a high false positive rate. The ANN predicted the largest number of positive and negative anomalies. However, looking at the prediction performance, this could indicate that the technique predicted temperatures too high or too low, resulting in a greater number of anomalies being predicted by chance.

Table 42 Count of TP, TN, FN, and FP predicted in Experiment 2 by technique and station

		Positive anomalies			Negative anomalies			TN
Technique	**Station**	**TP**	**FN**	**FP**	**TP**	**FN**	**FP**	
SVR	**JHB**	6	13	300	13	23	44	1062
	CAL	2	14	2	1	37	6	1399
	UPT	5	14	302	15	22	21	1082
Average		**4**	**14**	**201**	**10**	**27**	**24**	**1181**
ANN	**JHB**	4 ± 0	11 ± 0	232 ± 121	7 ± 0	15 ± 0	38 ± 9	1169±937
	CAL	2 ± 0	7 ± 1	10 ± 1	3 ± 0	12 ± 4	19 ± 1	1403±1370
	UPT	3 ± 1	9 ± 0	298 ± 160	13 ± 0	12 ± 0	323 ± 169	1031±595
Average		**4 ± 0**	**11 ± 0**	298 **± 1**	**13 ± 0**	**15 ± 0**	323 **±1**	1201±967
Huber regressor	**JHB**	6	13	283	13	23	40	1083
	CAL	2	14	1	0	38	5	1401
	UPT	8	13	360	27	15	310	728
Average		**5**	**13**	**215**	**13**	**25**	**118**	**1071**

Table 43 Result of the spatial analysis of anomaly predictions, per technique and station

		Positive anomalies				Negative anomalies			
Technique	**Station**	**A**	**P**	**R**	**F1**	**A**	**P**	**R**	**F1**
SVR	**JHB**	0.72	0.02	0.56	0.04	0.71	0.03	0.39	0.08
	CAL	0.98	0.31	0.10	0.23	0.97	0.11	0.03	0.17
	UPT	0.73	0.01	0.54	0.04	0.73	0.04	0.55	0.08
Average		**0.81**	**0.11**	**0.40**	**0.10**	**0.80**	**0.06**	**0.33**	**0.11**
ANN	**JHB**	0.95±0.68	0.05±0.00	1±0.00	0.95±0.68	0.95±0.68	0.32±0.00	0.64±0.06	0.4±0.03
	CAL	1±0.94	1±0.00	0.5±0.00	0.67±0.15	0.98±0.92	1±0.00	0.43±0.00	0.46±0.14
	UPT	0.82±0.34	0.09±0.01	0.05±0.01	1±0.1	0.81±0.36	0.13±0.02	0.08±0.00	1±0.00
Average		0.95±0.34	**1**±0.00	1±0.00	1±0.1	0.98±0.36	1±0.00	0.64±0.00	0.46±0.00
Huber Regressor	**JHB**	0.74	0.02	0.56	0.05	0.73	0.04	0.39	0.08
	CAL	0.99	0.40	0.16	0.27	0.97	0.00	0.00	0.00
	UPT	0.46	0.01	0.61	0.02	0.47	0.04	0.64	0.07
Average		**0.73**	**0.14**	**0.44**	**0.11**	**0.72**	**0.02**	**0.35**	**0.08**

A comparison of the results from Experiment 1 (walk-forward validation) and Experiment 2 showed that the inclusion of data from nearby stations did not improve the performance of the techniques. It was found that the inclusion of data from nearby stations decreased the prediction and classification accuracy.

4.3 Experiment 3: Comparison Between Regression and Classification Techniques

This experiment involved comparing the performance of classification and regression techniques for predicting anomalies in weather data. The walk-forward approach was used because it was shown to perform satisfactory in Experiment 1. As such, models were trained on 19 years of data between 1999 and 2017 and tested on 19 years of data between 2000 and 2018. The results presented below are based only on the last four iterations between 2015 and 2018. As noted in Experiments 1 and 2, the SVR provided the best performance in both next-day predictions and anomaly detection. The results reported on in Experiment 1 (Section 4.1.2) were therefore used for this evaluation.

The hyperparameter tuning of the SVC was performed using the process described in Section 3.3.4. Six different configurations were evaluated based on the ranges of hyperparameters discussed in Section 3.3.3. The results of the hyperparameter tuning for the selected stations are shown in Tables 44 to 46, with the best combinations highlighted in bold. Figure 34 shows the classification performance for each hyperparameters configuration for Johannesburg. Details for other stations are not included because results were not available for analysis. Only two parameter values resulted in anomalous predictions for the Johannesburg station. It was found that higher values of C resulted in better performance. The best value for the hyperparameter C was 15.

Table 44 show the results of the SVC hyperparameter tuning for Johannesburg. Where (P) is the Precision and (R) is the Recall.

		Positive anomalies			Negative anomalies		
Config	**C**	**P**	**R**	**F1**	**P**	**R**	**F1**
1	0.5	0	0	0	0	0	0
2	1	0	0	0	0	0	0
3	1.5	0	0	0	0	0	0
4	10	0	0		0.5	0.02	0.17
5	**15**	**0.17**	**0.03**	**0.18**	**0.33**	**0.05**	**0.17**
6	20	0.13	0.03	0.18	0.29	0.07	0.14

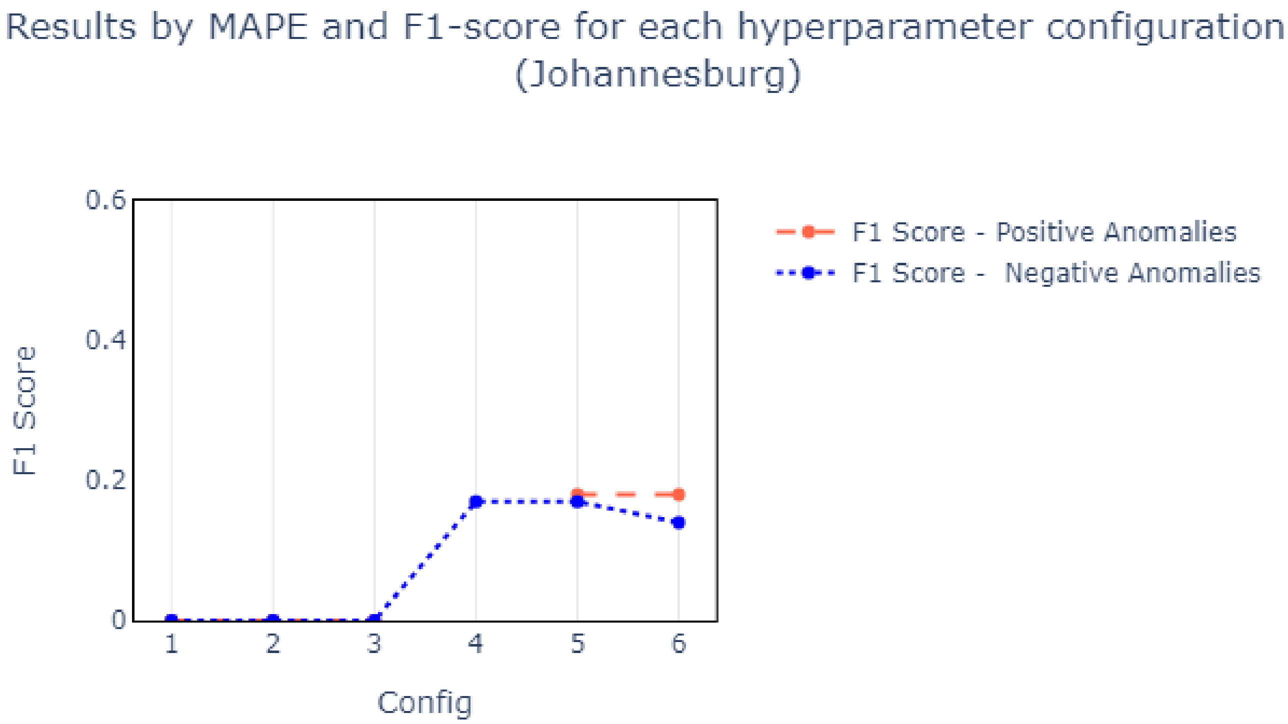

Figure 34 Show the results of the SVC hyperparameter tuning for Johannesburg.

Table 45 show the results of the SVC hyperparameter tuning for Calvinia.

		Positive anomalies			Negative anomalies		
Config	**C**	**P**	**R**	**F1**	**P**	**R**	**F1**
1	0.5	0	0	0	0	0	0
2	1	0	0	0	0	0	0
3	1.5	0	0	0	0	0	0
4	10	0	0	0	0	0	0
5	**15**	**0**	**0**	**0**	**0**	**0**	**0**
6	20	0	0	0	0	0	0

Table 46 show the results of the SVC hyperparameter tuning for Upington.

		Positive anomalies			Negative anomalies		
Config	**C**	**P**	**R**	**F1**	**P**	**R**	**F1**
1	0.5	0	0	0	0	0	0
2	1	0	0	0	0	0	0
3	1.5	0	0	0	0	0	0
4	10	0	0	0	0	0	0
5	**15**	**0**	**0**	**0**	**0**	**0**	**0**
6	20	0	0	0	0	0	0

The results of the best configuration are reported in Table 47. The only station at which the technique was able to predict anomalies were Johannesburg, with an F1 score of 0.18 for positive anomalies and 0.17 for negative anomalies. At the remaining stations, the technique predicted all observations as normal.

Table 47 Results of the anomaly classification of the SVC

	Positive anomalies				Negative anomalies			
Station	**Accuracy**	**Precision**	**Recall**	**F1-score**	**Accuracy**	**Precision**	**Recall**	**F1-score**
JHB	0.98	0.17	0.03	0.18	0.97	0.33	0.05	0.17
CAL	0.99	0	0	0	0.97	0	0	0
UPT	0.99	0	0	0	0.97	0	0	0
	0.99	**0.06**	**0.01**	**0.06**	**0.97**	**0.11**	**0.02**	**0.06**

Table 48 and Figure 35 shows the F1 score for positive and negative anomalies by the SVR and SVC. The blue bar shows the positive F1 score and the and red bar the negative F1 score for the SVC. The green bar shows the positive F1 score and the and purple the negative F1 score for the SVR. The SVR outperformed the SVC for both anomaly types and at all locations. At Calvinia and Upington, the classifier failed to predict anomalies, so no data are shown for the F1 score. A comparison of the number of anomalies detected by SVC and SVR is shown in Table 46 and a comparison of the scores obtained is shown in Table 49.

Table 48 Comparison of the number of TP, FN and TP predicted by the SVC and SVR.

Technique Type	Station	TP	FN	FP
Classification	**JHB**	3	67	4
	CAL	0	54	0
	UPT	0	71	0
Average		**1**	**64**	**1**
Regression	**JHB**	13	57	15
	CAL	1	53	6
	UPT	7	64	6
Average		**7**	**58**	**9**

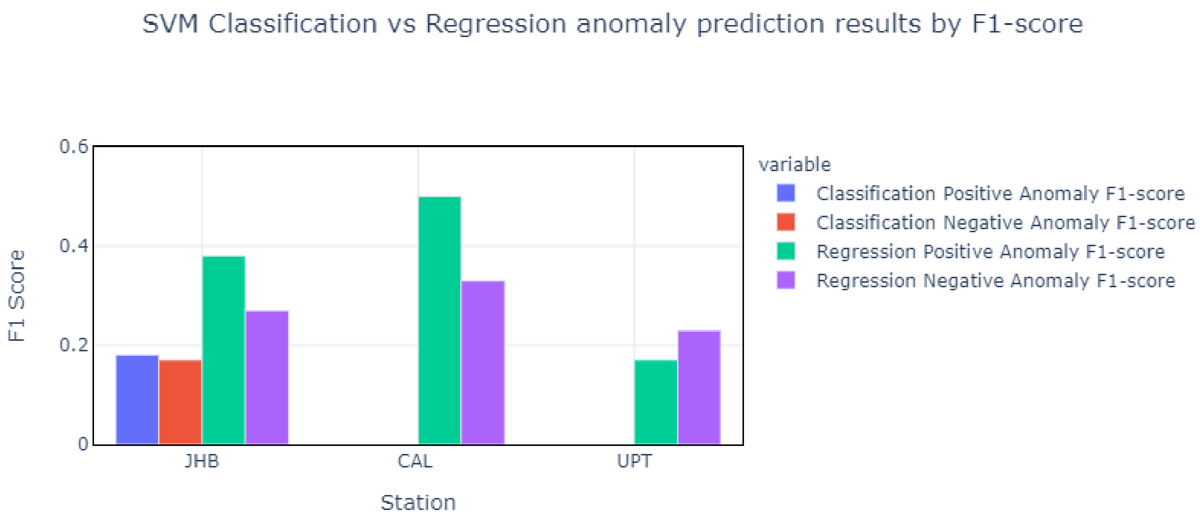

Figure 35 Comparison of the number of TP, FN and TP predicted by the SVC and SVR.

