Accurate Prediction of Global Mean Temperature through Data Analytics

Debdarsan Niyogi (debdarsann@iisc.ac.in)
Indian Institute of Science Bangalore
Jayaraman Srinivasan
Indian Institute of Science Bangalore

Research Article

Keywords:

Posted Date: September 14th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-2055545/v1

Accurate Prediction of Global Mean Temperature through Data Analytics

Debdarsan Niyogi*
J. Srinivasan†
Divecha Centre for Climate Change,
Indian Institute of Science, Bangalore 560012, India

September 12, 2022

*E-mail: debdarsann.iisc.ac.in (corresponding author) ORCID ID: 0000-0002-2376-5482
†E-mail: jayes@iisc.ac.in, Ph: +91-80-22933068

Abstract

There is a need to predict how the Global Mean Temperature (GMT) will evolve in the next few decades so that all nations can plan their reduction in the use of fossil fuels. The ability to predict historical data is a necessary first step towards the actual goal of making long-range forecasts. This paper examines the advantage of statistical and simpler Machine Learning (ML) methods instead of directly using complex ML algorithms and Deep Learning Neural Networks (DNN). Systematic data preparation methods have been applied as they have often been shown to offer predictive advantages to ML methods. The GMT time series is treated both as a univariate time series and also cast as a regression problem. Some steps of data preparation were found to be effective. Some simple ML methods did as well or better than the more well-known ones showing merit in trying a large bouquet of algorithms as a first step. The GMT data were divided into training and testing datasets. Predictions for the annual GMT testing data were better than that published so far, with the lowest RMSE value of 0.02 °C. RMSE for five-year mean GMT values for the test data ranged from 0.00002 to 0.00036 °C.

1 Introduction

The global mean temperature of the earth is 1.1 °C above its pre-industrial value. This rapid increase is a cause for global concern. Hence most of the countries in the world signed an agreement in Paris in 2015 to limit global warming to below 2 °C (preferably 1.5 °C). Most nations have not committed to a rapid reduction of emission of greenhouse gases and hence the Global Mean Temperature (GMT) will continue to increase. When will the GMT reach the value of 1.5 °C above the pre-industrial value? Can this be forecast using climate models or methods of data analytics? Climate models have been used to predict the GMT at 2100 based on various assumptions about the increase in greenhouse gases. These models were not very successful, and not accurate for predicting the increase in GMT in the near future. A data-driven approach to this problem is an alternative. Further, to enhance confidence in forecasts, it is necessary as a first step to predict recent trends in GMT. Hence, it is useful to examine if data analytic models will provide accurate predictions of historical record of the global mean temperature. Data analytic methods range from simple algorithms based on statistical analysis and simple Machine learning to complex ML models like Deep Learning Neural Networks (DNN). The more complex the method is, greater is the effort and expense in computation. A few publications strongly recommend using a gamut of simple ML models prior to attempting the more complex ones. Makridakis et al.[1] studied 1045 monthly time series used in the M3 Competition of the International Institute of Forecasters to compare the skills of eight traditional statistical models against popular ML methods. He found that statistical models showed better skills than ML models. Crone et al.[2] studied the data sets used in NN3 Neural Networks Forecasting competition under different conditions like (a) short and long series (b) seasonal and non-seasonal (c) short, medium and long forecast horizons. Quite surprisingly, the simple Theta and ARIMA models outperformed other algorithms including neural networks. The value and impact of knowledge from statistics on the methods and results of data science approaches have been pointed out by Weihs and Ickstadt[3]. Hence, in the present work only statistical and simple ML methods were used as the first step to predict the GMT for 5, 10 and 15 years and also mean GMT over one and a half decade.

2 Literature Review

The literature has centered around a few lines of thought. One is about analyzing regional temperature data to infer statistical properties. Lorentzen[4] examined sea temperature data from Norway with an aim to infer changes in climate. Mali[5] analyzed multifractal characteristics to examine if there are long term trends in global temperature anomaly. Silva et al.[6] characterized the variations of global sea surface temperature using wavelets. Coleman[7] used an advanced variant of Empirical Mode Decomposition to extract temperature cyles in global, hemispherical and tropical temperature anomalies. There have been many other studies of a similar nature which are aimed at understanding nature of the time series which will eventually go into forecast. However, predicting or forecasting global mean temperature with the yearly data available is also of great interest.

Mulrennan et al.[8] compared one standard statistical model ARIMA with two ML models: Artificial Neural Network (ANN) and Random forest. The time series data were on energy consumption in pharmaceutical manufacturing facility. Considerable effort was spent in developing ML models and they found Random Forest performed the best. This is indicative that ANN might perform poorly than a much simpler ML algorithm. It would have been interesting if predictions were made with other classical statistical models, which is proposed in this work.

The ML literature on GMT focused on developing correlations with greenhouse gas emissions, results of GCM and so on. Very few studies could be found on characterizing or predicting GMT by treating it as a time series. Partial Least Square Regression (PLR) was used by Brown and Caldeira [9] to analyze the GMT time series. Observed globally gridded Surface Air Temperature (SAT) anomalies were obtained from four primary data sets. SAT fields were used as predictor variables of GMT. PLSR was performed between predictors made up of gridded SAT fields and predictands of subsequent GMT deviations. Model (called BC2020) validation was achieved via leave-one-out-cross validation for years prior to 2000 (which is commonly referred to as hindcast mode) and through out-of-sample predictions made on the post-2000 data (which is commonly referred to as forecast mode). The walk-forward method was adopted where the window size varied from 1 to 4. The lowest value of RMSE in the forecast was 0.09. One noteworthy observation of Brown and Caldeira [9] is that the mean GCM had a larger RMSE than the Naïve benchmarks which suggests that the average GCM has difficulty in predicting GMT deviations.

Monthly data during 1974-2020 from data of 270 air temperature measuring stations in Turkey were analyzed by Citakoglu [10]. Data from 165 stations were used for training and the others for testing. Month numbers, latitude, longitude, and altitude variables were used as input data. Performances of four different ML methods (a) Long Short-Term Memory (LSTM), (b) Support Vector Machine Regression (SVMR), (c) Gaussian Process Regression (GPR), and (d) Multi-Gene Genetic Programming (MGGP) were compared. The models were used to predict maximum, minimum and average temperatures. A typical value of RMSE (lowest for the prediction of the average temperature) for all the four models was 1.28.

Panda et al. [11] used DNN methods (LSTM, GRU, Bi-LSTM, Bi-GRU, S-LSTM, S-GRU, SBi-LSTM, SBi-GRU) to model GMT data from Berkley's Earth between January 1900 and June 2020 with a train test split of 80:20. They compared the predictions with those of the statistical model SARIMAX. The latter was found to be more accurate with an RMSE of 0.0838.

Viola et al. [12] employed nonlinear signal processing tools to analyze temperature time-series data obtained at different locations in the world from January 1989 to December 2008. The aim was to establish state-space reconstruction and make predictions. Special techniques were used to reduce noise contamination. The testing period was about eight years. The average error is defined as the per cent of the difference between observed and predicted values over the time series divided by the observed mean. The error ranged from 0.6 to 5.3% for the different locations.

Romilly [13] used univariate time series of GMT to develop a forecasting model over the short-term horizon (5 to 10 years). The statistical techniques used include seasonal and non-seasonal unit root testing, as well as ARIMA and GARCH modelling. The dataset consists of 1602 monthly observations from 1870:1 to 2003:6 on global near-surface mean temperature, expressed as differences (or anomalies) between the actual monthly temperature and the average temperature for 1961–90. Temperature is measured in degrees Celsius, and the average temperature from 1961–1990 is 14.08 °C. The SARMA–GARCH(1,1)-M model gave the best RMSE = 0.111.

The monthly record of absolute surface temperature was modelled by Ye et al. [14] using a Deterministic and Stochastic Combined (DSC) approach, where the deterministic part consists of trend and cyclic oscillations, and the stochastic part is the remaining pattern involving SARIMA. Several DSC models were constructed and tested. The best RMSE obtained was 0.1112.

Himika et al. [15] used an ensemble approach for GMT prediction and reached a range of RMSE from 0.65 to 2.41 for different algorithms tried. Three models were picked out for assembling, and the RMSE 0.67 was obtained for the said ensemble model.

There are climate models [16], like the Global Circulation Model (GCM), that are physics-based simulations of real-world phenomena, developed to advance the understanding of climatic behaviour. Later, results of some of these are compared with those obtained in this work.

It can be seen that accuracy attained in prediction of historical data of GMT is still unsatisfactory.

3 Exploratory Data Analysis

3.1 Data and Characterization

The GMT data was obtained from NASA's website from 1880 to 2020 (https://data.giss.nasa.gov/gistemp/). The data column which lists the GMT average from January to December, namely "J-D", for each year, has been used in this univariate study. The line plot of GMT presented in Figure 1 shows an increasing trend. The descriptive statistics are given in Table 1. The diagnostic plots are shown in Figure 2 and the Kernel Density Estimation (KDE) [17] using default parameters is depicted in Figure 3. It is seen that the KDE exhibits multi-modality.

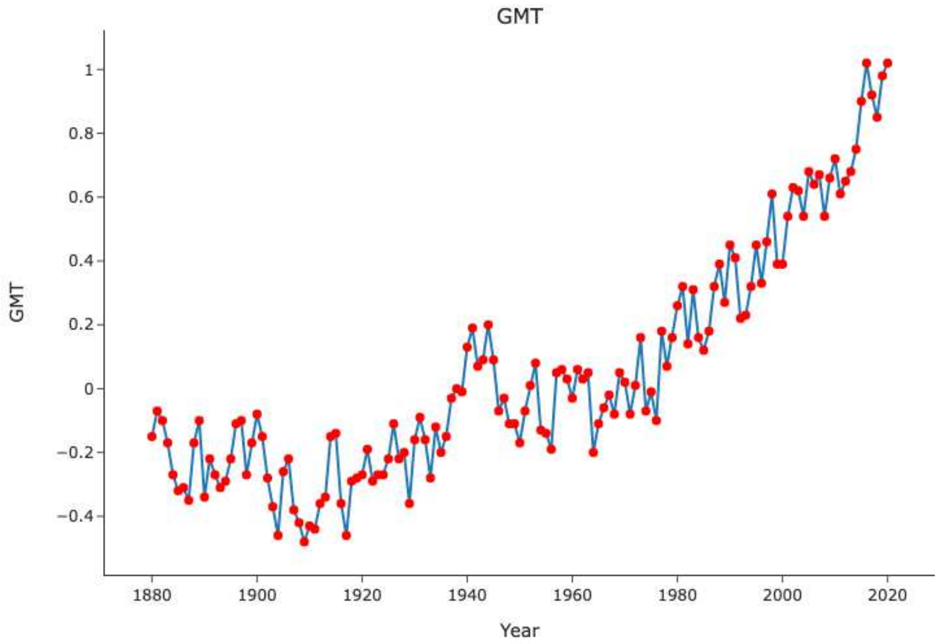

Figure 1: Line plot of GMT data (1880-2020)

Table 1: Descriptive Statistics

Count	141
Mean	0.0504
Standard Deviation	0.3579
Min. Value	-0.48
25%	-0.2
50%	-0.07
75%	0.23
Maximum Value	1.02
Kurtosis	0.0430
Skewness	0.9049

