Support Vector Machine Based an Efficient and Accurate Seasonal Weather Forecasting Approach with Minimal Data Quantities

S. Chandrasekara, S. Tennekoon, N.Abhayasinghe, and L. Seneviratne
Department of Electrical and Electronic Engineering
Sri Lanka Institute of Information Technology
Colombo, Sri Lanka
{[1]cmsmchan, [2]shentennekoon}@gmail.com, {[3]nimsiri.a, [4]lasantha.s}@sliit.lk

ABSTRACT

Climate change makes a big impact in our daily activities. Therefore, forecasting climate changes prior to its actual occurrences is important. Even though highly accurate weather prediction systems throughout the world are available, they require mass amounts of data exceeding thousands of data points to obtain a significant accuracy. This study was aimed at proposing a Support Vector Machine based approach to carryout seasonal weather predictions up to thirty-minute intervals, the results of which would be considerably effective with respect to predictions carried out with models trained with annual datasets. The model was trained utilizing a dataset corresponding to the district of Kandy which consisted of 136 samples, 20 features, and 5 labels. By means of carrying out numerous data preprocessing steps, the model was trained, and the relevant hyperparameters were optimized considering the grid search algorithm to yield a maximum accuracy of 86%, once tested via the k-fold cross validation. The performance of the Support Vector Machine was also then compared for the same dataset with that of the K-Nearest Neighbor algorithm which consumed relatively fewer computing resources. An optimal accuracy of 61% was observed for this model for a K-value of 27. This approach supported the concept of a Support Vector Machine's ability to perceive time series forecasts to a relatively higher degree and its ability to perform effectively in higher dimensional datasets with smaller number of samples. As per the future work, the Receiver Operating Characteristic analysis is proposed to be carried out to evaluate the performance of the model and the dataset size is proposed to be further enhanced to a maximum of a thousand samples to yield the best performance results.

KEYWORDS: Support Vector Machines, Principal Component Analysis, Receiver Operating Characteristic, Machine Learning, Weather Forecast, Hyperparameter Optimization

1 INTRODUCTION

It is not uncommon to need over thousands of data samples to train a weather prediction model to achieve an acceptable accuracy. To obtain a sufficiently large amount of data dating back to perhaps multiple decades to appropriately train a suitable model would be a highly taxing procedure. There are many sources available on the internet where one would be able to obtain a large enough dataset, however, at the expense of a considerable large sum of money. Upon carrying out a local survey, to carry out sufficiently accurate enough weather predictions, one would need to implement numerous weather nodes, each having a considerably large budget of around LKR 50 000. To obtain a sufficiently large dataset, these nodes would also have to be maintained throughout a sufficiently long enough time, which in turn will demand further financial liabilities. Either a better suited model or a better suited approach to solving the problem of optimal weather prediction accuracies can prove as a solution which would account for hence said liabilities.

In general, prediction models can yield inadequate results due to reasons ranging from inadequately taken measurements, insufficient understanding of the atmospheric phenomena, and the use of non-standard data acquisition equipment. One major factor which was also contribute to the overall performance is the selection of the most appropriate model. If these steps are not carried out with care, the results would most likely be unfavorable. Two statistical models namely the Bayesian and the Frequentist's statistics and the eligibility of each statistical model were introduced. Various steps that were to be considered to minimize the uncertainty in the predictions made and how the uncertainties

ISSN 2961-5011 | https://doi.org/10.54389/UPKF6369

present were to be effectively communed to the end users utilizing probabilities and more precise scales of likelihood via words such as highly likely, likely, unlikely, and highly unlikely were thoroughly elaborated.

According to (Isabel, 2021) and (Andrew, 2011), Support Vector Machines (SVM) perform considerably well when exposed to time series-based forecasting methods. When considering the finding from (Isabel, 2021), it can be noted how SVMs performs considerably well in the presence of higher dimensional datasets with limited number of samples. As most recognized and as observed from the results of the Principal Component Analysis (PCA) as illustrated in results, weather prediction is highly dependent of the time of year and the time of the date under consideration. The dataset which was available was not of great length either. Hence, a SVM was the algorithm expected to give the highest accuracies when considering the restrictions at hand.

According to a comparison carried out between the performance of a novel lightweight data-driven weather forecasting model by exploring temporal modelling approaches of LSTM and temporal CNNs with existing classical machine learning approaches by (Hewage, Trovati, Pereira, and Behera, 2020), deep learning-based models such as temporal convolutional neural networks which can take a sequence of any length and map it to an output sequence of the same length outperformed the WRF up to twelve hours for ten surface parameters. Though deep learning-based approaches yielded considerably accurate results, this was not considered a viable approach to result considerably promising results. The main reason behind this was the absence of a significant enough amount data that would be required to train the neural network. Like (Hewage, 2020), (Behera, Kumari, and Kumar, 2020) also proposes a deep learning model which was utilized to obtain high prediction accuracies. Due to the high data requirements of the deep learning models, and since the main purpose of this application was to carry out optimal predictions with minimal data quantities, this approach was neglected in this specific application for weather prediction.