Table 49 Comparison of the anomaly prediction results of the SVM classifier and regressor

	Positive anomalies		Negative anomalies	
Station	**F1-score (Regression)**	**F1-score (Classification)**	**F1-score (Regression)**	**F1-score (Classification)**
JHB	0.38	0.18	0.27	0.17
CAL	0.50	0	0.33	0
UPT	0.17	0	0.23	0
Average	**0.35**	**0.06**	**0.28**	**0.06**

4.4 Conclusion

The goal of the experiments conducted was to find the most optimal solution for weather prediction using machine learning. This included a comparative analysis of the different techniques, validation techniques and spatial analysis. Experiment 1 compared the results of the techniques evaluated using the hold-out and walk-forward validation techniques. The results showed that the techniques evaluated with the walk-forward method had an overall lower error in predicting the maximum temperatures for the next day and had a higher probability of correctly detecting anomalies. In addition, the walk-forward method offered the advantage of assessing performance over a longer period, allowing for an assessment of performance under different conditions (i.e., warmer years versus cooler years). The results of Experiment 2 showed that combining data from nearby stations did not improve the ability of the techniques to detect and predict anomalies and instead, resulted in a decrease in overall accuracy and prediction of anomalies. The classification vs. regression analysis performed in Experiment 3 showed that the regression-based SVM was better able to detect anomalies based on observations from the previous days than the classification-based SVM.

The analysis of the hyperparameter tuning found that for SVR, a value of less than 1 for ε led to better performance in both predictions and classification, while C had an insignificant effect on performance. The optimal architecture and configuration for the ANN consisted of an input layer, two hidden layers with four nodes each, and an output layer trained with the Adam solver and the ReLU activation function. For the Huber regressor, lower values of epsilon performed best.

Chapter 5

5. Analysis and Discussion

The analysis is divided into four sections and organised as follows: (1) an analysis of how maximum temperatures was predicted and classified based on the results reported in Experiment 1, (2) discussion of the spatial analysis, (3) analysis of the performance of the performance of classification and regression techniques, and (4) how the results compare to related literature.

5.1 Analysis of Predictions

During the analysis of the results, it was found that the techniques performed well in predicting maximum temperatures. However, low anomaly detection rates and high false positive rates was observed. The goal of this analysis was therefore to understand how next day temperatures was predicted and how anomalies were classified. The analysis further explored if false positive predictions indicated that an anomaly would occur in the days following the prediction. This analysis was performed based on the results of the best performing models identified in Chapter 4.

5.1.1. Temperature Prediction

Figures 36 to 38 show the predictions and actual values between 2015 and 2018 for the SVR, ANN and Huber Regressor, respectively. The blue solid line represents the actual temperatures, and the dotted red line the predicted temperatures. The SVR and Huber regressor performed well in predicting maximum temperatures one day ahead. The SVR, however, could only predict values within a specific range of values. This resulted in extreme temperatures outside of this range not being predicted. The Huber regressor performed well in predicting next-day maximum temperatures that were close to actual temperatures, especially at more extreme temperatures. The ANN performed the least satisfactorily in predicting next day temperatures.