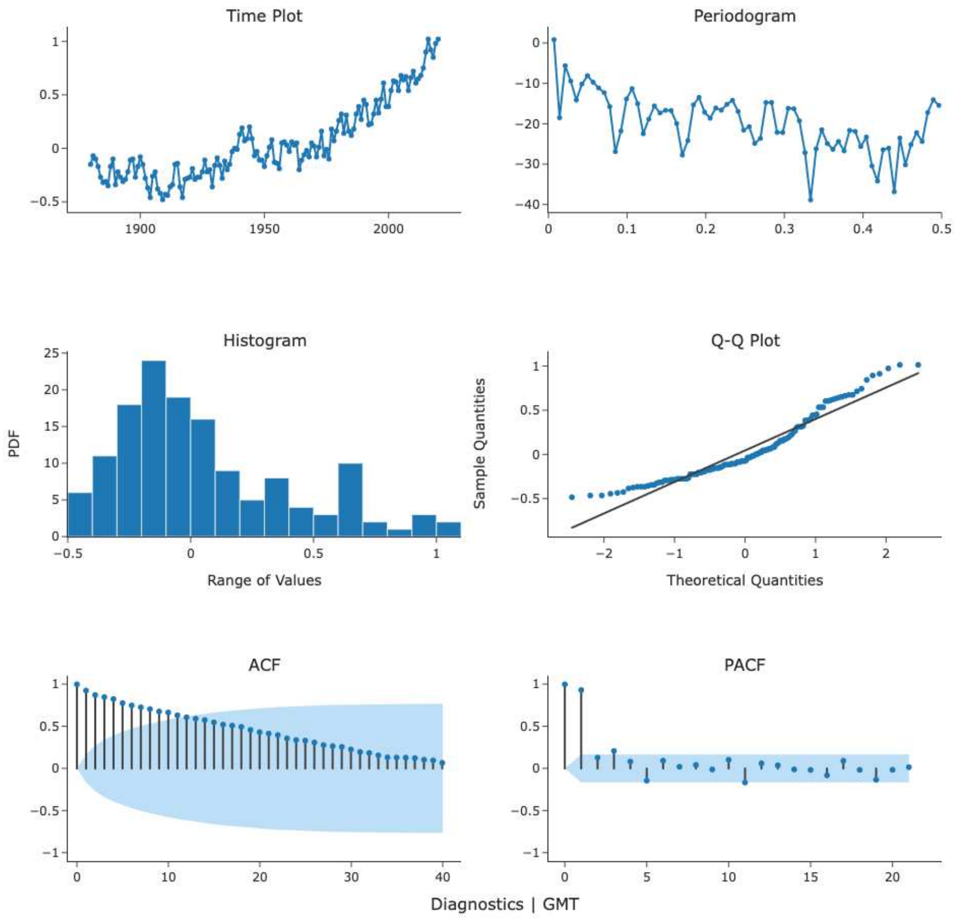

Figure 2: Diagnostic Plots

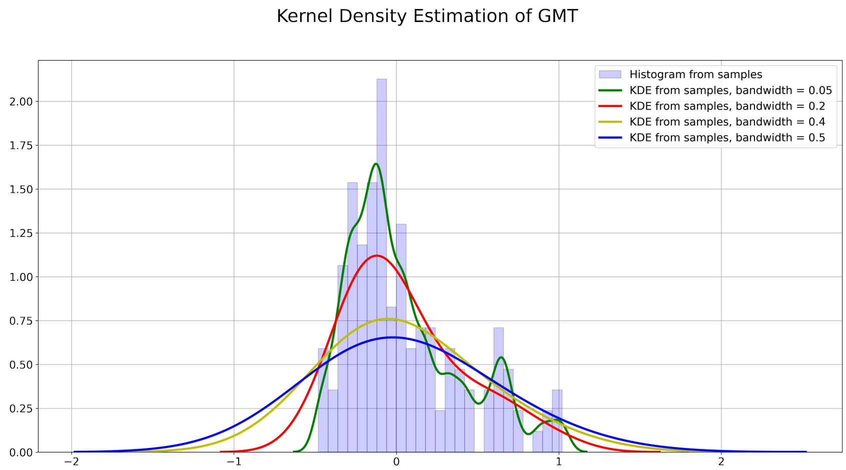

Figure 3: Kernel Density Estimation

The Normality test showed that the distribution is not Gaussian. The Augmented Dickey-Fuller ("ADF") test and Kwiatkowski-Phillips-Schmidt-Shin ("KPSS") test indicate that the time series is non-stationary. The time series after first-order differencing is shown in Figure 4. Both ADF and KPSS tests are repeated on the first-ordered differenced data and they confirm the time series is stationary after differencing.

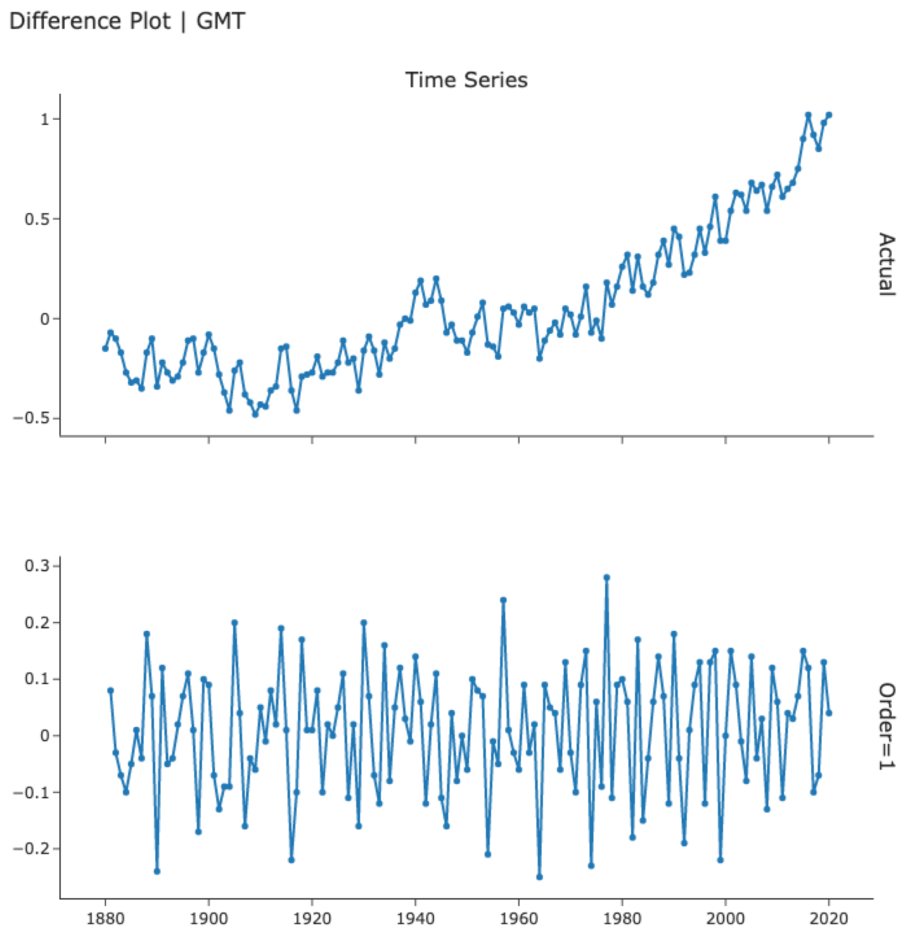

Figure 4: First order difference of GMT data

The systematic part of the GMT time series was decomposed into the level, trend and seasonality. Since GMT data contains negative values, the multiplicative decomposition is not applicable. The upward trend of GMT can be seen in Figure 1. There cannot be any seasonality since GMT data is an annual average, and is confirmed in Figure 5 where seasonal as well as noise components are zero.

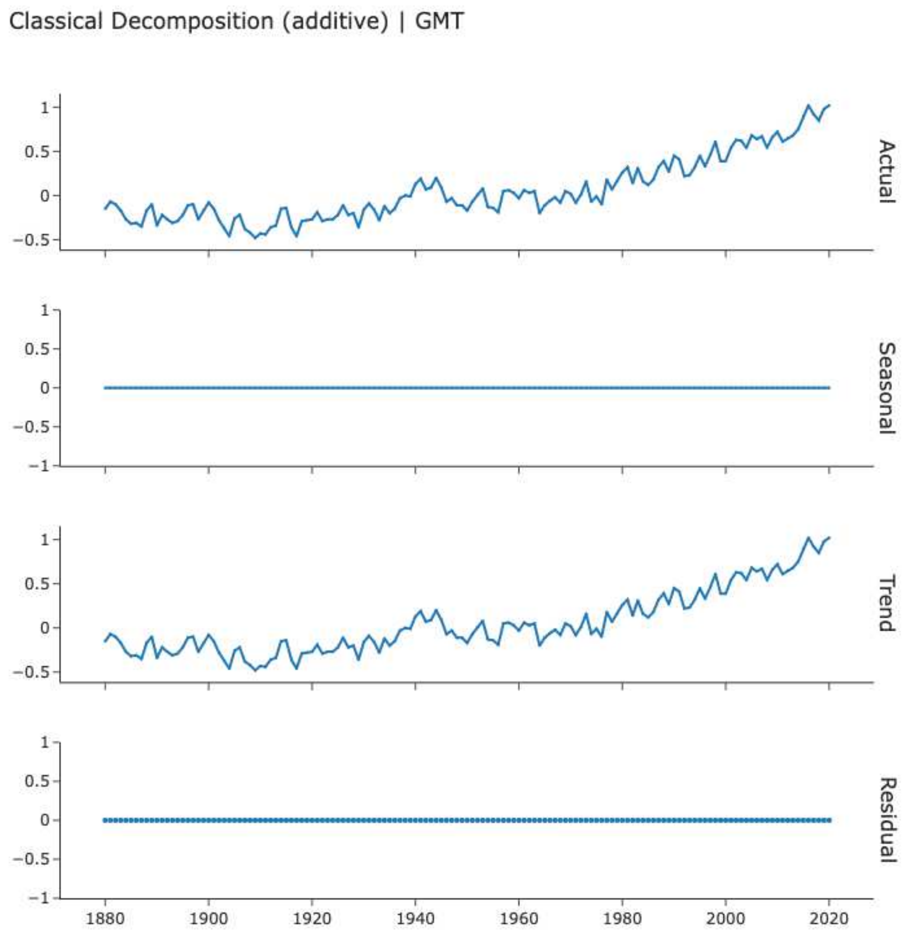

Figure 5: Additive decomposition of GMT time series

It may be noted that additive decomposition is done by using *statsmodels 0.14.0* (*https://www.statsmodels.org/dev/generated/statsmodels.tsa.seasonal.seasonal_decompose.html*) and not to be confused with decomposition done in signal processing (CEEMDAN, EWT and so on).

3.2 Analysis of Outliers

Inter Quartile Range and Isolation Forest methods were used to identify the outliers and the results are shown on Figure 6. GMT for the last 8 years are identified as outliers by both the methods. It is evident that forecasting these by ML methods will be difficult, as they show departure from the dominant pattern of variation of GMT time series.

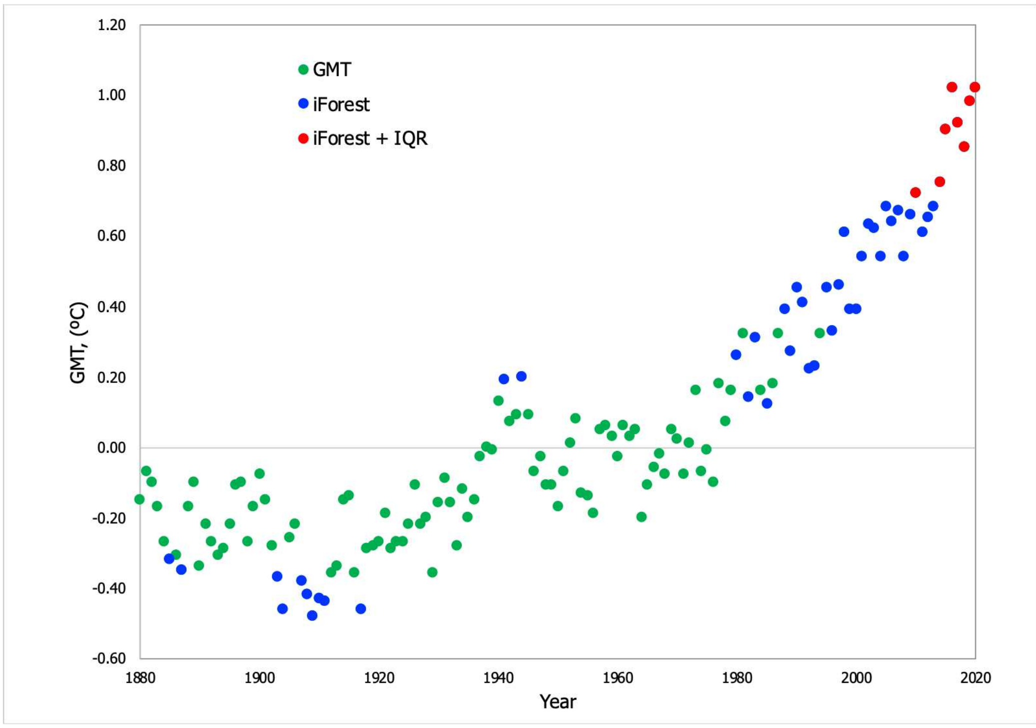

Figure 6: Test for Anomalous Values in GMT Time Series

4 Experiments and Methodology

A time-series predictive problem can be formulated in different ways: (a) Univariate: where the time axis is the independent variable and the target variable alone is the dependent one. The time interval must be uniform and any missing value must be imputed. Conveniently, the time series of GMT does not have missing values. (b) Multivariate with exogenous variables: here other independent variables having the same time axis are combined together with the target variable. (c) Multivariate formulation using the lagged values as one or more parallel series, converting the time series to a regression problem, the lagged series acting as the exogenous variables. The formulations (a) and (c) are selected in the present work where the former is discussed in details, and only the best results for the latter are presented.