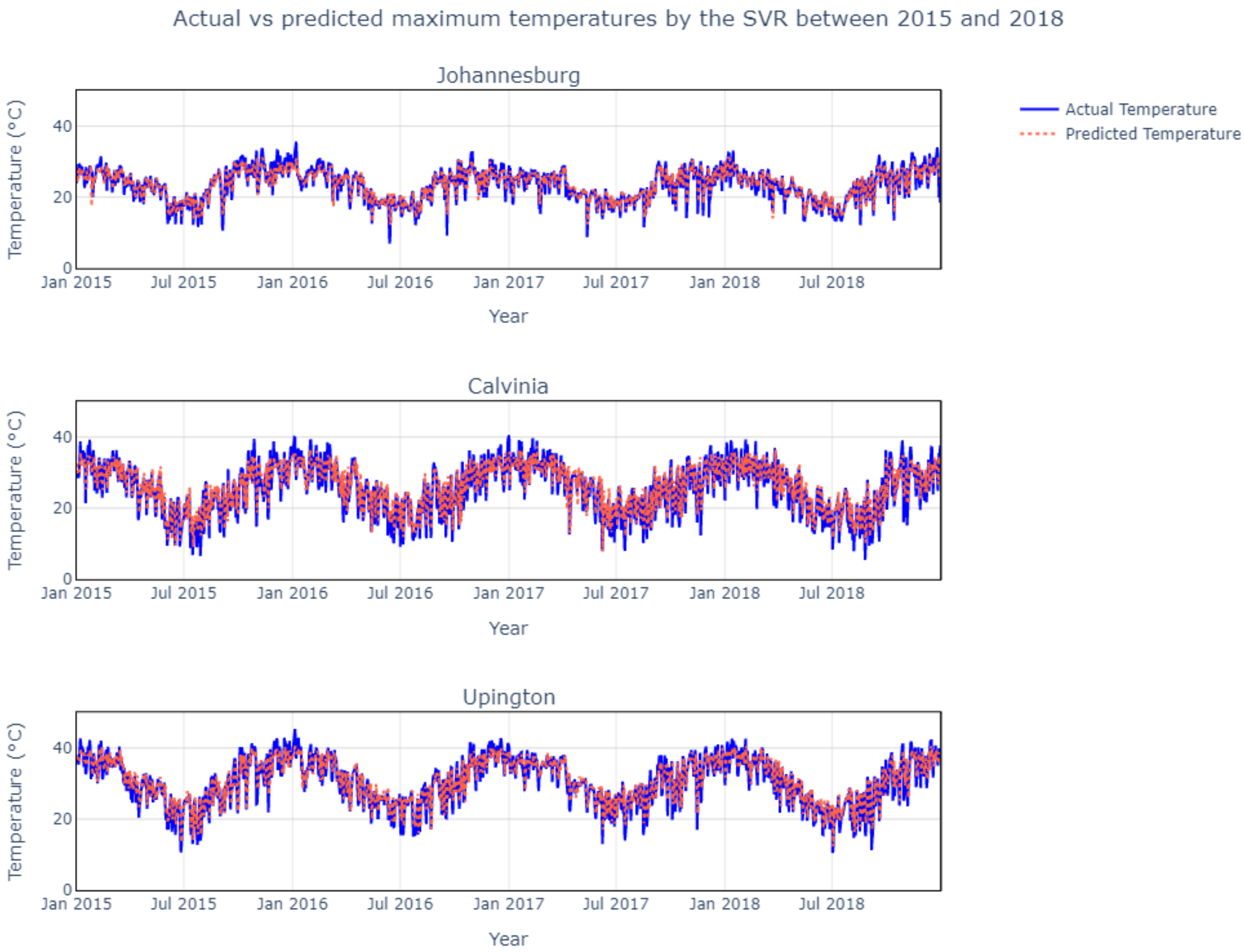

Figure 36 Actual and predicted temperatures predicted by the SVR

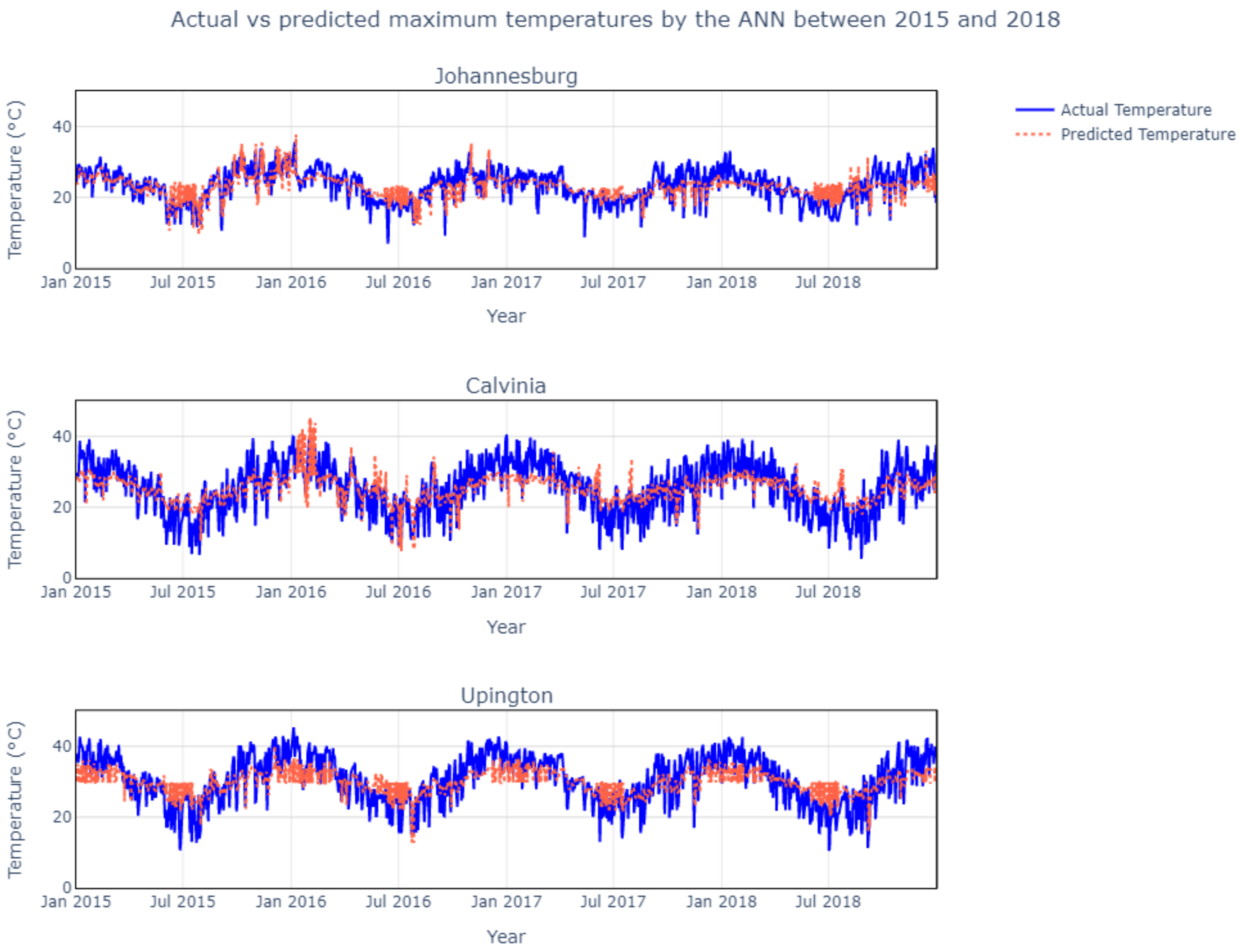

Figure 37 Actual and predicted temperatures predicted by the ANN

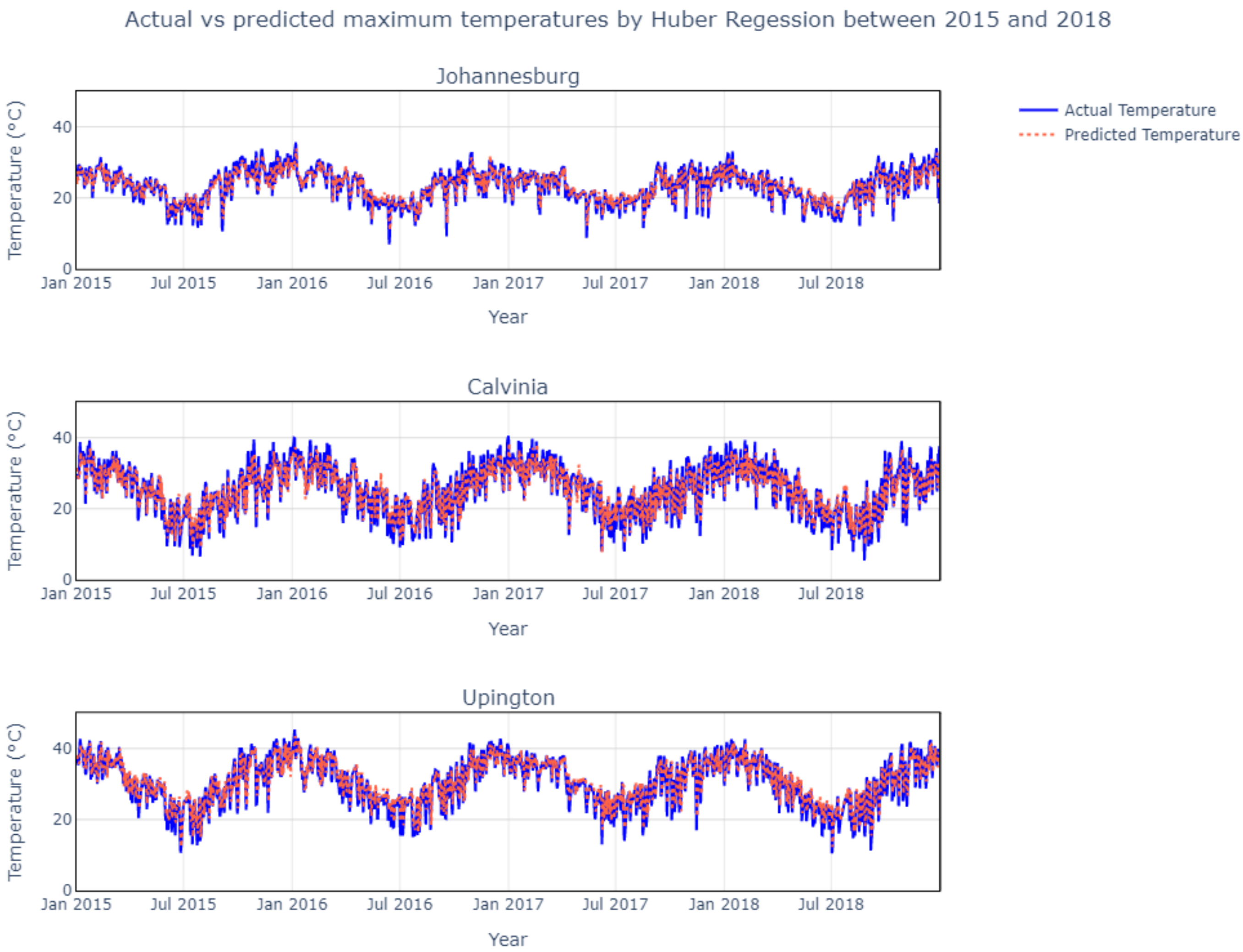

Figure 38 Actual and predicted temperatures predicted by the Huber Regressor

Considering the temperature prediction trends described above, the Huber Regressor provided a more reliable estimation of maximum temperatures. The performance showed that the Huber Regressor was able to adapt to different climates and performed well in predicting warmer or colder than average conditions.

5.1.2. Anomaly Classification

The classification results for the SVR are shown in Figure 39 for the selected stations between 2015 and 2018. The red and blue lines show the positive and negative anomaly boundaries, respectively. The markers in green shows where the technique successfully detected anomalies. The red markers show where the technique predicted that an anomaly would occur, but the actual value was within the boundaries. The orange markers show where anomalies were not successfully detected by the technique.

The high prevalence of negative anomalies in the datasets resulted in the SVR detecting more true anomalies when temperatures were colder. To assess whether false alarms resulted in true anomalies, a lag function was applied to examine what happened three days before and after the prediction. The analysis showed that most false anomalies were predicted due to anomalies on the previous day. In general, the false anomalies did not become actual positives.

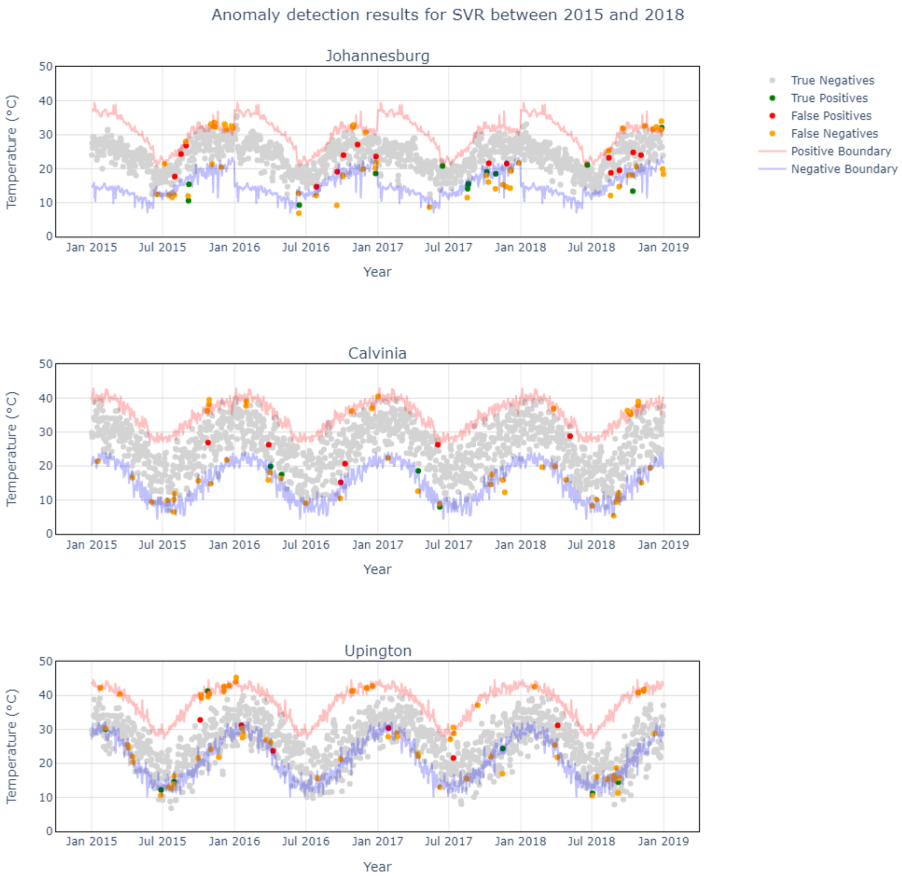

Figure 39 Anomaly classification results by the SVR

The classification results for the ANN are shown in Figure 40 for the selected stations between 2015 and 2018. The technique had a high false alarm rate, especially in winter and summer. The analysis of the false alarms found that most false alarms were detected when anomalies occurred day(s) before.

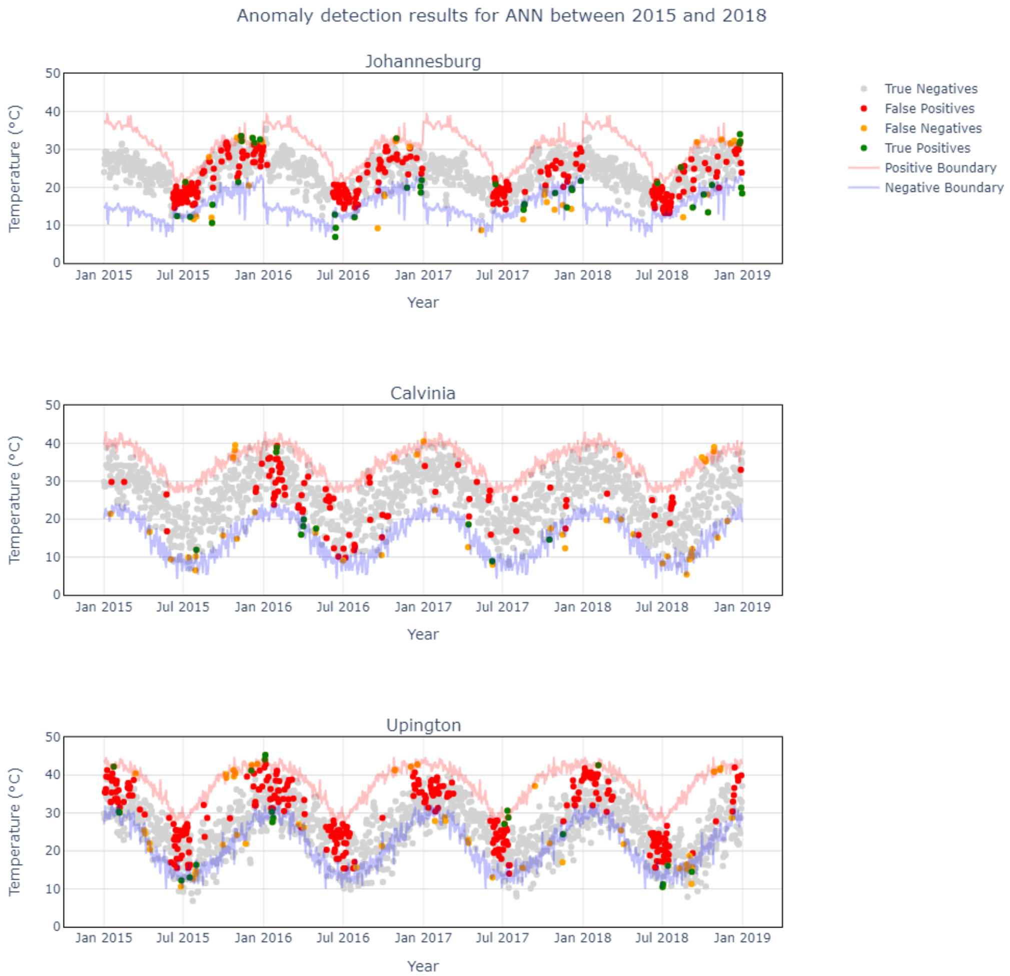

Figure 40 Anomaly classification results by the ANN

Classification results for the Huber regressor are shown in Figure 41 for the selected stations between 2015 and 2018. The technique had a higher probability of successfully detecting anomalies when temperatures were below the climate baseline. This resulted in more anomalies being predicted during winter. In analysing the false positive predictions, it was found that the Huber regressor predicted false anomalies when the previous day(s) were anomalous and did not indicate that an anomaly would occur in future.