Open-source *Python* 3.8 is used as the programming language. To train, tune and predict a large set of models, *PyCaret* (an open-source, low-code machine learning library in Python) is used. *PyCaret* is essentially a Python wrapper overarching several ML libraries and frameworks such as *sktime*, *sklearn*, regressors like *XGBoost*, *LightGBM*, *CatBoost*, as well as connecting to other time-series packages like *pmdarima* (for Auto ARIMA), *statsmodel* (statistical algorithms like ARIMA, ETS) and *Prophet* (Ali [18]). It is a recent development, but gaining popularity in diverse fields. [19,20,21,22]. For the documentation and code examples please refer to *PyCaret's* GitHub site: https://pycaret.org/.

Both the *Time-series* and *Regression* modules of *PyCaret* have been used for implementing formulations (a) and (c) respectively and they are discussed under separate heads with the same names.

A large set of ML algorithms are used in this study (listed in the Table 5 in Appendix) for both time series and regression. As details of these algorithms are available in the books [23,24,25,26]. A few rare algorithms used in *PyCaret*, for example ARD, the discussion can be found on the internet by searching the full names. Hence, details of the algorithms are not discussed here.

4.1 Model Parameters

4.1.1 Time-series

A few parameters are to be defined for carrying out a time series prediction experiment. The GMT data are available from the year 1880 to 2020, comprising a total of 141 records. Defining the test size is mandatory, and test size was varied as 5, 10, and 15 years. The training period gets defined accordingly. Other optional techniques can be used to get a more robust prediction, for example, multiple train and test splitting. Three-fold splitting which results in three sets of training and testing data was used here, and the training size was expended keeping the testing size constant. A schematic diagram (Figure 19) is shown in the Appendix.

4.1.2 Data preparation

Three kinds of data preparations were tried in this work. This is done to make the data more suitable to the data analytic methods. Advantages of each of these three types of transformed are also described below.

1. Differencing: As shown earlier, a first-order differencing made the GMT time series stationary, stationarity of time series data being an implicit assumption of the applicability of many ML algorithms.

2. Power Transform: The Normality test described earlier showed that GMT data is non-Gaussian. Much better performance is obtained if the distribution can be made closer to Gaussian, as algorithms like linear regression and logistic regression explicitly assume the variables have a Gaussian distribution. Although this requirement is not stringent for non-linear algorithms, they often perform better when variables have a Gaussian distribution having less skew. A power transform of the raw data is expected to have the desired effect. Two popular approaches for such power transforms have been tried. They are (a) Box-Cox and (b) Yeo-Johnson Transforms. Details of these are presented in the Appendix.

3. Scaling: Standardization makes the data have zero-mean (when subtracting the mean in the numerator) and unit-variance.[27]

$$y' = \frac{x - \bar{y}}{\sigma}$$

where y is the original feature vector, $\bar{y} = \text{average}(y)$ is the mean of that feature vector, and σ is its standard deviation. Scaling is applied regardless of the usage of differencing and power transforms.

The problem of choosing a transform or a sequence of transforms, for that matter, which makes the data more suitable to the models often defies science, and borders upon art.[28,29] To complicate the matter further, different ML algorithms have different requirements. It is worth mentioning that sometimes the raw data can be found to be more effective than the power transformed data, too. In this study, along with using the raw data, the data preparation methods are applied, to test their efficacies. This is also shown that while data preparation helped for some algorithms, for others it did not have any significant effect. In order to reap the maximum benefit, data preparation is to be considered as a potential way of improving the results. Thus, supported by our calculations, it is recommended that data preparation be carried out.

Not many studies include the train-test split as a parameter affecting the model's skill. Medar et al.[30] showed that by keeping the test size constant when the length of the training data is varied, the skill of the model's prediction also varies. For a larger time series, this effect may not be observed. The GMT time series is quite short and hence train-test split was also considered as a parameter in this study as described above.

A total of six combinations are thus possible for data preparation, two for whether differencing is done or not multiplied by three for power transforms (two power transforms plus one for the absence of them). When the number of folds, fold strategy, tuning iteration and search method are kept constant, as test sizes are varied from 5, 10, and 15, combined with the data preparation strategy, a total of 18 experimental set-ups are possible. The modelling parameters are summarized in Table 2.

Table 2: Parameters in Models

Context	**Parameter**	**Symbol**	**Values**
Data Preparation	Differencing order	D_n	$n = 0, 1$
	Power Transform	PT	$PT = NO$ (No Transform), BC (Box-Cox), YJ (Yeo-Johjnson)
Train-Test Split	Test Size	Tm	$m = 5, 10$, and 15
Scaling	Type of Scaling		Standard
Train/Tune	Fold	F_l	$l = 3$
	Fold Strategy	F_s	s = Expanding
	Tuning Iterations	I_u	$u = 20, 50$
	Search Method	H_v	v = Random

For each of the mentioned 18 combinations in Table 2 all the time series models available in *PyCaret* listed in Table 5 of Appendix were trained, tuned and predictions made. The predicted data are inverse transformed to get back the data in the original scale.

It is customary to define and evaluate a baseline model (Naïve Forecaster in the Table 5). The simplest Naïve model does not do any computation, it simply returns the last observed value. As algorithms, scoring less than Naïve are to be discarded, it was decided to adopt a slightly more skilful type[31] of the Naïve models. It makes variable forecasts which increase or decrease over time, where the amount of change over time (called the "drift") is set to be the average change seen in the historical data. Thus the forecast for time $t+h$ is represented as:

$$\hat{y}_{t+h|t} = y_t + \frac{h}{t-1}\sum_{t=2}^{T}(y_t - y_{t-1}) = y_t + h\left(\frac{y_t - y_1}{t-1}\right)$$

4.1.3 Regression

Machine learning methods that use regression techniques can also be applied to time-series once the latter is converted to a supervised learning problem, i.e., the time series data is manipulated in such a way that we have a set of input variables (independent), used as predictors to predict the dependent variable(s). A window size (or lag) value (W) is defined to achieve this such that $GMT(t-W), GMT(t-W+1), ..., GMT(t-1)$ are used as predictor-variables to predict the dependent variable $GMT(t)$, where t denotes time. Therefore, for the regression problem, W forms another parameter, added to the parameters described in Table 2, taking on integer values 1, 2, 3, ... and so on. The regression algorithms available in *PyCaret* are listed in Table 5 of Appendix. The discussion about the algorithms can be found in the books mentioned before (at the end of section *Experiments and Methodology*.

4.2 Model Evaluation

Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root-mean Squared Error (RMSE), Mean Absolute Percent Error (MAPE), and Normalised Root-mean squared Error (NRMSE) are commonly used. RMSE was used in this work. Thus, the RMSE of annual predictions of GMT during the test period was one metric that was used to evaluate the efficacy of the models. While the annual variation of GMT is important, as mentioned earlier, trends in the variation of GMT averaged over a time interval also have acquired importance as it gives an indication of long-range trends. Decadal mean has been commonly employed in climate literature. However, given the small size of data, only two predictions would be possible if a decade is chosen as the averaging period even for a testing size of 20 years. Hence an averaging period of five years has been chosen in this work. The decadal mean is easier to predict more accurately than annual variation. Choice of a five-year averaging period can give an indication of what was observed about the predictability of decadal mean applies to five-year average as well.

If the test size is m, the actual GMT data can be represented by $y = [y_1, y_2, ..., y_m]$. Given y the model predicts $\hat{y}$:

$$\hat{y} = [\hat{y_1}, \hat{y_2}, ..., \hat{y_m}]$$

RMSE is given by

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}$$

When two mean values, that is observed mean and the mean calculated from the annual prediction data ($\bar{\hat{y}}$)

$$\bar{\hat{y}} = \frac{1}{m}\sum_{i=1}^{m}\hat{y}_i$$

are considered, m becomes unity and the $RMSE(ofMean)$ is given by:

$$RMSE(of\ mean) = abs(\bar{y} - \bar{\hat{y}})$$

5 Results & Discussion

5.1 Time-series

All the results presented here were obtained with the number of folds equal to 3, expanding training window, random hyperparameter search method and tuning iteration of 50. Tables 6 and 7 summarizing all the results are placed in the Appendix, and they show the performances of all the algorithms (thirty and one Naïve) for all the six data preparation methods. It is to be noted that GRM and PAR are excluded because both of them scored very high RMSE values. The parameters are indexed by the choice made for differencing, power transformation, and test sizes. For example, D1-YJ-T10 means first-order differencing is used (D1), power transform is Yeo-Johnson (YJ), and test size was 10 (T10). In the following discussion, a *run* signifies a particular data preparation and test size. As mentioned earlier, the total number of runs is 18 for each of the algorithms considered. The results obtained are discussed as follows. A broad comparison of the effect of data preparation on predictions is presented. It is followed by a comparing RMSE of predictions made by various algorithms. The effect of test size is then discussed. The results of one model are used to study the agreement of a 5-year sliding mean between the predicted and the observed values. The accuracy of this prediction is then compared with that of the annual prediction.

5.1.1 Choice of Algorithms for Discussion

All the thirty time-series algorithms and one Naïve, available in *PyCaret* have been evaluated for all the runs. Since the obtained result set is huge, a few representative algorithms are discussed here. The choices are based on two aspects: (a) simplicity (easy to understand and implement) and (b) representative of different classes of algorithms. For this purpose, a broad classification of algorithms was done comprising of classical statistical method, regression, distance/instance-based, tree-based and ensemble algorithms:

(a) Classical statistical algorithm: This class of algorithms employ statistical methods like auto-regression, moving average, exponential smoothing and so on. ARIMA (ARI), is chosen which integrates the autoregressive and moving average terms and is widely used.

(b) Regression: Simple linear regression (LIN).

(c) Distance/Instance based: K-Nearest Neighbour (KNN).

(d) Tree based: Decision tree (DTR).

(e) Ensemble: Between “bagging” and “boosting” ensemble methods, “boosting” is chosen, where the latter boosts the skill by ensembling weak-learners, such as DTR. AdaBoost (ADA) is chosen to be specific.

5.1.2 Effect of Data Preparation

The algorithms applied to the time series often assume or require that the time series be stationary. As discussed earlier, the GMT time series is not stationary and as can be seen from Figure 4, it was made stationary by first-order differencing the time series. Power transforms help remove skewness of the data which may render a time series more suitable to ML algorithms. The input data to the algorithms is thus a joint effect of differencing and power transforms.

5.1.3 Effect of Differencing

Figure 7 shows the effect of differencing for all the models for a test size of 5.

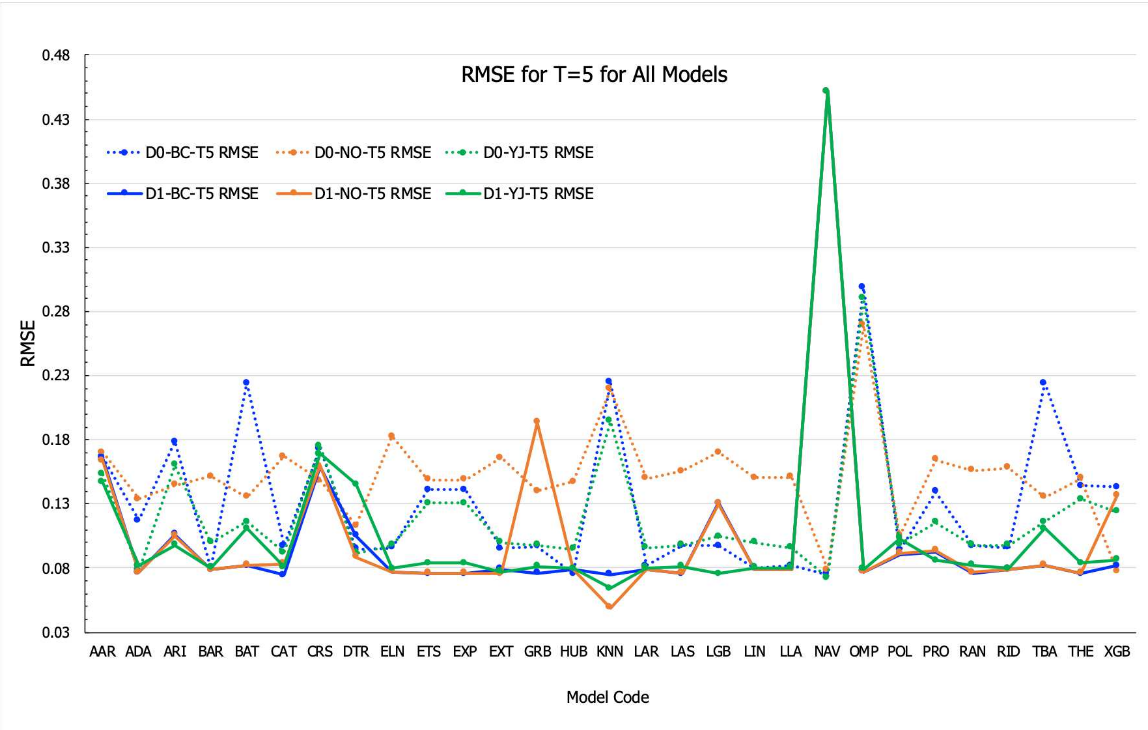

Figure 7: Effect of Data Preparation: All Algorithms, T = 5

It can be seen that by and large the dotted lines representing no differencing (D0) lie above the solid lines representing first-order differencing (D1). Thus, first-order differencing gives better results than no differencing. This behavior was also observed for the vast majority of the algorithms for the other test sizes as well, and this can be seen from the tables of detailed results presented in the Appendix. However, there are exceptions. For example, D0-BC-T5 does far better than D1-BC-T5 for NAV. However, these are few and constitute a minor fraction. It should also be mentioned that for D0-BC-T5, and D0-YJ-T5, NAV has the lowest RMSE of 0.067 among the all available algorithms, a surprising result indeed. Moreover, no other algorithm in the whole set with D0 could reach the lowest RMSE. This also signifies the importance of first-order differencing. This is reinforced by taking KNN as an example and is pictorially represented in Figure 8.

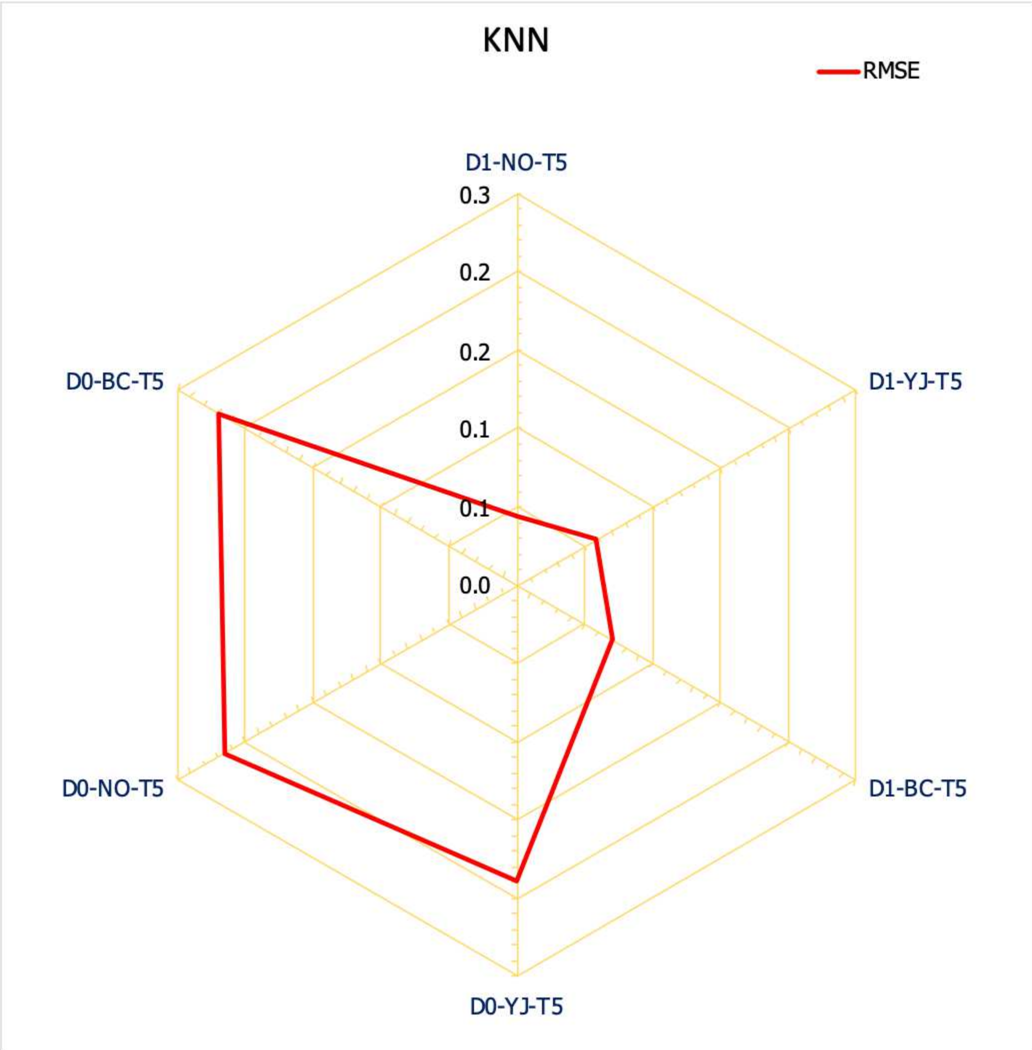

Figure 8: Effect of Data Preparation: KNN

Here, it can be seen that D1 does perform better than D0 when power transform and test size are kept the same.

Therefore, only results where differencing was implemented are discussed and the results where data has not been differenced (D0) are not considered for further discussion.

5.1.4 Effect of Power Transform

Figure 9 compares the effect of no transform or BC or YJ transformed data, and for test sizes of 5, 10, and 15. Results for the five selected models are shown as a bar chart. As mentioned earlier, the comparison is shown only for D1. Each panel begins with red representing ADA and ends with grey representing LIN. Panels corresponding to BC, NO (no transform), and YJ for a constant test size are placed next to each other. The arrangement is repeated for other test sizes. The test size increases from left to right of the figure. No discernible trends are seen. This is because the effect of test size is mixed with the effect of power transform, and the latter is different on different algorithms. To illustrate this, Figure 10 shows the effect of power transform on ARI and KNN. It is seen that for ARI, the order is YJ, NO and BC, whereas for KNN it is NO, YJ and BC.

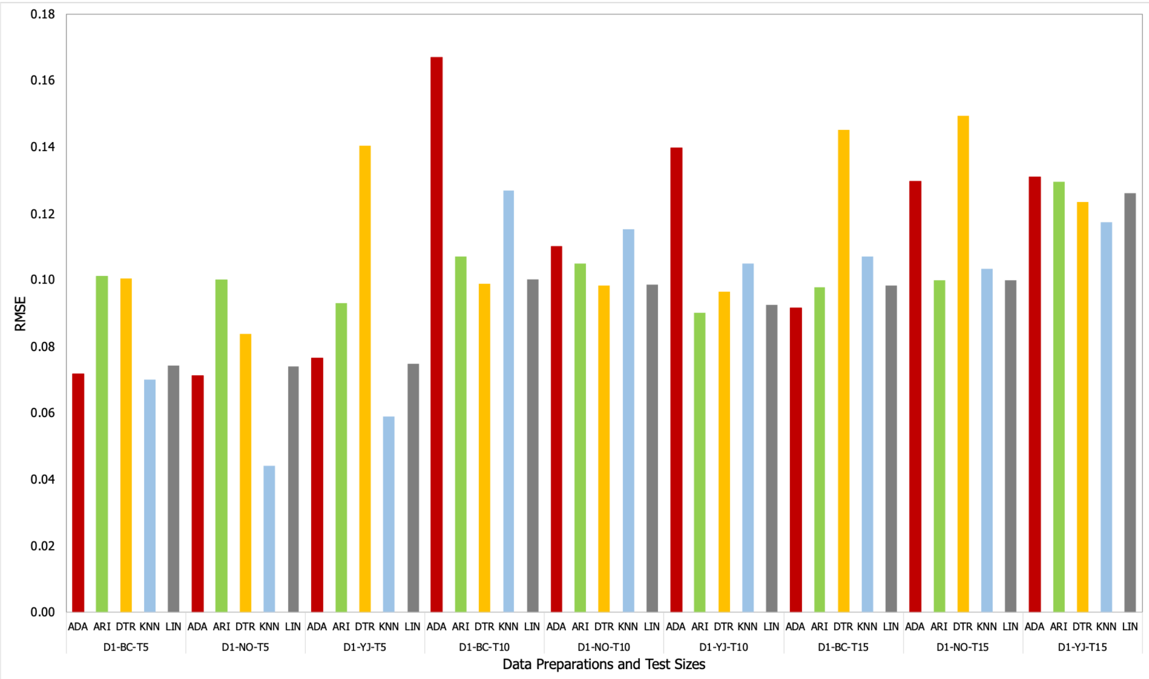

Figure 9: Joint Effect of Differencing, Power Transforms and Test Sizes for 5 Selected Models

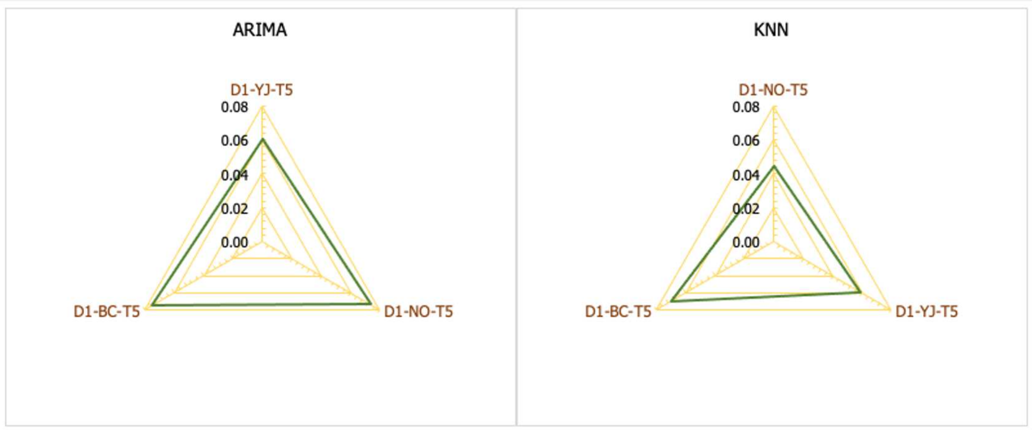

Figure 10: Effect of Power Transform for ARI and KNN (T=5)

5.1.5 Effect of Test Size

An important parameter studied was the train-test size split. Figure 9 shows the effect of test sizes too. The effect of test size can be thought of as twofold. Firstly, the training data available for the algorithm is less as the test size is increased. As the GMT time series is constituted of only 141 values, increasing the test size from 5 to 15 is a significant change in the experimental set-up. Secondly, the primary assumption of all ML algorithms is that the underlying patterns learnt in the training phase are also present in the test data. These two have contradictory effects. Here, no clear trend emerges, however. The test size of 5, and 10 give lower RMSE than 15. This is expected as one anticipates better results as the size of the training set increases. However, the total size of the data set itself is not large, and the train-test split may not come into play significantly. Also, as discussed earlier, the leading eight data were found to be outliers, which complicates the issue further.

5.1.6 Comparison of Predictions vs Observations

Comparison of predictions with observations of GMT for the five representative models chosen for discussion is shown in Figure 11 for the test size of ten (T = 10). *It is to be noted that since the test data has been set aside and not used in the training in any way, the predictions for the test data are virtually equivalent to forecasts, for both the time series methods and regression formulation*. Display in a graph for all the models will be crowded, and a table will be equally unwieldy. However, observed trends and hence comments for the other models are similar to what is presented for the representative models. The observed values of GMT are plotted against the predicted values in Figure 11.