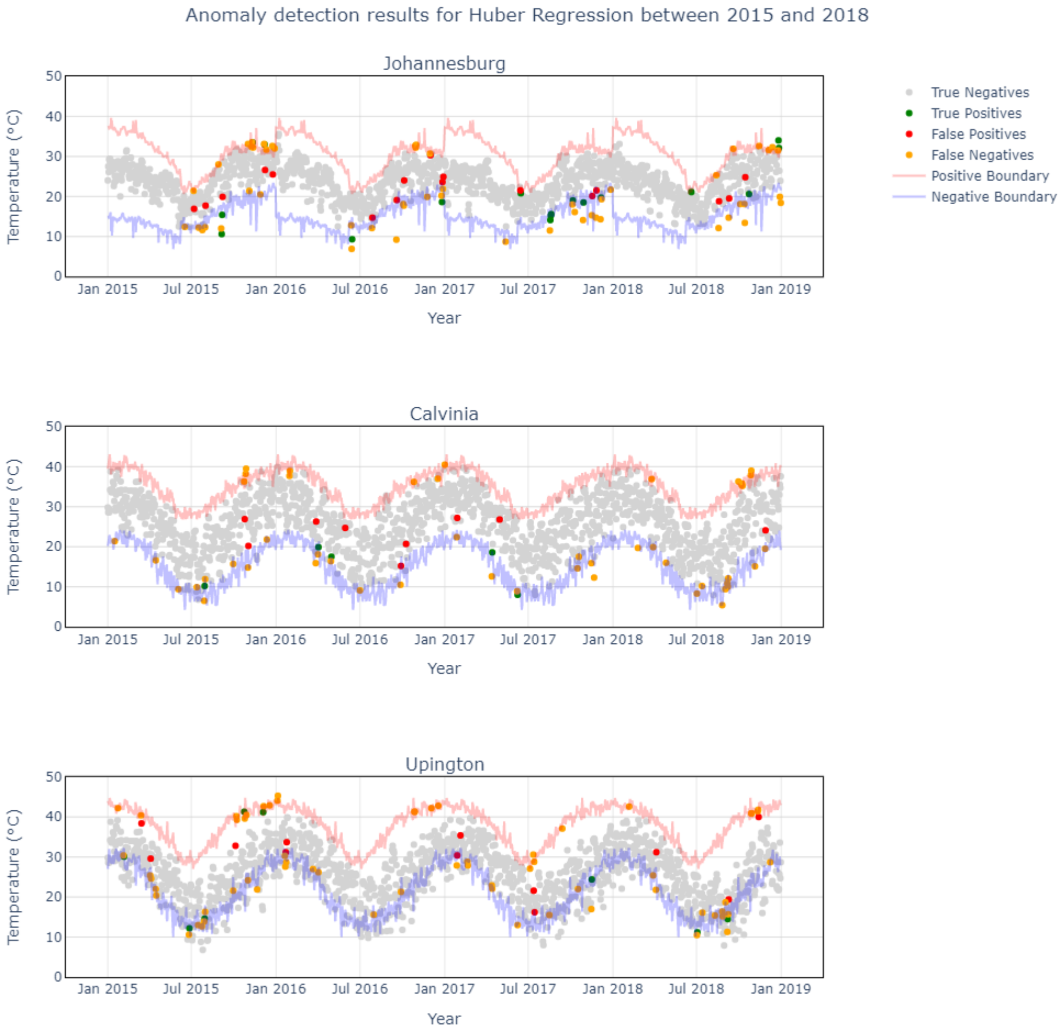

Figure 41 Anomaly classification results by the Huber Regressor

All techniques were found to have low rates of anomaly detection and high false positive rates. A possible reason for the low detection rates could be linked to high threshold values. Possible adjustment of the threshold values could be considered to improve the results obtained. Due to the higher rate of negative anomalies in the dataset, the techniques were better able to classify anomalies when temperatures were below the climate baseline. The Huber regressor provided better performance based on the prediction performance which resulted in a greater number of anomalies being detected. However, when a low false alarm rate is preferred over higher detection rates, the SVR may be a better fit.

5.2 Spatial Analysis Discussion

Compared to the results of Experiment 1, there was a significant decrease in performance. The techniques had high prediction errors and low classification rates when spatial data were included. In addition, high false positive rates were observed. The reason for the poor performance can be attributed to the increasing complexity due to the input features. Future work should consider selecting only features that contain information about events of interest such as anomalies.

5.3 Analysis of Prediction and Classification Results

The results obtained in Experiment 3 found that SVR detected anomalies better compared to SVC. The poor performance of SVC may be related to the fact that the technique had to learn multiple patterns across four seasons. This increased the complexity as different patterns had to be considered to distinguish between normal and anomalous conditions. As shown in the study by Nayak & Ghosh (2013) and described in Section 2.4.2, two SVM models had to be used to capture the two patterns that occur before extreme events. It might be useful to investigate the use of temperatures for a specific period, e.g., the maximum temperature anomalies in summer (December to March), like a study by Cho et al. (2020) as described in Section 2.4.1.

5.4 Comparison to Related Work

The purpose of this analysis is to compare the results obtained in this study to evaluate how they compare to the benchmarks set by related literature. One gap identified in the review of the literature is the availability of research papers that address the prediction of weather anomalies using machine learning. Most of the work focused on either normal or extreme conditions for the prediction of precipitation rather than temperature. This study has filled this gap by focusing on the prediction of temperatures and evaluating the ability of machine learning techniques to detect anomalies in their predictions. In addition, one of the validation techniques investigated in this study (walk-forward validation) has not been used in related research. This study enabled the evaluation of the technique for weather prediction. The results show that the walk-forward technique provided better insight into the predictive behaviour of the technique over time and improved the predictive and detection performance of the techniques.

Like [9, 11, 18], the result of this study also showed that SVM outperformed other techniques in terms of next day predictions. The results of this study was compared with the study of [18] summarized in Section 2.5.2. This study was used for comparison because the selection of hyperparameters was based on this study. In [18] an MSE of 7.15 was obtained for the SVM and 8.07 for the ANN.

For this comparison, the results of the best performing models reported in Section 4.1.2 was used. In this study, the average MSE for the SVM was 10.96 and the ANN, 54.96±8.1. The MSE obtained in this study was higher than the results obtained by [18]. However, given above the SVM performed better than other techniques by a significant margin. The large variations in MSE can be attributed to the differences in the climates across stations.

In evaluating the performance across different stations, studies by [9, 10] showed that the method used for weather prediction can be applied to different climates. To evaluate the robustness with respect to climate, the performance was compared across three stations. Moreover, [11] found that because the tuning was not performed per station it cannot be guaranteed that the best configuration per station was used. Therefore, the tuning was performed per station. The results showed that climate affects the performance of the techniques and that it is necessary to adjust the hyperparameters for each climate. Climates where temperatures that tend to be more symmetrically distributed (such as Johannesburg) were found to have a lower prediction error compared to stations that are more skewedly distributed with larger spread of possible values (such as Calvinia). An overview of the distribution of the maximum temperatures at selected stations are provided in Section 3.1.2, Figure 11.

There are not many examples that can be used to measure how well the temperature anomalies were predicted in this study. From the available data for precipitation, the authors also faced the problem of a high false positive rate, as shown by the results of Nayak & Ghosh (2013). In this study, extreme precipitation was predicted using the SVM, where the technique predicted less than 1% of all extremes and the probability that the technique correctly predicted an event was about 10%. The study by [8], which focused on an early warning system for heavy rainfall, obtained an F1 between 0.0014 and 0.46 for logistic regression. In this study, the SVR had an F1 value between 0.17 and 0.50, the ANN between 0.03 and 0.40, and the Huber regressor between 0.27 and 0.40 for positive anomalies. For negative anomalies, the SVR had an F1-score between 0.23 and 0.33, the ANN had 0.05 and 0.40, and the Huber regressor between 0.21 and 0.28. Although the models developed in this study had a higher maximum temperature prediction error, the results show that the anomaly detection rates are similar to the results of [8].

Chapter 6

6. Conclusion

Three machine learning approaches for predicting temperature anomalies were evaluated and compared. The objective of the study was to investigate which machine learning techniques and techniques are best suited for weather prediction. The analysis showed that the machine learning techniques were able to predict the next day's temperatures well. However, in terms of anomalies, the techniques performed less satisfactorily as only a small number of anomalies were detected. Another aspect affecting performance is the high false positive rates. The SVM outperformed the other techniques in terms of performance scores. However, the technique had difficulty predicting extreme temperatures. The Huber regressor offered the advantage of being able to model extreme values well, resulting in higher rates of anomaly detection, which in turn was associated with higher prediction errors and false positives. The ANN performed least satisfactorily and failed to predict the next day's maximum temperatures.

Limitations of this study include the lack of expertise in weather anomalies. The scope of the work focused mainly on the broader topic of machine learning, so an in-depth evaluation of the topic, namely weather anomalies, could not be performed. Consideration should be given to involving subject matter experts in decision making regarding input features to ensure that those that influence, and control temperature changes are considered. Future work could include the use of an ensemble model to leverage the power of the SVM and ANN. As seen in the results of [16], the ensemble model had better generalisation performance than the individual machine learning techniques.

References

1. Kotu, V. and B. Deshpande, *Data scrience: Concepts and Practice*. 2019, United States: Morgan Kaufmann publishers.
2. Zhang, G.B., E. Patuwo, and M.Y. Hu, *Forecasting with artificial neural networks: the state of the art.* International journal of Forecasring, 1997. **14**: p. 35-62.
3. Badr, H.S., B.F. Zaitchik, and S.D. Guikema, *Application of Statistical Models to the Prediction of Seasonal Rainfall Anomalies over the Sahel.* Journal of Applied Meteorology and Climatology, 2014. **53**(3): p. 614-636.
4. Grumm, R.H. and R. Hart, *Standardized Anomalies Applied to Significant Cold Season Weather Events: Preliminary Findings.* Weather and Forecasting, 2001. **16**(6): p. 736-754.
5. Dewce, T., *Guidelines on the definition and monitoring of extreme weather and climate events*. 2015, World Meteorological Organisation.
6. R.E. Schulze, M.M., *South African Atlas of Climatology and Agrohydrology*. 2007.
7. Holmstrom, M., D. Liu, and C. Vo, *Machine learning applied to weather forecasting*. 2016, Standord University.
8. Moon, S.-H., et al., *Application of machine learning to an early warning system for very short-term heavy rainfall.* Journal of Hydrology, 2019. **568**: p. 1042-1054.
9. Nayak, M. and S. Ghosh, *Prediction of extreme rainfall event using weather pattern recognition and support vector machine classifier.* 2013. **113**.
10. Ahuna, M.N., T.J. Afullo, and A.A. Alonge, *Rain Attenuation Prediction Using Artificial Neural Network for Dynamic Rain Fade Mitigation.* SAIEE Africa Research Journal, 2019. **110**: p. 11-18.
11. Cramer, S., et al., *An extensive evaluation of seven machine learning methods for rainfall prediction in weather derivatives.* Expert Systems with Applications, 2017. **85**: p. 169-181.
12. Richman, M. and L. Leslie, *The 2015-2017 Cape Town Drought: Attribution and Prediction Using Machine Learning.* Procedia Computer Science, 2018. **140**: p. 248-257.
13. Onyari, E.K. and F.M. LLunga, *Application of MLP Neural Network and M5P Model Tree in Predicting Streamflow: A case study of Luvuvhu catchment, South Africa.* International Journal of Innvovation, management and technology, 2013. **4**(1).
14. Scher, S. and G. Messori, *Predicting Weather Forecast Uncertainty with Machine Learning*. 2018.
15. Tyagi, H., S. Suran, and V. Pattanaik, *Weather - Temperature pattern prediction and anomaly identification using artifical neural network.* International Journal of Computer Applications, 2016. **140**(3): p. 15-21.

16. Perera, A., H. Azamathulla, and U. Rathnayake, *Comparison of different Artificial Neural Network (ANN) training algorithms to predict atmospheric temperature in Tabuk, Saudi Arabia.* Mausam, 2020. **71**: p. 233-244.

17. Cho, D., et al., *Comparative Assessment of Various Machine Learning-Based Bias Correction Methods for Numerical Weather Prediction Model Forecasts of Extreme Air Temperatures in Urban Areas.* Earth and Space Science, 2020. **7**(4): p. e2019EA000740.