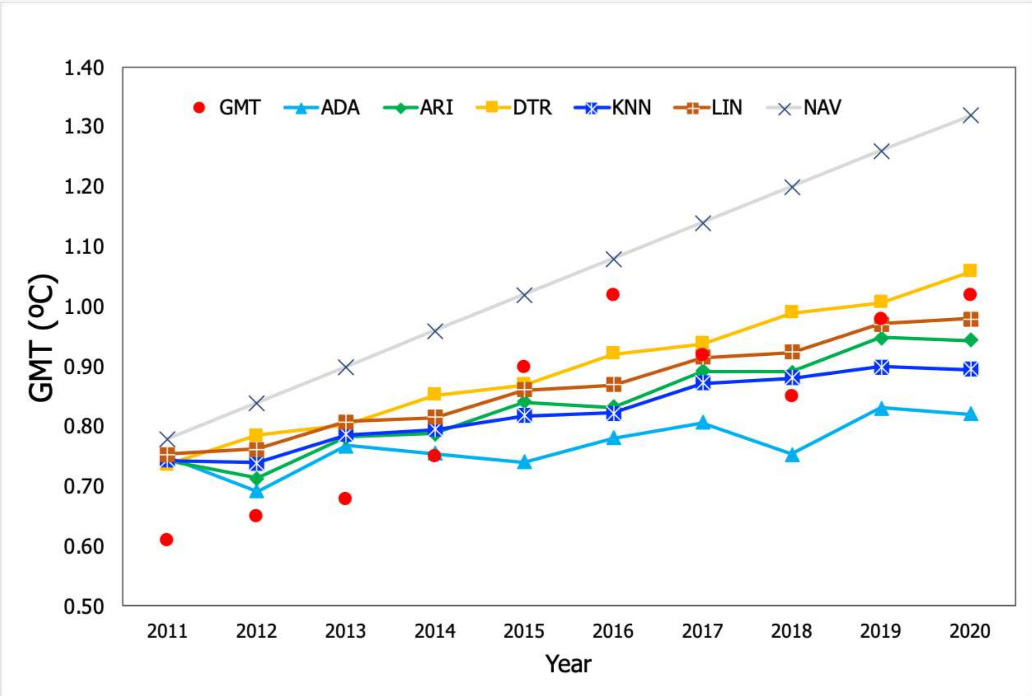

Figure 11: Observed vs Prediction for the Chosen Algorithms and Naïve (T = 10)

The lowest RMSE obtained for ARI is 0.090 in this set.

Figure 12 shows the observed vs prediction for T = 5. For this set, the lowest RMSE (0.044) obtained is by using KNN. This is also the lowest RMSE observed for all the algorithms, all test sizes and data preparation methods.

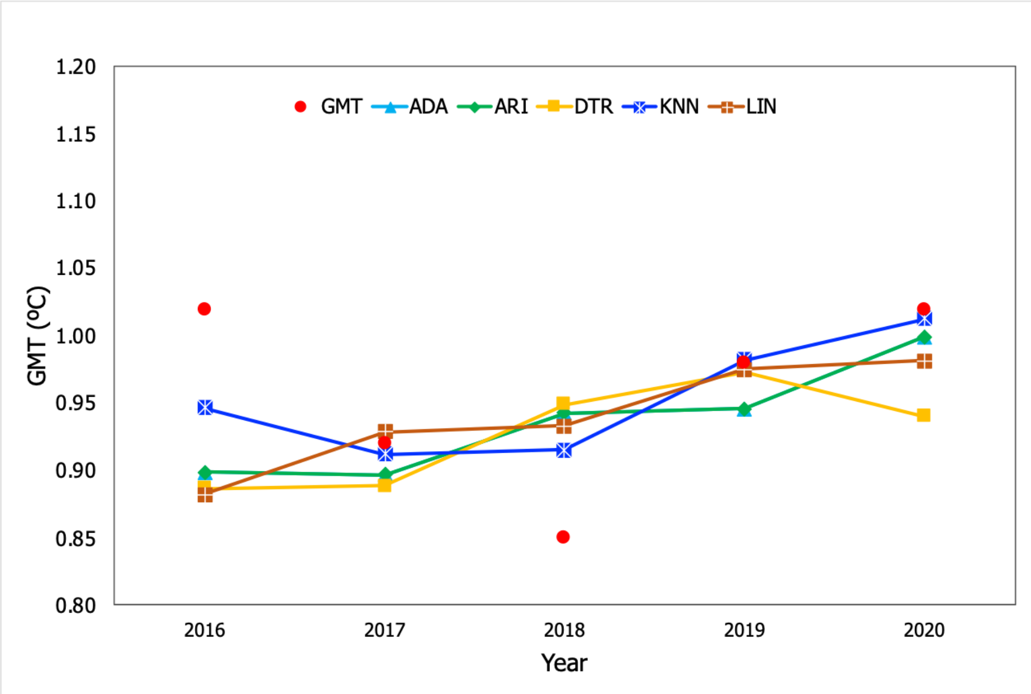

Figure 12: Observed vs Prediction for the Chosen Algorithms (T = 5)

This value is comparable but lower than what is quoted by Brown and Caldeira[9] for predictions by GCM. RMSE values obtained can be considered good since it is comparable to but less than those obtained in GCM models. The results can be considered satisfactory since only the simple ML methods were employed. Further, this may be considered as a benchmark that must be crossed by the more complicated ML models.

5.1.7 RMSE of Averaged Values

The usefulness of GMT data averaged over five-year periods (referred to here as sliding mean) was mentioned earlier. This is demonstrated with a test size of 15 years and with the results of the AdaBoost algorithm (ADA), which was found to be the top performer. The calculation details are as follows. The period is 2006-2020 (15 years). The first mean is one-fifth of the sum of values for 2006 to 2010. The second and the third are for 2011-2015 and 2016-2020. The same applies to computed and observed values. So, three means are calculated from the predicted annual values and another three are actual means calculated from the observed data. RMSE is calculated using these three pairs of mean values. Predictions of the 5-year sliding mean of GMT made by the ADA are compared with observations in Figure 13 for each 5-year block. An RMSE of 0.059 was obtained.

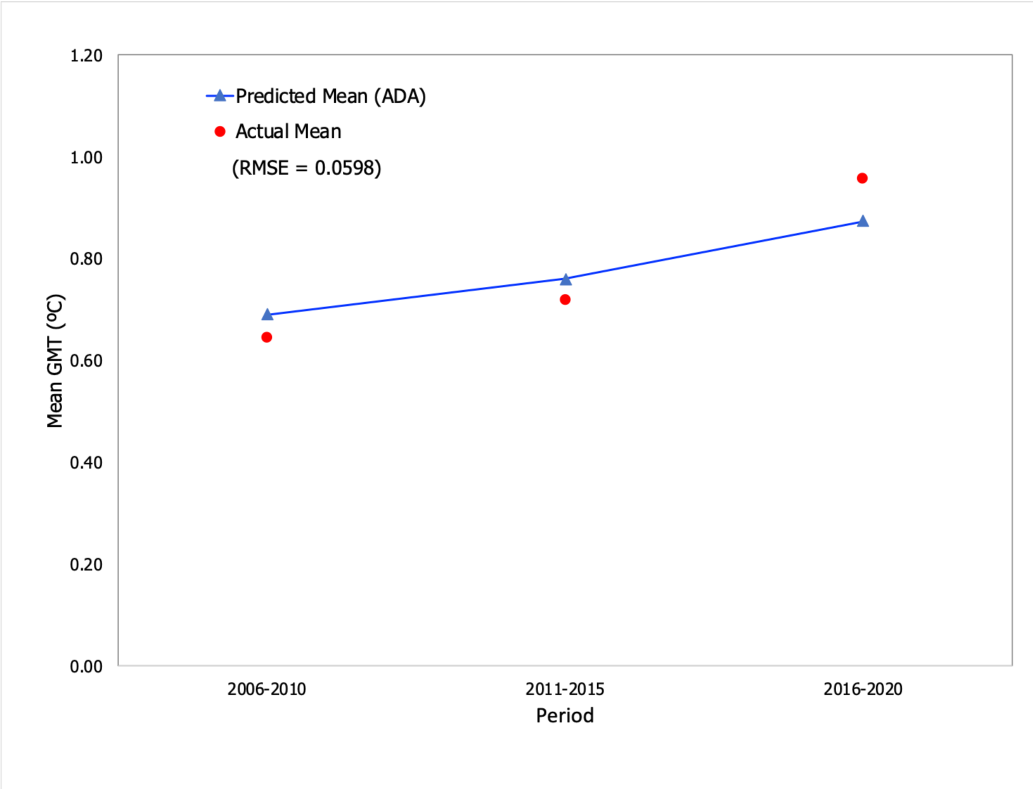

Figure 13: 5-year Sliding Mean Observed vs Predicted

This confirms that values averaged over five-year blocks can be predicted with simple ML methods with great accuracy.

Predictions of the mean against observations also show a similar trend for other test sizes. The comparison is presented for KNN as an example. Figure 14 presents the comparison of observed mean GMT over 5, 10 and 15 years with that of predicted (with RMSE 0.004, 0.029, and 0.023 respectively). Figure 15 shows how the RMSE (of Mean) varies with other algorithms, data preparation and test sizes. It can be seen the RMSE (of Mean) values here are significantly lower than the RMSE of annual variation of GMT.