18. Radhika, Y. and M. Shashi, *Atmospheric Temperature Prediction using Support Vector Machines.* International Journal of Computer Theory and Engineering, 2009: p. 55-58.

19. Anjali, T., et al. *Temperature Prediction using Machine Learning Approaches*. in *2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*. 2019.

20. Afrifa-Yamoah, E., et al., *Missing data imputation of high-resolution temporal climate time series data.* Meteorological Applications, 2020. **27**(1): p. 1-18.

21. Falessi, D., et al., *On the need of preserving order of data when validating within-project defect classifiers.* Empirical Software Engineering, 2020. **25**(6): p. 4805-4830.

22. Kouassi, K. and D. Moodley, *Automatic deep learning for trend prediction in time series data*. 2020.

23. Bergmeir, C. and J.M. Benítez, *On the use of cross-validation for time series predictor evaluation.* Information Sciences, 2012. **191**: p. 192-213.

24. Parmezan, A.R.S., V.M.A. Souza, and G.E.A.P.A. Batista, *Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model.* Information Sciences, 2019. **484**: p. 302-337.

25. Adhikari, R. and R. Agrawal, *An Introductory Study on Time series Modeling and Forecasting*. 2013.

26. Brockwell, P.J. and R.A. Davis, *Introduction to time series and forecasting*. 2002, New York: Springer.

27. Jitkajornwanich, K., et al., *A survey on spatial, temporal, and spatio-temporal database research and an original example of relevant applications using SQL ecosystem and deep learning.* Journal of Information and Telecommunication, 2020. **4**(4): p. 524-559.

28. Janipella, R. and V. Gupta, *Application of Geographic Information System in Energy Utilization*. 2019. p. 143-161.

29. Endsley, K. and M. Billmire, *Distributed visualization of gridded geophysical data: the Carbon Data Explorer, version 0.2.3.* 2016.

30. Shalev-Shwartz, S. and S. Ben-David, *Understanding machine learning from theory to algorithms*. 2014, United States of America: Cambridge University press.

31. Kotsiantis, S.B., *Supervised Machine Learning: A review of classification techniques.* Informatica, 2007. **31**: p. 249-268.

32. Chollet, F., *Deep Learning with Python*. 2017: Manning Publications Co.

33. Veerasamy, R., et al., *Validation of QSAR Models - Strategies and Importance.* International Journal of Drug design and Discovery, 2011. **2**(3): p. 511-519.

34. Luo, L. and X. Chen, *Integrating piecewise linear representation and weighted support vector machine for stock trading signal prediction.* Applied Soft Computing, 2013. **13**(2): p. 806-816.

35. Willmott, C.J. and K. Matsuura, *Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance.* Climate research, 2005. **30**: p. 79-82.

36. Khair, U., et al., *Forecasting Error Calculation with Mean Absolute Deviation and Mean Absolute Percentage Error.* Journal of Physics: Conference Series, 2017. **930**: p. 012002.

37. Pratama, I. and A.E. Permanasari. *A review of missing values handling methods on time-series data*. in *2016 International conference on Information Technology Systems and Innovation*. 2016. Bandung, Bali.

38. Robinson, A. and J. Hamann, *Forest Analytics with R*. 2011.

39. Acuna, E. and C. Rodriguez, *The Treatment of Missing Values and its Effect on Classifier Accuracy*. 2004. p. 639-647.

40. Moritz, S. and T. Bartz-Beielstein, *imputeTS: Time Series Missing Value Imputation in R.* The R Journal, 2017. **9**.

41. Basheer, I.A. and M. Hajmeer, *Artifical neural networks: fundamentals, computing, design and application.* Journal of microbiological methods, 2000. **43**: p. 3-31.

42. Jayaweera, C.D. and N. Aziz, *Reliability of Principal Component Analysis and Pearson Correlation Coefficient, for Application in Artificial Neural Network Model Development, for Water Treatment Plants.* IOP Conference Series: Materials Science and Engineering, 2018. **458**: p. 012076.

43. Kumar, N. and K.G. Jha, *A time series ANN Approach for weather forecasting* International Journal of Control theory and Computer Modeling, 2013. **3**(1): p. 19-25.

44. Stathakis, D., *How many hidden layers and nodes?* International Journal of Remote Sensing, 2009. **30**: p. 2133-2147.

45. Krose, B. and P. van de Smagt, *An introduction to neural networks*. 1996: University of Amsterdam.

46. Smola, A.J. and B. Schölkopf, *A tutorial on support vector regression.* Statistics and Computing, 2004. **14**(3): p. 199-222.

47. Awad, M. and R. Khanna, *Support Vector Regression*, in *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*, M. Awad and R. Khanna, Editors. 2015, Apress: Berkeley, CA. p. 67-80.

48. Ye, Y., et al., *Robust support vector regression with generic quadratic nonconvex ε-insensitive loss.* Applied Mathematical Modelling, 2020. **82**: p. 235-251.

49. Karmy, J.P. and S. Maldonado, *Hierarchical time series forecasting via Support Vector Regression in the European Travel Retail Industry.* Expert Systems with Applications, 2019. **137**: p. 59-73.

50. Fan, J., Y. Guo, and B. Jiang, *Adaptive Huber regression on Markov-dependent data.* Stochastic Processes and their Applications, 2019.

51. Wang, L., et al. *A New Principle for Tuning-Free Huber Regression*. 2019.

52. Sun, Q., W.-X. Zhou, and J. Fan, *Adaptive Huber Regression.* Journal of the American Statistical Association, 2020. **115**(529): p. 254-265.

53. Chen, B., W. Zhai, and Z. Huang, *Low-rank Elastic-net Regularized Multivariate Huber Regression Model.* Applied Mathematical Modelling, 2020.

54. Organization, W.M. *WMO Guidelines on Generating a Defined Set of National Climate Monitoring Products* 2017.

55. Organisation, W.M. *Guidelines on the definition and monitoring of extreme weather and climate events*. 2018 [cited 2020 2020/11/15]; Available from: https://www.wmo.int/pages/prog/wcp/ccl/documents/GUIDELINESONTHEDEFINTIONANDMONITORINGOFEXTREMEWEATHERANDCLIMATEEVENTS_09032018.pdf.

56. Organisation, W.M. *WMO Guidelines on the calculation of Climate Normals*. 2017 [cited 2020 2020/11/15]; Available from: https://library.wmo.int/index.php?lvl=notice_display&id=20130#.X7BeVWgzY2x.

57. Sillmann, J., et al., *Understanding, modeling and predicting weather and climate extremes: Challenges and opportunities.* Weather and Climate Extremes, 2017. **18**: p. 65-74.

58. Affairs, E., *Climate information and early warning systems to support disaster risk reduction and management under future climate conditions in South Africa*, E. Affairs, Editor. 2014.

59. Banitz, E., *Evaluation of short-term weather forecasts in South Africa.* Water SA, 2001. **27**(4): p. 489-498.

60. Services, S.A.W. *Weather smart news : scientific meteorological and climatological news from the South African Weather Service*. 2018.

61. Services, S.A.W. *Climate Questions*. 2020 [cited 2020 2020/11/15]; Available from: https://www.weathersa.co.za/home/climateques.

62. Sura, P., *A general perspective of extreme events in weather and climate.* Atmospheric Research, 2011. **101**(1): p. 1-21.

63. Aggarwal, C., *Outlier Analysis*. 2013.

64. Chandola, V., A. Banerjee, and V. Kumar, *Anomaly detection: A survey.* ACM Comput. Surv., 2009. **41**(3): p. Article 15.

65. Himani Tyagi, S.S., Vishwajeet Pattanaik, *Weather - Temperature pettern perdiction and anomaly identification using Artificial Neural Network.* International Journal of Computer Applications, 2016. **140**(3): p. 15 - 21.

66. Mozaffari, L., A. Mozaffari, and N. Azad, *Vehicle speed prediction via a sliding-window time series analysis and an evolutionary least learning machine: A case study on San Francisco urban roads.* Engineering Science and Technology, an International Journal, 2014. **6**.

67. Pedregosa, F., et al., *Scikit-learn: Machine Learning in Python.* J. Mach. Learn. Res., 2011. **12**(null): p. 2825–2830.

68. SAWS. *Regional Weather and Climate of South Africa: Gauteng*. 2021; Available from: https://www.weathersa.co.za/Documents/Corporate/Regional%20Weather%20and%20Climate_Gauteng_16042021150251.pdf.

69. Climate-Data.org. *Climate: Northern Cape*. Available from: https://en.climate-data.org/africa/south-africa/northern-cape-470/.

Appendix A:

A1. Chapter 2: Summary of related literature

Table A1 Summary of related work

#	Paper	Dataset	Techniques Used	Configuration Details	Input and Output Variables	Evaluation	Results
1	Application of machine learning to an early warning system for very short-term heavy rainfall Aim: Short term rainfall with a lead time of 3 hours Authors: Moon et al (2019)	Description: Regional meteorological dataset between 2007 and 2012 Source: EWS in South Korea Region: South Korea	• Logistics regression • ANN • 1-nearest neighbour • Random forest • LIBSVM • RIPPER • C4.	Rainfall thresholds: • 70mm in 6 hours or, • 110mm in 12 hours Data pre-processing: The principal component analysis was used to select features. Missing instance with more than three features was removed. Configuration: Training was repeated 30 times. ANN trained on backpropagation	Input: • Date • Time • Wind direction • Scalar wind speed. • Vertical wind speed • Horizontal wind speed • Temperature • Humidity	Metrics: • Precision • Recall • F-measure • Threat score • ETS Protocol: 3-fold cross validation	Logistics regression was deemed to be the most suited for early warning signals based on the F-measure and ETS scores. **Logistics** F-measure: 0.4601 ± 0.0014 ETS: 0.2985 ± 0.0012

					• Atmospheric pressure • MSLP • Rain sensor • Precipitation <u>Output</u>: • Hourly rainfall		
2	Prediction of extreme rainfall event using weather pattern recognition and support vector machine classifier <u>Aim:</u> Extreme rainfall with a lead of 6-48 hours. <u>Authors:</u> Nayak and Ghosh (2013)	<u>Description:</u> Daily values between 1969 to 2008. <u>Source:</u> NCEP/NCAR global reanalysis data set <u>Region:</u> India (Mumbai)	• SVC • Fingerprint technique	<u>Rainfall thresholds:</u> 1.25 times the climatological standard deviation from the climatological mean. <u>Data pre-processing:</u> The data was normalization using the climatological mean. A 21-day rolling centred mean was calculated over the 40-year period. Data partitioning:	<u>Input:</u> • Air temperature • Mean sea level pressure. • Precipitable water • Relative humidity • U-wind and v-wind • Vertical wind velocity	<u>Metrics:</u> • Precision • Recall <u>Protocol:</u> Hold-out validation <u>Data partitioning:</u> Training: 1969-1999: Testing: 2000-2008:	The two phase SVM outperformed the fingerprint technique. Based on the 4392 instances in validation period the following results were obtained: **SVM** • Total extremes predicted: 149.

				• 1969-1999: training • 2000-2008: testing Configuration: Kernels was determined using hit and search method. Quadratic programming sequential minimal optimization and least squares method was used to determine the weights and biases.	Output: • Extreme rainfall indicator		• True extremes: 16 • False positives: 133 **Fingerprint** • Total extremes predicted: 900. • True extremes: 0 • False extremes: 900
3	Rain Attenuation Prediction Using Artificial Neural Network for Dynamic Rain Fade Mitigation Aim: Signal attenuation due to rain. Authors: Ahuna	Description: Rainfall dataset between 2013 and 2018 Source: University of KwaZulu-Natal Region:	• **BPNN**	Configuration: Three layered networks with three neurons in the input and hidden layers Training: TRAINLM training function Learning functions: LEARNGDM adaption function	Input: • Rainfall • Rainfall regimes Output: • Rain attenuation	Metrics: • MSE • Correlation coefficient ($\boldsymbol{R^2}$) Protocol: Hold-out validation (75:25) Data partitioning:	The best performing neural network reached optimum mapping at 371 of 377 epochs with an MSE of 6.017.