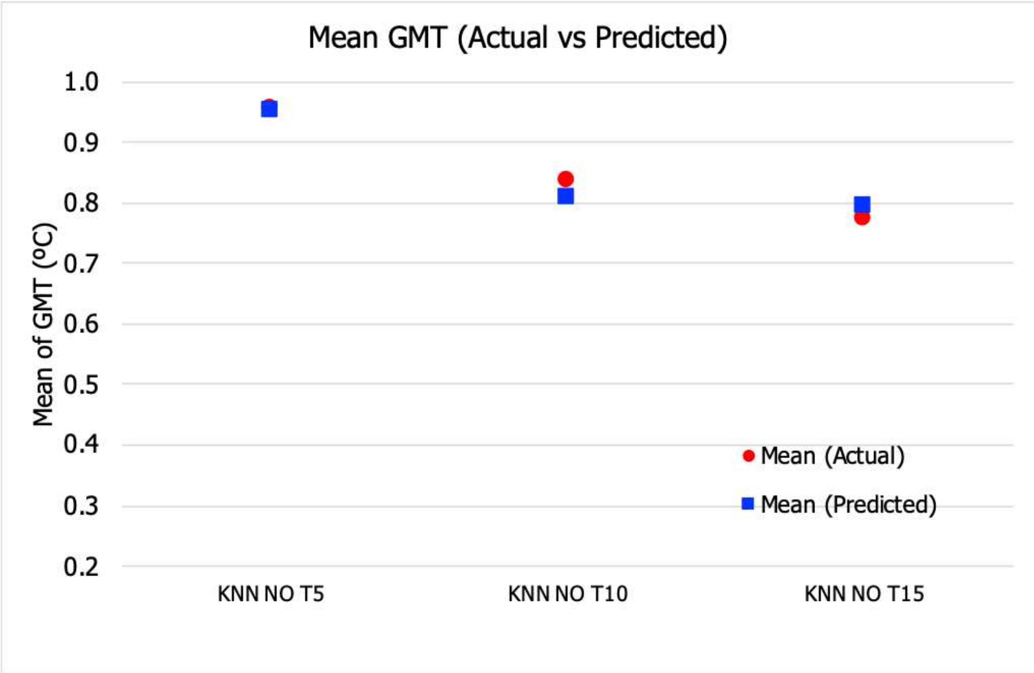

Figure 14: Comparison of 5, 10 and 15 Years Observed and Predicted Means for KNN

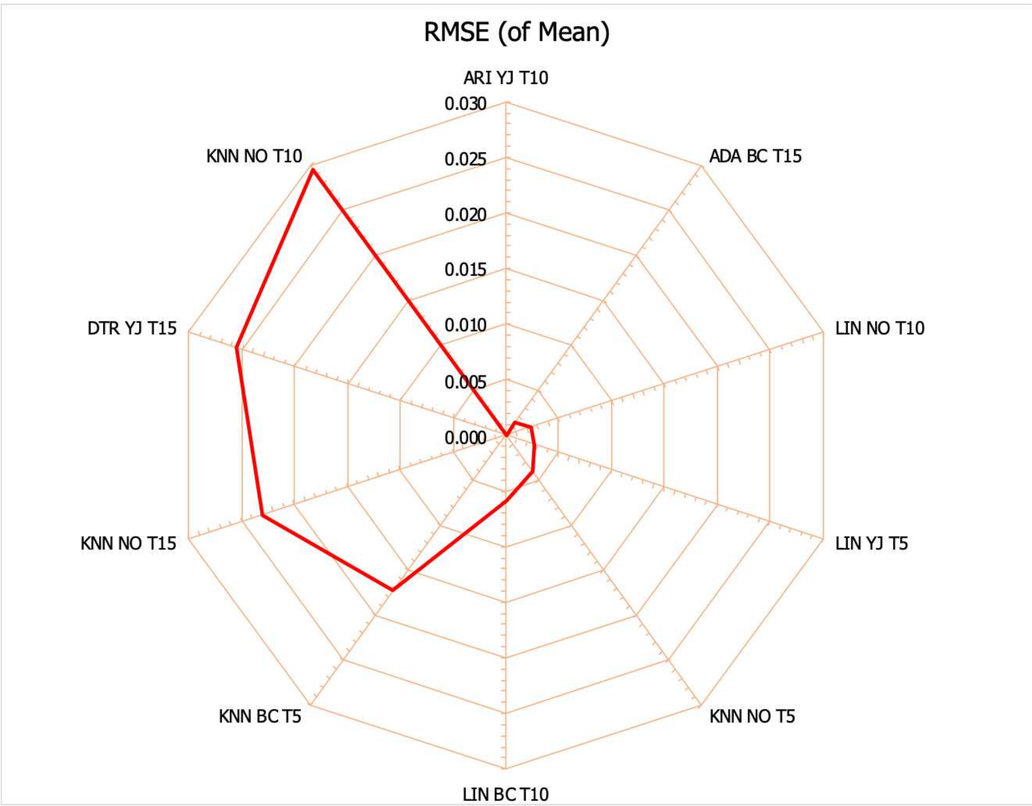

Figure 15: Variation of RMSE (of Mean)

5.1.8 Comparing Models Across All Algorithms All Scenarios

As stated earlier, six combinations of the data preparation varieties form six different experimental set-ups. For each of these experimental set-ups, all the models are trained, tuned and tested. The result is shown in Figure 16. It can only be inferred that the effect of the combination of data preparation, test-train split, and algorithm used produces complex patterns.

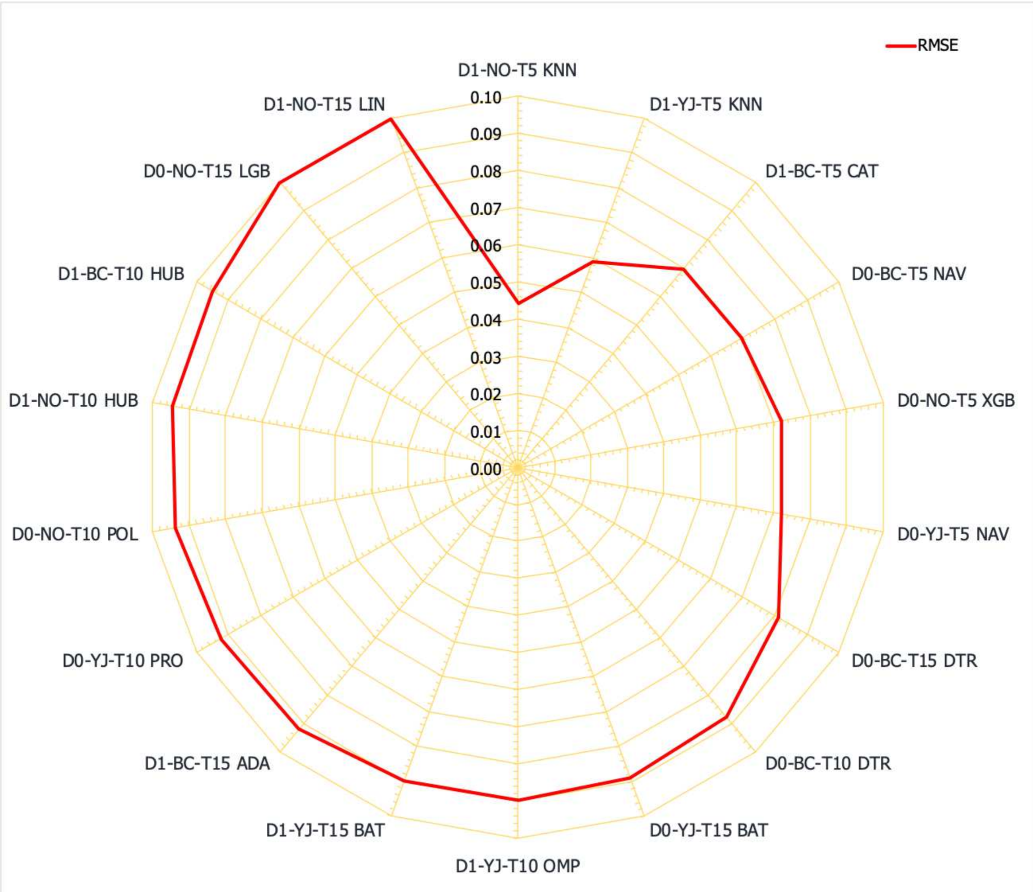

Figure 16: Best Model for Each Run

5.2 Regression

Machine learning methods that use regression techniques can also be applied to time-series once the latter is converted to a supervised learning problem, i.e., the time series data is manipulated in such a way that we have a set of input variables (independent), used as predictors to predict the dependent variable(s). A window size (or lag) value (W) is defined to achieve this such that $GMT(t-W), GMT(t-W+1), ..., GMT(t-1)$ are used as predictor-variables to predict the dependent variable $GMT(t)$, where t denotes time. Therefore, for the regression problem, W forms another parameter, added to the parameters described in Table 2, taking on integer values 1, 2, 3, ... and so on. The regression algorithms available in *PyCaret* are listed in Table 5 of Appendix. Since the time-series methods have been discussed in detail, only the highlights of the results from the regression algorithms are provided.

5.2.1 Effect of Data Preparation

It will be of interest to see the effects of first-order differencing and power transforms on the Regression algorithms as well. A handful of algorithms, out of a total 26 has been selected to demonstrate it. Figure 17 shows the utility of data preparation.

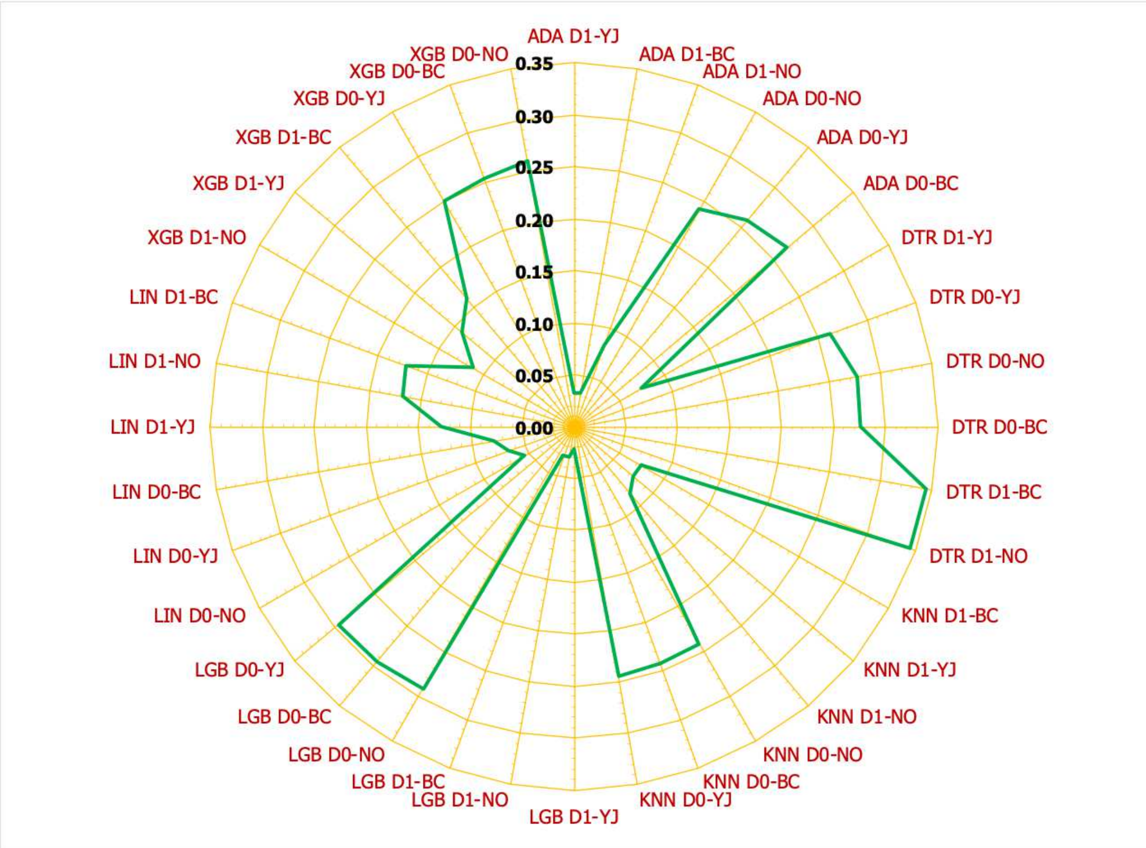

Figure 17: Effects of Data Preparation on the Regression Algorithms

It is seen that, first-order differencing which de-trends the data is beneficial to a large extent for most of the algorithms, barring DTR and LIN. Also, the combination D1-YJ was found to increase the accuracy dramatically for ADA, DTR, KNN, LGB, but the effect is not so pronounced for XGB.

The observed vs predicted GMT for a number of regression algorithms for T = 5 are presented in Figure 18. The lowest RMSE of 0.02 was observed for *LGB*.

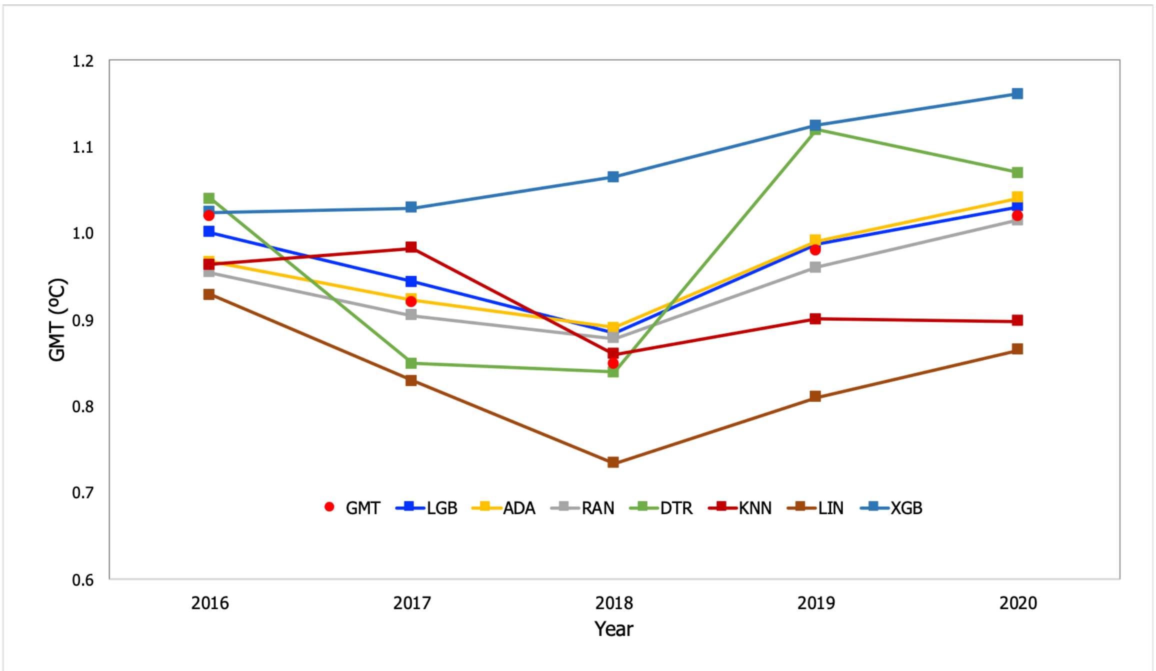

Figure 18: Observed vs Prediction for Regression Algorithms (T = 5)

The Tables 3 and 4 summarise the top RMSE and the top RMSE (of Mean) values obtained for regression algorithms, respectively. Here the key W-29-D1-YJ-T5 represents a window size of 29, first-order differencing, Yeo-Johnson power transform and the test size of 5. It is seen that LGB scored the lowest RMSE and lowest RMSE (of Mean) for test size equal to five.