	(2019)	Durban, South Africa				Training: 2013 to 2016 Validation/Testing: 2017 to 2018	
4	Application of Statistical Models to the Prediction of Seasonal Rainfall Anomalies over the Sahel <u>Aim:</u> Seasonal precipitation during summer periods (July – Sept) <u>Authors:</u> Badr et al (2013)	<u>Description:</u> **Data set 1:** Rainfall anomaly index historical oceanic and atmospheric data. **Data set 2:** Sahel rainfall anomaly data. (1900 and 2011) <u>Source</u>: Dataset 1 -NOAA NCDC Dataset 2 - University of Washington <u>Region:</u>	• Selected generalized linear model (SGLM) • Generalized additive models (GAM) • Selected generalised additive model (SGAM) • Multivariate adaptive regression spline (MARS) • Tree based: Classification and regression tree (CART)	<u>Data pre-processing:</u> Multicollinearity was solved by performing a PCA on the predictors. All principal components were retained. <u>Configuration:</u> *ANN:* • One input layer with nine units. • A single hidden layer with three units. • An output layer with one output variable. • Hyperparameters configured for the Neural Network are the hidden	<u>Input:</u> • PC1 to 8 and • PC13 (Top performing combination predictors) <u>Output:</u> • Expected seasonal precipitation.	<u>Metrics:</u> • MAE • MSE • MAD • RMSE • R^2 <u>Protocol:</u> Leave one out cross validation where 90% is used for training and 10% for testing.	ANNs predictive performance in terms of accuracy and goodness of fit outperformed all techniques evaluated in this study followed by GLM. **ANN**: COR: 0.81 MAE: 0.48 MSE: 0.31 RMSE: 0.59 **GLM:**

		Sahel region of Africa	Bagged categorical and regression trees (BCART) BART Random forest (RF) • ANN • Generalized linear models (GLM) Baseline models: • Average • Null • Memory	layer size, weigh decay and bagging.			COR: 0.71 MAE: 0.57 MSE: 0.49 RMSE: 0.70
5	An extensive evaluation of seven machine learning methods for rainfall prediction in weather derivatives	<u>Description:</u> Rainfall data between 1990 and 2009	• GP • SVR • RBF • M5 Rules	<u>Data pre-processing:</u> Rainfall rolling totals was calculated. <u>Configuration:</u>	<u>Input</u>: • % Dry days • Average dry spell • Average wet spell	<u>Metrics:</u> • CV(RMSE) • U-test <u>Protocol:</u>	SVR, RBF and GP were the top performing techniques.

	Aim: Rainfall prediction Authors: Cramer et al (2017)	Source: National Centres for Environmental Information National Oceanic and Atmospheric Administration (NOAA NCDC) Region: • USA • Europe	• M5 model trees • K-nearest neighbour (K-NN) • MCRP	iRace (online tool) was used for hyper parameters tuning based on a small subset of similar data. GP and RBF were trained 50 times the median result was taken from the best run. SVR, K-NN, M5P and M5R was only trained once.	• Annual rainfall • Daily volatility • Highest intensity • Interquartile range Output: • Daily rainfall amount	Hold-out validation (90:10) Data partitioning: Training: 1990 to 2009 Testing: 01/01/2010 to 21/12/2010	Average RMSE across the different cities for accumulated prediction SVR: 1.90 RBF: 2.05 GP: 3.25
6	The 2015-2017 Cape Town Drought: Attribution and Prediction Using Machine Learning Aim: Cool season rainfall Authors: Richman and Leslie (2018)	Description: Historical precipitation data for Cape Town between 1920 to 2017 Region: Cape Town, South Africa	• SVR (polynomial and radial basis function were used)	Configuration Polynomial SVR: Kernel = Poly E = 1 C = 1 Configuration Radial Basis Function SVR: Kernel = RBF E = .01	Input: • Precipitation (mm) • Temperatures Output: • Precipitation	Metrics: • R^2 • MAE Protocol: 10-fold cross validation	Polynomial SVR performed better than the RBF SVR the MAE is 73.04 and 72.61, respectively. Correlation: ~.56 for the two SVR methods.

				C = 50			
7	Weather - Temperature pattern prediction and anomaly identification using artificial neural network Aim: Predicting daily temperature patterns. Authors: Tyagi (2016)	Description: 20 years between 1 January 1994 to 31 December 2014. Source: CEES Weather station Region India: Delhi and Mumbai	• **ANN**	Data pre-processing: Spline interpolation was used for missing data. Configuration: • Hidden layer: 1 • Neurons: 4 • Training function: Levenberg-Marquardt • Transfer function: Tanh sigmoid Hyperparameters tuning was performed on one station to find the optimal values for the following: • Number of neural networks, • Number of neurons per layer, • Sample sizes,	Input: Air/soil temperatures Relative humidity Barometric pressure Wind speed Solar Flux density Total solar flux Dew point temperature Heat index Wind Rainfall Rainfall Intensity Hourly/Daily Evapotranspiration Output: Daily temperatures	Metrics: • **MSE** Protocol: Hold-out validation (80:10:10)	ANN on long term temperature analysis is more effective **Input-Output-Data ratio: 19:1** 66.66% of test cases achieved a minimum MSE of +0.21 degree Celsius. **Input-Output-Data ratio: 4:1** 48.53% of test cases achieved a minimum MSE of +0.95 degree Celsius.

				• Input-output data ratios • Transfer function and • Stopping criteria.			
8	Comparison of different Artificial Neural Network (ANN) training techniques to predict atmospheric temperature in Tabuk, Saudi Arabia <u>Aim:</u> Temperature prediction <u>Authors:</u> Perera et all (2020)	<u>Description:</u> Monthly weather data for 30 years from 1986 to 2015 <u>Source:</u> Saudi General Authority of Meteorology and Environment Protection <u>Region:</u> Saudi Arabia - Tabuk	• **ANN**	<u>Configuration:</u> 10 hidden neurons 2 delays The following training techniques was used: • BFGS Quasi Newton (BFG) • Conjugate gradient with Powell-Beale restarts (CGB) • Levenberg-Marquardt (LM) • Scaled Conjugate Gradient (SCG) • Fletcher-Reeves update Conjugate Gradient technique (CGF) • One step secant (OSS)	<u>Input:</u> monthly rainfall Maximum and minimum humidity Maximum and minimum air pressure Maximum and minimum air temperature <u>Output:</u> Monthly____maximum and minimum temperature	<u>Metrics:</u> • MSE • R^2 <u>Protocol:</u> Hold-out validation (70:15:15)	The following training techniques outperformed the rest: • BFG specifically for minimum temperature, • LM for maximum temperature, • SCG In terms of correlation coefficient BFG: 0.95

				• Polak-Ribiere update conjugate gradient (GCP) • Resilient Back propagation (RP) MATLAB was used to develop techniques			LM: 0.94 SCG: 0.93
9	Comparative Assessment of Various Machine Learning-Based Bias Correction Methods for Numerical Weather Prediction Model Forecasts of Extreme Air Temperatures in Urban Areas Aim: Bias correction Authors: Cho et al (2020)	Description: July and August from 2013 to 2017 Source: Automatic Weather Stations (AWSs) operated by KMA in Seoul Region: Seoul, South Korea	• Random forest • SVR • ANN • Multi-model ensemble	Data pre-processing: Missing data was replaced using Cokriging technique Configuration: ANN: two hidden layers costing of 22 and 29 neurons, respectively. With the Rectified linear unit (RELU) activation function and Adam optimizer.	Input: • Latitude • Longitude • Elevation • Slope • Solar radiation Output: Next day maximum and minimum temperature	Metrics: • RMSE • $\boldsymbol{R^2}$ • Bias Protocol Used hindcast validation and leave-one-station-out-cross validation	The MME had better generalization performance than the three single machine learning models. **SVR:** RMSE: 1.58 Bias: -0.15 R^2: 0.78 **ANN:** RMSE: 1.58 Bias: -0.16

							R^2: 0.77 **MME**: RMSE: 1.55 Bias: -0.13 R^2: 0.78
10	Atmospheric Temperature Prediction using Support Vector Machines Aim: Temperature Prediction Authors: Radhika and Shashi (2009)	Description Daily temperatures between 2003 and 2008. Source: University of Cambridge Region: United Kingdom	• SVR • ANN	Data pre-processing: Missing values is replaced with the mean maximum temperature for that month. The data was normalized using the z-score Sliding window size: 5 (optimal value found through experimentation) Configuration: MPL consisted of 3 layers using the sigmoid activation function. The number of neurons in the hidden layer are (2i+1) (i is the number of inputs) and was	Input: *N* previous maximum temperature Output: Maximum temperature	Metrics: • **MSE** Protocol: Hold-out validation (70:15:15) Data partitioning: 5 Years of data for training and half a year for testing	SVM performed better than the MLP with back propagation. The results are based a window size of 5 **SVM** 7.15 **MLP** 8.07

				trained with the back propagation technique SVR: RBF kernel, a grid search was performed to find the optimal parameter values $\varepsilon \; and \; C$. $\varepsilon = 1{,}2{,}0.001{,}0.01{,}0.1{,}1$ $C = (1, 10{,}100), (0.5, 0.7, 0.9\}$			
11	Temperature prediction using machine learning <u>Aim:</u> Temperature prediction <u>Authors:</u> Anjali et al (2019)	<u>Description</u>: Daily hourly data between 2009 to 2012 <u>Source:</u> Agricultural Research Station, Mannuthy, Thrissur <u>Region:</u>	• MLR • SVR • ANN	<u>Data pre-processing:</u> • Missing data filled using linear interpolation <u>Configuration:</u> WEKA used to develop the experimental framework.	<u>Input:</u> Humidity Pressure Wind direction Wind speed <u>Output:</u> Temperature	<u>Metrics:</u> • ME • MAE • RMSE • Average error (AE) • $\boldsymbol{R^2}$ <u>Protocol:</u> 5-fold cross validation was used	The MLR provided more precise results compared the SVM and ANN. **MLR** RMSE: 1.8568 R^2: 0.8119 **ANN** RMSE: 2.0100 R^2: 0.7932

		Kerala: Malappuram, Calicut, and Thrissur					**SVR** RMSE: 1.8727 R^2: 0.8110
12	Application of MLP Neural Network and M5P Model Tree in Predicting Streamflow: A case study of Luvuvhu catchment, South Africa Aim: Using rainfall runoff to predict streamflow. Authors: Onyari and Llunga (2013)	Description: **Data set 1:** Rainfall data **Data set 2:** streamflow measurement data between 2007 July and 2010 July Source: Data set 1: SAWS Data set 2: Department of Water Affairs Region: Limpopo, South Africa	• MLP-ANN • M5P Model tree	Data pre-processing: Sliding window size: Previous 5 days input was used for training. Feature reduction was performed using the mRMR method. Configuration: One input layer One hidden layer with four hidden nodes. WEKA software was used for M5P model tree and MLP-ANN calibration and verification.	Input: • Rainfall • Streamflow measurements Output: Rainfall runoff	Metrics: • RMSE • MAE • R^2 Protocol: 66% of data was used for training and the remainder for testing.	M5P Model tree outperformed MLP-ANN in predicting the river streamflow. **M5P-model tree** RMSE - 2.666 R^2– 0.89 **MLP-ANN** RMSE: 3.42. R^2: 0.82

13	Predicted weather forecast uncertainty Aim: Forecast uncertainty Authors: Scher and Messori (2018)	Details: Daily forecast consisting of 26 years between 1990 to 2016 Source: GEFS reforecast dataset Region: Europe	• Convolutional neural network (CNN) • Nearest neighbour • SVR • Linear regression • Random forest regression	Data pre-processing: A 30-day running climatology was calculated. The input variables and target values were normalized to a zero mean and unit variance Configuration: Hyperparameter tuning done through trial-and-error basis. The CNN architecture consisted of an input layer with three units and a hidden layer with 32 units. Activation function: ReLU Dropout probability: 0.7 Optimizer: Adam The network was trained for 30 epochs, the model with the highest score per epochs was selected. The network was retrained 10 times to ensure the validity of	Input: • Geopotential height at 500hPa • Meridional Zonal wind at 300hPa Output: Forecast uncertainty	Metrics: • F1-score • R^2 • Ensemble spread. Protocol: Hold-out validation (75:25) Data partitioning: Training: 1990 to 2008 Validation/Testing: 2010 to 2016	CNN significantly outperformed other methods and performed computationally efficiently

				results. Forecast error was linearly detrended.			