Table 3: List of Top RMSE Values for Regression Algorithms for Different Data Preparations and Test Sizes

Data Key and Test Size	Model Code	RMSE
W29-D1-YJ-T5	LGB	0.0218
W29-D1-NO-T5	LGB	0.0290
W29-D1-BC-T5	LGB	0.0302
W35-D0-YJ-T10	KRI	0.0642
W38-D0-BC-T10	OMP	0.0716
W35-D0-NO-T10	KRI	0.0718
W32-D0-YJ-T15	BAR	0.0729
W39-D1-NO-T15	XGB	0.0737
W41-D1-BC-T15	XGB	0.0744

It is observed that employing regression ML techniques significantly improved the results from those obtained by using only the time-series methods. Thus, considering both time-series and regression ML methods, the lowest RMSE and the lowest RMSE (of Mean) becomes 0.02 and 0.00002 respectively, which are better results to the knowledge of the authors. The results are good and certainly will form benchmarks for the more complex ML methods to meet.

Table 4: List of Top RMSE (of Mean) Values for Regression Algorithms for Different Data Preparations and Test Sizes

Data Key and Test Size	Model Code	RMSE (of Mean)
W32-D1-YJ-T5	LGB	0.00002
W45-D1-NO-T5	MLP	0.00009
W42-D0-BC-T5	ARD	0.00050
W35-D0-YJ-T10	KRI	0.01389
W38-D0-BC-T10	OMP	0.00284
W35-D0-NO-T10	KRI	0.02009
W44-D1-NO-T15	ELN	0.00017
W30-D0-YJ-T15	ARD	0.00035
W26-D0-BC-T15	TSR	0.00037

6 Conclusion

Accurate prediction of GMT or its mean variability over a span of years is a challenge. Annual GMT data during the period 1880-2020 were analysed by treating them both as a time series and also converting it to a regression problem with a systematic implementation of simple ML methods.

Firstly, the often neglected step of data preparation was implemented. This consisted of differencing, power transforms, and scaling. It was found that first-order differencing made the data stationary and had a positive impact on the results obtained for the time-series and regression ML methods. Secondly, the ML algorithms are selected in an unbiased way, emphasising variety. It was found that for time-series methods, simple algorithms like KNN, HUB and LIN did perform better than the more popular algorithms like XGB, LGB and Prophet, where that is not the case with the regression formulation. This points to the value of an unbiased selection of a wider set of algorithms. Thirdly, the benefits of employing a Naïve model have been demonstrated.

It was seen that the performances of the models vary with both the data preparation strategies and test sizes. It is difficult to explain why one model does well compared to all the others. Performance is mostly the result of complex chemistry between three factors: (i) mode of data preparation, (ii) forecast horizon and (iii) suitability of 1 and 2 for the ML algorithm. While it is true that a fair amount of insight can be obtained for selecting appropriate algorithms from the analysis of data under study, and the knowledge of the algorithm, choosing the right ML algorithm is an iterative process and the success of this depends on the experience of the modellers.

Some important features of annual GMT data could be observed. Though simple ML methods may not be adequate to make a yearly prediction of GMT, they are sufficient to accurately capture the 5-year mean variability.

In future work, as mentioned earlier, the GMT time series will be converted to a multivariate regression problem with additional features generated from the GMT's lagged values or by adding exogenous variables. It is to be investigated whether the regressors used are sufficient to accurately predict GMT. It is planned to use all the successful techniques to forecast GMT into the future.

7 Appendix

7.1 Details of Power transforms

Detailed formulae of Box-Cox and Yeo-Johnson transforms are presented here. (a) Box Cox[32]

The one-parameter Box-Cox transformation is defined as

$$y_i^{(\lambda)} = \begin{cases} \dfrac{y_i^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0, \\ \ln y_i & \text{if } \lambda = 0, \end{cases}$$

It is to be noted, that Box-Cox transformation cannot be applied to a data series that is not strictly > 0. Therefore, the data must be offset by the minimum value in the series and adding a very small number to make all the data greater than zero.

$$y^T(t) = y(t) + abs(y_{Min}) + \varepsilon$$

where y_{Min} is the minimum value of the entire series and ε is a small positive number.

(b) Yeo-Johnson[33]

The Yeo-Johnson transformation allows for zero and negative values of y.

λ can be any real number, where $\lambda = 1$ produces the identity transformation. The transformation law reads

$$y_i^{(\lambda)} = \begin{cases} ((y_i+1)^\lambda - 1)/\lambda & \text{if } \lambda \neq 0, y \geq 0 \\ \log(y_i+1) & \text{if } \lambda = 0, y \geq 0 \\ -((-y_i+1)^{(2-\lambda)} - 1)/(2-\lambda) & \text{if } \lambda \neq 2, y < 0 \\ -\log(-y_i+1) & \text{if } \lambda = 2, y < 0 \end{cases}$$

7.2 Train Cross Validation Split

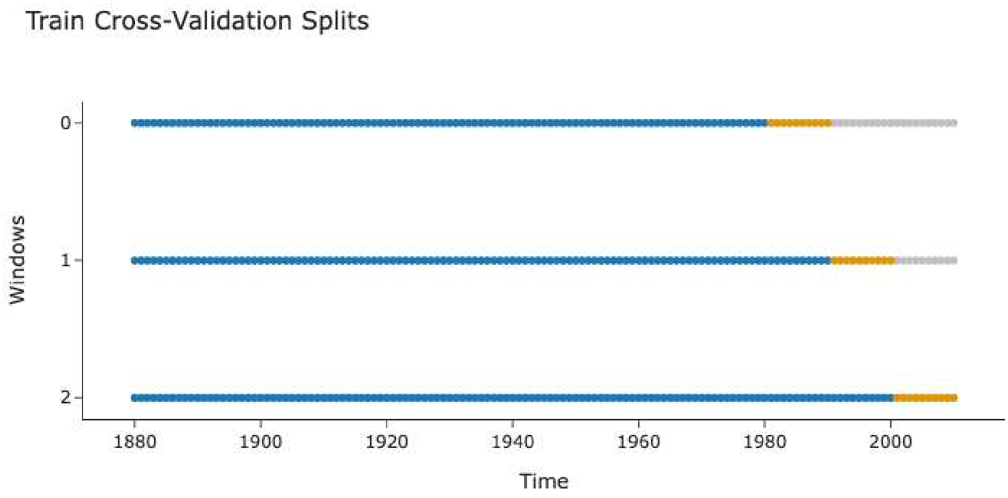

Figure 19: Expanding Training Window with 3 Fold Cross Validation

7.3 Predictive Models Used

The following models listed in Table 5 are considered in this study. The Time series and Regression algorithms are listed under separate heads.

Table 5: Time-series and Regression Models

Time-series		Regression	
Code	**Name**	**Code**	**Name**
AAR	Auto ARIMA	ADA	AdaBoost Regressor
ADA	AdaBoost	ARD	Automatic Relevance Determination
ARI	ARIMA	BAR	Bayesian Ridge
BAR	Bayesian Ridge	CAT	CatBoost Regressor
BAT	BATS	DTR	Decision Tree Regressor
CAT	CatBoost	DUM	Dummy Regressor
CRS	Croston	ELN	Elastic Net
DTR	Decision Tree	EXT	Extra Trees Regressor
ELN	Elastic Net	GRB	Gradient Boosting Regressor
ETS	ETS	HUB	Huber Regressor
EXS	Exponential Smoothing	KNN	K Neighbors Regressor
EXT	Extra Trees	KRI	Kernel Ridge
GBO	Gradient Boosting	LAR	Least Angle Regression
GRM	Grand Means	LAS	Lasso Regression
HUB	Huber	LGB	Light Gradient Boosting Machine
KNN	K Neighbors	LIN	Linear Regression
LEA	Least Angular	LLA	Lasso Least Angle Regression
LGB	Light Gradient Boosting	MLP	MLP Regressor
LIN	Linear	OMP	Orthogonal Matching Pursuit
LLA	Lasso Least Angular	PAR	Passive Aggressive Regressor
LSO	Lasso	RAN	Random Forest Regressor
NAV	Naïve	RID	Ridge Regression
OMP	Orthogonal Matching Pursuit	RSC	Random Sample Consensus
PAG	Passive Aggressive	SVM	Support Vector Regression
POT	Polynomial Trend	TSR	TheilSen Regressor
PRO	Prophet	XGB	Extreme Gradient Boosting
RAN	Random Forest		
RID	Ridge		
TBA	TBATS		
THE	Theta		
XGB	Extreme Gradient Boosting		

7.4 RMSE Values for all the Models where no Differencing is Done (D0)

Table 6: Results for No Differencing

Case: D0	D0-BC			D0-NO			D0-YJ		
Model Code	5	10	15	5	10	15	5	10	15
AAR	0.162	0.181	0.243	0.165	0.191	0.179	0.148	0.159	0.248
ADA	0.112	0.090	0.321	0.129	0.148	0.423	0.073	0.121	0.143
ARI	0.173	0.156	0.125	0.140	0.213	0.139	0.155	0.147	0.113
BAR	0.075	0.093	0.336	0.146	0.151	0.140	0.095	0.147	0.329
BAT	0.219	0.208	0.197	0.130	0.155	0.214	0.111	0.113	**0.089**
CAT	0.092	0.091	0.161	0.162	0.140	0.258	0.087	0.134	0.141
CRS	0.168	0.206	0.208	0.143	0.200	0.182	0.170	0.207	0.209
DTR	0.090	**0.088**	**0.081**	0.108	0.144	0.428	0.087	0.112	0.135
ELN	0.091	0.091	0.351	0.178	0.149	0.140	0.093	0.146	0.346
ETS	0.136	0.167	0.147	0.143	0.181	0.164	0.125	0.153	0.130
EXP	0.136	0.167	0.147	0.143	0.181	0.164	0.125	0.153	0.129
EXT	0.090	0.096	0.094	0.161	0.141	0.135	0.095	0.125	0.135
GRB	0.092	0.091	0.360	0.135	0.182	0.158	0.093	0.136	0.385
HUB	0.070	0.091	0.303	0.142	0.147	0.137	0.089	0.135	0.298
KNN	0.220	0.227	0.225	0.215	0.203	0.440	0.189	0.141	0.152
LAR	0.077	0.093	0.334	0.145	0.151	0.140	0.091	0.140	0.329
LAS	0.092	0.091	0.354	0.150	0.149	0.144	0.093	0.148	0.349
LGB	0.092	0.091	0.131	0.165	0.150	**0.100**	0.100	0.132	0.104
LIN	0.075	0.093	0.334	0.145	0.151	0.140	0.095	0.147	0.327
LLA	0.077	0.093	0.334	0.146	0.151	0.140	0.091	0.140	0.329
NAV	**0.070**	0.141	0.123	0.073	0.154	0.137	**0.067**	0.129	0.111
OMP	0.293	0.340	0.340	0.264	0.363	0.377	0.285	0.331	0.333
POL	0.089	0.090	0.354	0.100	**0.094**	0.143	0.093	0.148	0.348
PRO	0.134	0.107	0.272	0.159	0.119	0.103	0.111	**0.093**	0.222
RAN	0.092	0.091	0.085	0.151	0.149	0.144	0.093	0.148	0.285
RID	0.091	0.091	0.339	0.153	0.151	0.141	0.093	0.147	0.334
TBA	0.219	0.208	0.197	0.130	0.155	0.214	0.111	0.113	**0.089**
THE	0.139	0.181	0.168	0.145	0.196	0.186	0.129	0.170	0.155
XGB	0.138	0.140	0.162	**0.072**	0.106	0.292	0.119	0.115	0.121