Appendix B:

B.1 Chapter 3: Maximum temperatures at Johannesburg

Johannesburg's climate have cool summers with summer rainfall [68]. Average temperatures for the 20-year period are 22.56°C, with temperatures varying between 19.4°C and 26°C. Average temperatures have decreased by 0.30°C compared to the climate baseline, however, the intensity of temperature conditions has increased. The increase in intensity is reflected in the decrease of 2.32°C of the coldest maximum temperatures, and an increase of 8.54°C in the hottest temperature during the period. Figure B.1 shows the daily averages observed compared to the baseline value.

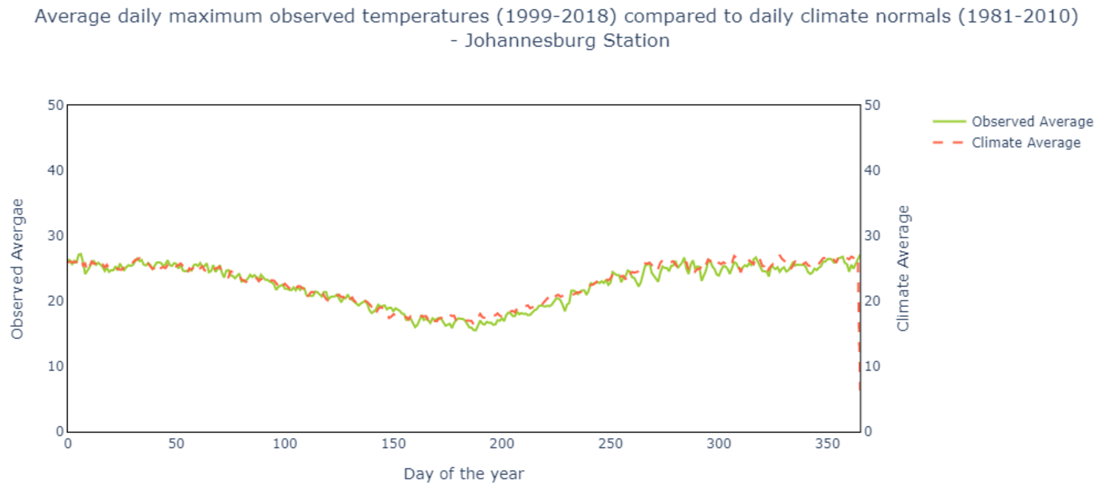

Figure B.1 Average daily maximum temperatures compared to the daily climate baseline in Johannesburg

Johannesburg had a total of 364 anomalous observations out of 7305 observations. Negative anomalies are more common than positive anomalies at this station, with a total of 325 negative and 39 positive anomalies observed. On average, about 5% of the observations per year are anomalous. Table B.1 provides an overview of the number of observed anomalies per year between 1999 and 2018.

Table B.1 Number of negative and positive anomalies per year for Johannesburg

Year	Negative Anomaly	Positive Anomaly	Total Anomalies Per Year	Average Anomalies per Year
1999	11	0	11	3%
2000	26	0	26	7%
2001	23	0	23	6%
2002	21	1	22	6%
2003	14	2	16	4%
2004	18	0	18	5%
2005	3	3	6	2%
2006	13	1	14	4%
2007	24	0	24	7%
2008	14	1	15	4%
2009	21	1	22	6%
2010	14	1	15	4%
2011	21	2	23	6%
2012	24	0	24	7%
2013	17	0	17	5%
2014	17	1	18	5%
2015	9	12	21	6%
2016	11	4	15	4%
2017	16	1	17	5%
2018	8	9	17	5%
Total	**325**	**39**	**364**	5%

As can be seen from Figure B.2, the number of negative anomalies occurring each year is gradually decreasing. The decrease in negative anomalies is accompanied by an increasing trend in maximum temperatures, with the warmest year recorded in 2015, which also had the most positive anomalies. It is also observed that the number of positive anomalies increases from 2015.

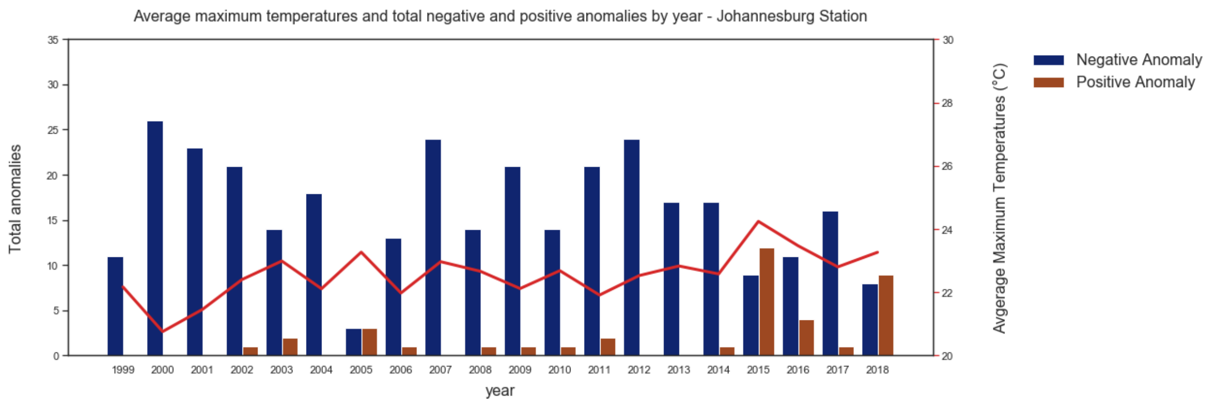

Figure B.2 Average maximum temperatures compared to total anomalies by year between 1999 and 2018 in Johannesburg

B.2 Chapter 3: Maximum temperatures at Calvinia

The climate in Calvinia is classified as a cold desert climate [69]. The average maximum temperatures recorded during the observation period (1999-2018) are 24.93°C, which is 0.08°C cooler than the climate baseline (1981-2010) of 25.01°C. The intensity of hot and cold maximum temperatures has increased, with the hottest recorded temperature being 8.81°C warmer compared to baseline. Figure B.3 shows the daily averages observed compared to the baseline value.

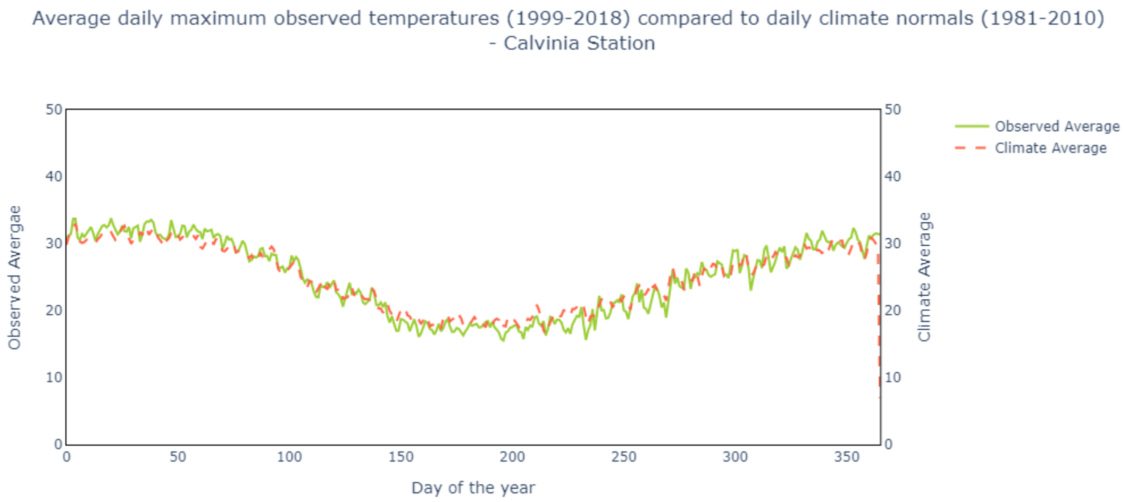

Figure B.3 Average daily maximum temperatures compared to the daily climate baseline in Calvinia

The Calvinia dataset contained a total of 271 maximum temperature anomalies (between 1999 and 2018), of which 244 were negative anomalies and 57 were positive anomalies. Calvinia recorded a low frequency of temperature anomalies, but there has been an increase in the percent anomalies per year since 2009. Table B.2 summarises the number of anomalies observed per year between 1999 and 2018.

Table B.2 Number of negative and positive anomalies per year for Calvinia

year	Negative Anomaly	Positive Anomaly	**Total Anomalies per Year**	**Average Anomalies per Year**
1999	7	3	10	3%
2000	9	1	10	3%
2001	13	2	15	4%
2002	17	0	17	5%
2003	10	1	11	3%
2004	7	0	7	2%
2005	6	3	9	2%
2006	10	0	10	3%
2007	10	1	11	3%
2008	8	4	12	3%
2009	16	1	17	5%
2010	12	2	14	4%
2011	16	2	18	5%
2012	16	5	21	6%
2013	16	3	19	5%
2014	14	2	16	4%
2015	10	3	13	4%
2016	7	5	12	3%
2017	9	1	10	3%
2018	12	7	19	5%
Total	**225**	**46**	**271**	

From Figure B.4, the following can be seen with respect to anomalies. Negative anomalies are more common at the station than positive anomalies. Moreover, an increase in negative anomalies is observed in the following periods (2001-2002, 2009 and 2011-2014). As can be seen from the average temperatures, these years are associated with cooler average temperatures. It can also be observed that there is an upward trend in positive anomalies starting in 2008. It is also interesting to note that the highest annual average temperatures were recorded in 2017, however, this was not associated with any significant change in anomalies compared to other years.

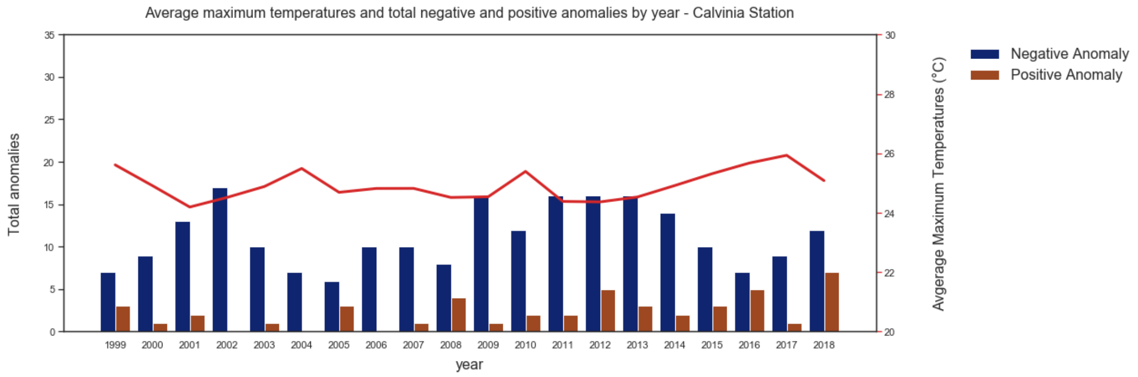

Figure B.4 Average maximum temperatures compared to total anomalies by year between 1999 and 2018 in Calvinia

B.3 Chapter 3: Maximum temperatures at Upington

Hot desert conditions prevail in Upington [69]. The average temperatures recorded during the observation period were 29.82°C, which is 0.23°C warmer than the climate baseline. The increase in temperatures can also be seen in the minimum and maximum temperatures recorded during the observation period. Figure B.5 shows the daily averages observed compared to the baseline value.