7.5 RMSE Values for all the Models where First-Order Differencing is Done (D1)

Table 7: Results for First-Order Differencing

Case: D1	D1-BC			D1-NO			D1-YJ		
Model Code	5	10	15	5	10	15	5	10	15
AAR	0.160	0.181	0.163	0.158	0.177	0.159	0.142	0.150	0.124
ADA	0.072	0.167	**0.092**	0.071	0.110	0.130	0.077	0.140	0.131
ARI	0.101	0.107	0.098	0.100	0.105	0.100	0.093	0.090	0.130
BAR	0.074	0.100	0.100	0.074	0.099	0.102	0.075	0.093	0.131
BAT	0.077	0.122	0.106	0.077	0.135	0.106	0.106	0.102	**0.090**
CAT	**0.069**	0.100	0.137	0.078	0.099	0.111	0.076	0.096	0.152
CRS	0.154	0.218	0.420	0.155	0.220	0.423	0.164	0.238	0.451
DTR	0.100	0.099	0.145	0.084	0.098	0.149	0.140	0.096	0.123
ELN	0.072	0.100	0.103	0.072	0.099	0.105	0.075	0.096	0.135
ETS	0.070	0.099	0.116	0.071	0.097	0.119	0.079	0.092	0.154
EXP	0.070	0.099	0.116	0.071	0.097	0.119	0.079	0.092	0.154
EXT	0.074	0.178	0.136	0.071	0.099	0.101	0.072	0.095	0.124
GRB	0.071	0.169	0.109	0.188	0.098	0.111	0.076	0.096	0.192
HUB	0.074	**0.095**	0.111	0.074	**0.094**	0.113	0.075	0.093	0.126
KNN	0.070	0.127	0.107	**0.044**	0.115	0.103	**0.059**	0.105	0.117
LAR	0.074	0.100	0.100	0.074	0.099	0.102	0.075	0.092	0.130
LAS	0.071	0.100	0.110	0.071	0.099	0.112	0.076	0.096	0.145
LGB	0.126	0.100	0.108	0.125	0.099	0.103	0.070	0.151	0.141
LIN	0.074	0.100	0.098	0.074	0.099	**0.100**	0.075	0.092	0.126
LLA	0.074	0.099	0.100	0.074	0.098	0.102	0.075	0.094	0.130
NAV	0.446	0.227	**1.131**	0.446	0.227	1.131	0.446	0.227	1.131
OMP	0.072	0.101	0.100	0.072	0.099	0.102	0.074	**0.090**	0.130
POL	0.086	0.099	0.116	0.087	0.097	0.119	0.098	0.092	0.154
PRO	0.088	0.147	0.106	0.089	0.145	0.109	0.081	0.100	0.106
RAN	0.071	0.098	0.111	0.071	0.158	0.139	0.077	0.097	0.147
RID	0.074	0.099	0.100	0.074	0.098	0.102	0.075	0.093	0.130
TBA	0.077	0.122	0.106	0.077	0.135	0.106	0.106	0.102	**0.090**
THE	0.070	0.100	0.115	0.071	0.098	0.117	0.079	0.092	0.150
XGB	0.077	0.130	0.307	0.131	0.096	0.157	0.081	0.109	0.182

Declarations

- Acknowledgements: The authors sincerely thank Prof. K.S. Gandhi, Department of Chemical Engineering, Indian Institute of Science, Bangalore, for insightful discussions and critical comments on the manuscript.
- Funding: The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.
- Conflict of interest/Competing interests: The authors have no competing interests to declare that are relevant to the content of this article.
- Ethics approval: The research meets all ethical guidelines, including adherence to the legal requirements of the study country.
- Consent to participate: Not applicable.
- Consent for publication: All authors agreed with the content and that all gave explicit consent to submit and that they obtained consent from the responsible authorities at the Indian Institute of Science, where the work has been carried out, before the work is submitted.
- Availability of data and materials: The Global Mean Temperature data was obtained from NASA's website (https://data.giss.nasa.gov/gistemp/). The datasets generated during and/or analysed during the current study are available from the corresponding author, Dr D. Niyogi, on reasonable request.
- Code availability: Available from the corresponding author, Dr D. Niyogi, upon reasonable request.
- Authors' contributions: Prof. J. Srinivasan contributed to the study conception. Material preparation, data collection and analysis were performed by Dr Debdarsan Niyogi. The first draft of the manuscript was written by Dr Debdarsan Niyogi and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

References

[1] S. Makridakis, E. Spiliotis, and V. Assimakopoulos. Statistical and machine learning forecasting methods: Concerns and ways forward. *PLOS ONE*, 13(3):e0194889, March 2018. doi: 10.1371/journal.pone.0194889. URL https://doi.org/10.1371/journal.pone.0194889.

[2] Sven F. Crone, Michèle Hibon, and Konstantinos Nikolopoulos. Advances in forecasting with neural networks? empirical evidence from the NN3 competition on time series prediction. *International Journal of Forecasting*, 27(3):635–660, July 2011. doi: 10.1016/j.ijforecast.2011.04.001. URL https://doi.org/10.1016/j.ijforecast.2011.04.001.

[3] Claus Weihs and Katja Ickstadt. Data science: the impact of statistics. *International Journal of Data Science and Analytics*, 6(3):189–194, 2018.

[4] Torbjørn Lorentzen. A statistical analysis of sea temperature data. *Theoretical and Applied Climatology*, 119(3-4):585–610, mar 2014. doi: 10.1007/s00704-014-1119-x. URL https://doi.org/10.1007%2Fs00704-014-1119-x.

[5] Provash Mali. Multifractal characterization of global temperature anomalies. *Theoretical and Applied Climatology*, 121(3-4):641–648, aug 2014. doi: 10.1007/s00704-014-1268-y. URL https://doi.org/10.1007%2Fs00704-014-1268-y.

[6] Carlos Batista Silva, Maria Elisa Siqueira Silva, Nísia Krusche, Tércio Ambrizzi, Nelson de Jesus Ferreira, and Pedro Leite da Silva Dias. The analysis of global surface temperature wavelets from 1884 to 2014. *Theoretical and Applied Climatology*, 136 (3-4):1435–1451, jul 2018. doi: 10.1007/s00704-018-2569-3. URL https://doi.org/10.1007%2Fs00704-018-2569-3.

[7] Charles D. Coleman. Improved complete ensemble empirical mode decompositions with adaptive noise of global, hemispherical and tropical temperature anomalies, 1850–2021. *Theoretical and Applied Climatology*, jul 2022. doi: 10.1007/s00704-022-04064-x. URL https://doi.org/10.1007%2Fs00704-022-04064-x.

[8] Konrad Mulrennan, Mohamed Awad, John Donovan, Russell Macpherson, and David Tormey. Modelling the electrical energy profile of a batch manufacturing pharmaceutical facility. *International Journal of Data Science and Analytics*, 10(3):285–300, 2020.

[9] Patrick T. Brown and Ken Caldeira. Empirical prediction of short-term annual global temperature variability. *Earth and Space Science*, 7(6), June 2020. doi: 10.1029/2020ea001116. URL https://doi.org/10.1029/2020ea001116.

[10] Hatice Citakoglu. Comparison of multiple learning artificial intelligence models for estimation of long-term monthly temperatures in turkey. *Arabian Journal of*

Geosciences, 14(20), October 2021. doi: 10.1007/s12517-021-08484-3. URL https://doi.org/10.1007/s12517-021-08484-3.

[11] Anindita Panda, Deepanshu, Ashwani Varshney, and Vibha Gaur. Forecasting temperature anomalies of planet earth: A comparative analysis of AI models. In *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*. IEEE, September 2021. doi: 10.1109/icrito51393.2021.9596567. URL https://doi.org/10.1109/icrito51393.2021.9596567.

[12] Flavio M. Viola, Susana L.D. Paiva, and Marcelo A. Savi. Analysis of the global warming dynamics from temperature time series. *Ecological Modelling*, 221(16): 1964–1978, August 2010. doi: 10.1016/j.ecolmodel.2010.05.001. URL https://doi.org/10.1016/j.ecolmodel.2010.05.001.

[13] Peter Romilly. Time series modelling of global mean temperature for managerial decision-making. *Journal of Environmental Management*, 76(1):61–70, July 2005. doi: 10.1016/j.jenvman.2005.01.008. URL https://doi.org/10.1016/j.jenvman.2005.01.008.

[14] Liming Ye, Guixia Yang, Eric Van Ranst, and Huajun Tang. Time-series modeling and prediction of global monthly absolute temperature for environmental decision making. *Advances in Atmospheric Sciences*, 30(2):382–396, February 2013. doi: 10.1007/s00376-012-1252-3. URL https://doi.org/10.1007/s00376-012-1252-3.

[15] Himika, Shubhdeep Kaur, and Sukhchandan Randhawa. Global land temperature prediction by machine learning combo approach. In *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. IEEE, July 2018. doi: 10.1109/icccnt.2018.8494173. URL https://doi.org/10.1109/icccnt.2018.8494173.

[16] J David Neelin. *Climate change and climate modeling*. Cambridge University Press, 2010.

[17] Stanisław Węglarczyk. Kernel density estimation and its application. *ITM Web of Conferences*, 23:00037, 2018. doi: 10.1051/itmconf/20182300037. URL https://doi.org/10.1051/itmconf/20182300037.

[18] M Ali. Pycaret: An open source, low-code machine learning library in python, pycaret version 2.2 (april 2020). *URL https://www. pycaret. org*.

[19] Ashok Munjal, R Khandia, and B Gautam. A machine learning approach for selection of polycystic ovarian syndrome (pcos) attributes and comparing different classifier performance with the help of weka and pycaret. *Int J Sci Res*, pages 59–63, 2020.

[20] Faisal Bin Iqbal, Srijon Biswas, Rubabatul Urba, et al. *Performance analysis of intrusion detection systems using the PyCaret machine learning library on the UNSW-NB15 dataset*. PhD thesis, Brac University, 2021.

[21] Ghader Manafiazar, Mohammad Riazi, John A Basarab, Changxi Li, Paul Stothard, and Graham Plastow. Psxv-29 late-breaking: Investigating use of machine learning algorithms to predict days in herd for commercial beef cattle. *Journal of Animal Science*, 99(Supplement_3):381–381, 2021.

[22] Paola Ortelli, Davide Ferrazzoli, Viviana Versace, Veronica Cian, Marianna Zarucchi, Anna Gusmeroli, Margherita Canesi, Giuseppe Frazzitta, Daniele Volpe, Lucia Ricciardi, et al. Optimization of cognitive assessment in parkinsonisms by applying artificial intelligence to a comprehensive screening test. *npj Parkinson's Disease*, 8 (1):1–9, 2022.

[23] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning - From Theory to Algorithms.* Cambridge University Press, 2014. ISBN 978-1-10-705713-5.

[24] Peter Flach. *Machine learning: the art and science of algorithms that make sense of data*. Cambridge university press, 2012.

[25] Gavin Hackeling. *Mastering Machine Learning with scikit-learn*. Packt Publishing Ltd, 2017.

[26] Jason Brownlee. Master machine learning algorithms: Discover how they work and implement them from scratch, 2016. *URL https://books. google. ca/books*.

[27] J. Han, J. Pei, and M. Kamber. *Data Mining: Concepts and Techniques*. The Morgan Kaufmann Series in Data Management Systems. Elsevier Science, 2011. ISBN 9780123814807. URL `https://books.google.co.in/books?id=pQws07tdpjoC`.

[28] J. Brownlee. *Data Preparation for Machine Learning: Data Cleaning, Feature Selection, and Data Transforms in Python*. Machine Learning Mastery, 2020. URL `https://books.google.co.in/books?id=uAPuDwAAQBAJ`.

[29] M. Kuhn and K. Johnson. *Feature Engineering and Selection: A Practical Approach for Predictive Models*. Chapman & Hall/CRC Data Science Series. CRC Press, 2019. ISBN 9781351609463. URL `https://books.google.co.in/books?id=q5alDwAAQBAJ`.

[30] Ramesh Medar, Vijay S. Rajpurohit, and B. Rashmi. Impact of training and testing data splits on accuracy of time series forecasting in machine learning. In *2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA)*. IEEE, August 2017. doi: 10.1109/iccubea.2017.8463779. URL `https://doi.org/10.1109/iccubea.2017.8463779`.

[31] Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

[32] Özgür Asar, Ozlem Ilk, and Osman Dag. Estimating box-cox power transformation parameter via goodness-of-fit tests. *Communications in Statistics - Simulation and Computation*, 46(1):91–105, December 2014. doi: 10.1080/03610918.2014.957839. URL `https://doi.org/10.1080/03610918.2014.957839`.

[33] I.-K. Yeo. A new family of power transformations to improve normality or symmetry. *Biometrika*, 87(4):954–959, December 2000. doi: 10.1093/biomet/87.4.954. URL `https://doi.org/10.1093/biomet/87.4.954`.