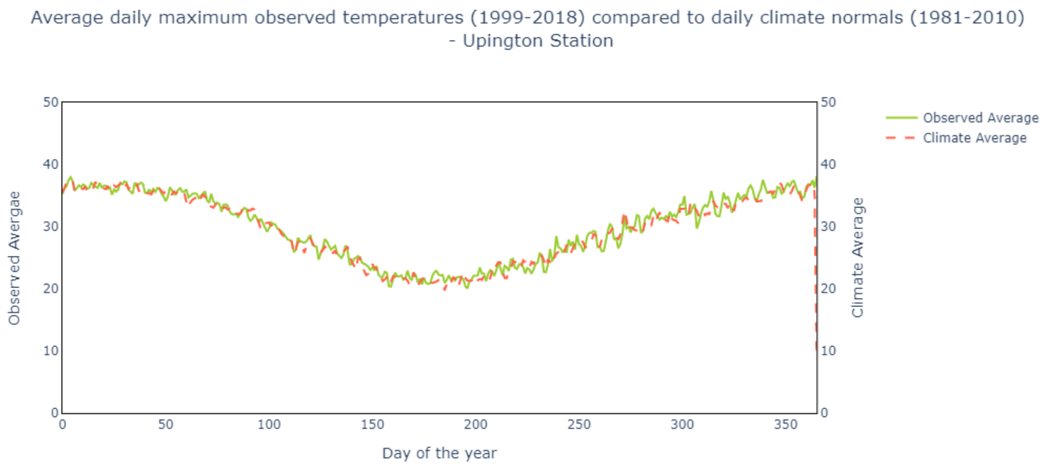

Figure B.5 Average daily maximum temperatures compared to the daily climate baseline in Upington

The Upington station had a total of 320 anomalies, of which 268 were negative and 52 were positive. Table B.3 summarises the number of anomalies observed per year between 1999 and 2018.

Table B.3 Number of negative and positive anomalies per year for Upington

year	Negative Anomaly	Positive Anomaly	Total Anomalies per Year	Percentage Anomalies per Year
1999	15	1	16	4%
2000	17	0	17	5%
2001	10	1	11	3%
2002	20	0	20	5%
2003	16	0	16	4%
2004	6	0	6	2%
2005	5	1	6	2%
2006	17	0	17	5%
2007	14	4	18	5%
2008	6	1	7	2%
2009	11	2	13	4%
2010	8	2	10	3%
2011	14	2	16	4%
2012	25	6	31	8%
2013	19	6	25	7%
2014	19	1	20	5%
2015	16	10	26	7%
2016	7	6	13	4%
2017	10	4	14	4%
2018	13	5	18	5%
Total	**268**	**52**	**320**	

Figure B.6 shows that more negative than positive anomalies occurred. It can also be observed that a decrease in annual average temperatures is accompanied by a higher number of negative anomalies, as was observed between 2000 and 2002 and then again between 2012 and 2014. However, the number of positive anomalies increases from 2007 onwards, with most recorded in 2015, which is also the warmest temperature during this period. Furthermore, the increase in positive anomalies is accompanied by an increase in average maximum temperatures, as shown by the average maximum temperatures.

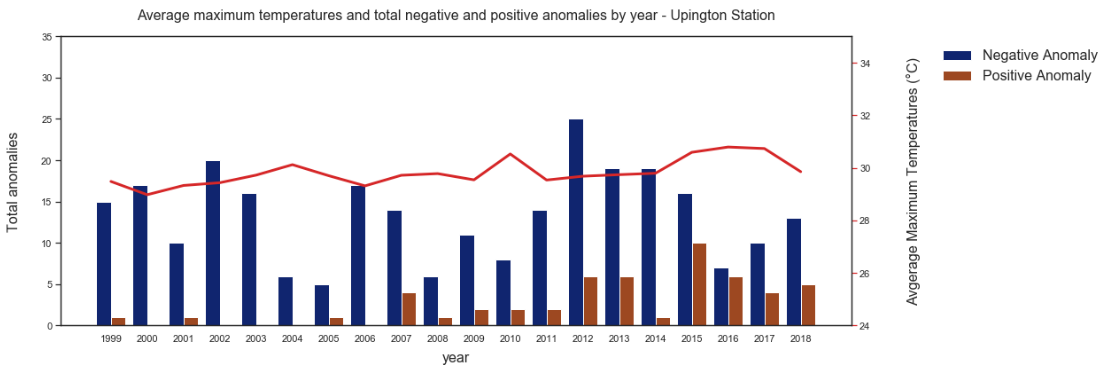

Figure B.6 Average maximum temperatures compared to total anomalies by year between 1999 and 2018 in Upington

Appendix C:

C1. Chapter 4: Results of ANN training

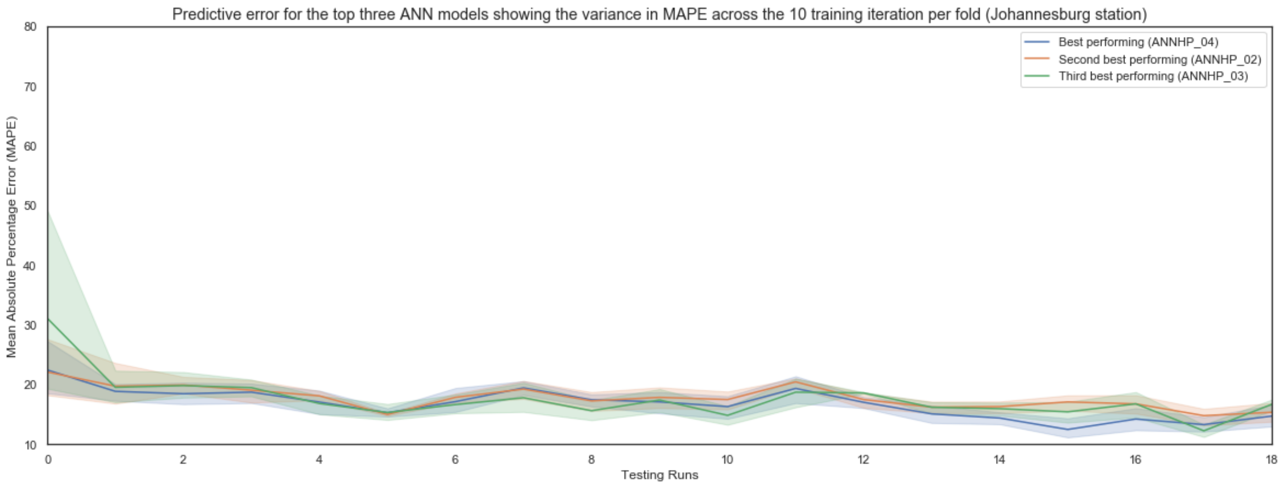

Figure C.1 Predictive error of the top three ANN trained at the Johannesburg station showing the variances in the MAPE across the 10 training runs per training iteration

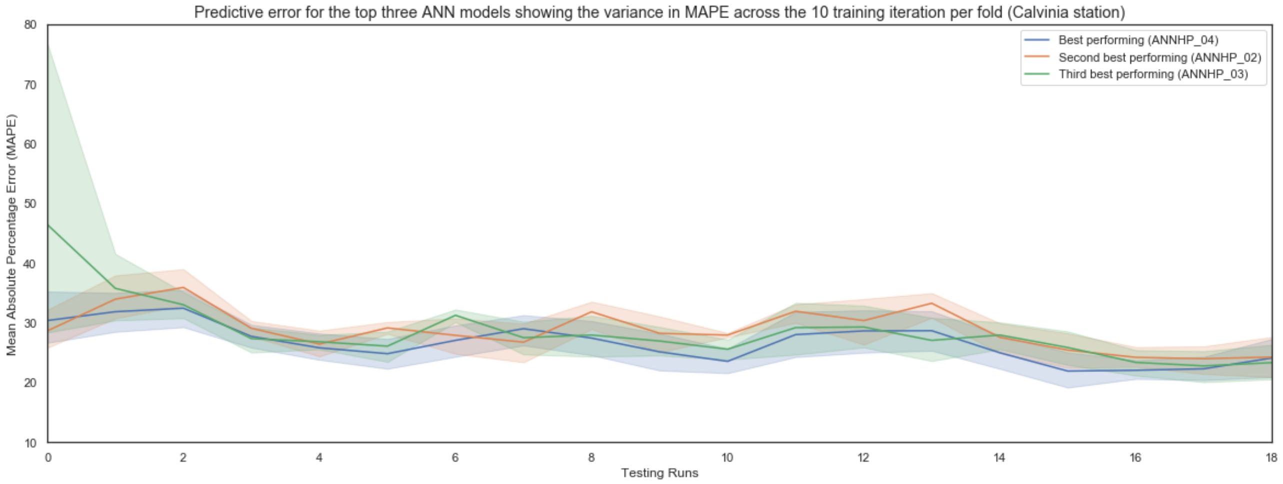

Figure C.2 Predictive error of the top three ANN trained at the Calvinia station showing the variances in the MAPE across the 10 training runs per training iteration

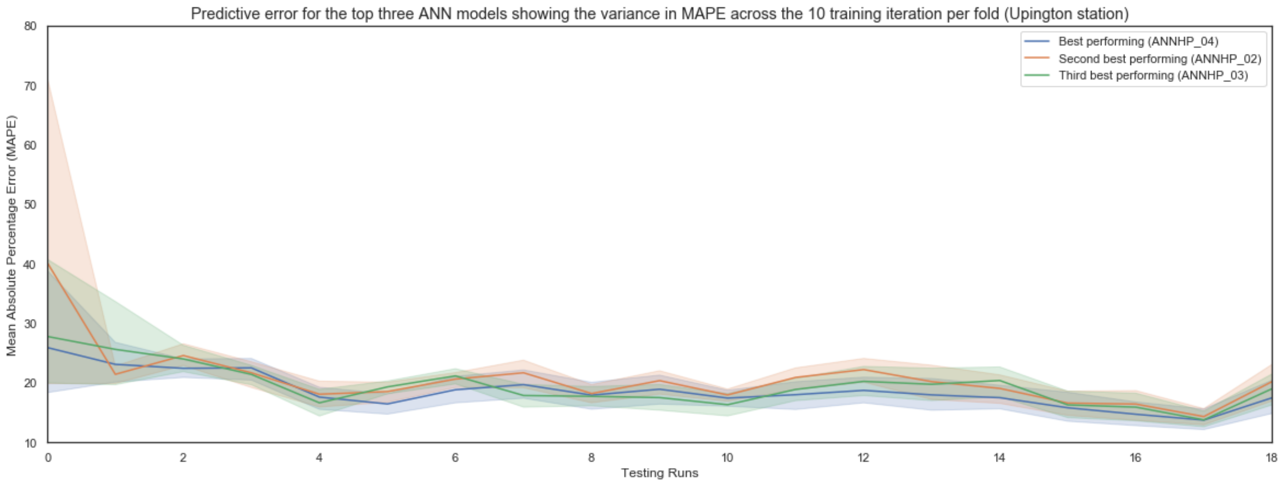

Figure C.3 Predictive error of the top three ANN trained at the Upington station showing the variances in the MAPE across the 10 training runs per training iteration