1.1 Computational Model Selection and Implementation

Two models were considered and implemented to carry out the weather predictions. Then the performance of these two algorithms were then compared to observed which yielded the better accuracy for the dataset of consideration.

Due to the K-Nearest Neighbor (KNN) algorithm's simplistic nature in terms of comprehension, implementation, and parameter optimization, it was initially selected for the task of carrying out predictions. The whole algorithm was based around performing a lightweight calculation of Euclidean distances between the training data samples and the testing data sample which resulted in the processing capabilities required to deploy the algorithm in an embedded device highly viable.

SVMs are known to perform in higher dimensional spaces of datasets with considerably lower number of features. The initial dataset which yielded considerably lower accuracies consisted of 62 dimensions and only 19 samples. This nature of SVMs was expected to prevail when carrying out the predictions. Above all, it was the SVMs history of performing exceptionally well when exposed to time series forecasts that won its place in the selection process.

1.2 Diagnosing and Optimizing the Model

If a machine learning algorithm makes unacceptably large errors, one of four steps can be considered to improve its performance according to (Andrew, 2011). Getting more training samples, trying a smaller set of features, trying additional features, and varying the contributing parameters.

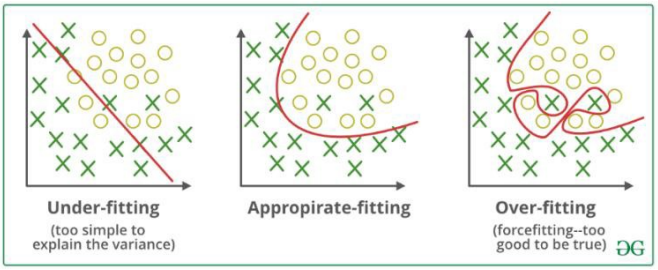

Figure 1. Illustration of three common faults in a model
Adapted from: (Nautiyal, 2021)

Figure 1 above illustrates the three types of behaviors most observed in unoptimized machine learning algorithms. If a certain machine learning model is performing considerably well on the training set but performing noticeably bad on the test set, it can be concluded that the model is overfitting on the training dataset. To fix such a scenario, reducing the number of features available in the dataset may increase the overall performance of the model. If a certain model is neither performing exceptionally well on the training nor the test dataset, the reason maybe high bias. This may be overcome by collecting more samples for the data frame. Using additional features can also help solve this issue.

The approach most used to train a suitable weather prediction model consists of gathering an annual dataset and training a single model to carry out predictions for the span of the whole year. This prompts the need for a larger dataset to yield better generalized results, the generation of which can be both time and resource consuming. Hence it was planned to carry out the weather predictions considering a shorter time span to reduce the number of data points needed for a better generalized prediction. The approach to train the model using a dataset consisting of samples from the two months June and July is proposed to train a model to make predictions for the exact same two months. SVM in addition to these factors will have more parameters which contribute to the overall performance. By varying these the trends can be observed and hence the errors can be rectified. For this purpose of evaluating hypothesis, the dataset can be divided into the most basic form, two parts known as the training set and the test set. This convention requires that around 70% of the dataset be allocated for the training set and the remaining 30% for the test set. However due to the smaller dataset that was considered for this implementation, for the model to be properly trained, the training set was allocated 90% of the data and the test set around10% of the data.

2 METHODOLOGIES

Weather data collected was observed to have considerable number of anomalies and deviations. Due to this fact the following data preprocessing steps were carried out.

2.1. Cleaning, and Preprocessing the Dataset

Since legal means of web scrapping from any of the relevant online resources were not available, the dataset was collected manually and, due to this fact, it consisted of a considerable number of deficiencies. To ensure that none of these missing values were processed as features, certain variables were dropped from the dataset.

2.2. Dimensionality Reduction

Due to the high variance observed for the higher number of features used, it was required to reduce the number of features of the dataset. For this Principal Component Analysis was considered. By calculating the percentage of variation each principal component accounts for, the relevant scree plot was generated as illustrated in Figure 6.

By considering the Kaiser-Guttman Criterion, the maximum number of feasible linear combinations to be extracted from the dataset was considered. According to the Kaiser-Guttman Criterion, any principal with variance less than one, contains less information than one of the original variables and

hence is considered not worth retaining. It was noted that the first three components were in accordance with the nature of the variance required according to (Steiger, 2015).

2.3. Accounting for Missing Data Values

To observe the presence of any missing data values, the unique samples available for each feature available were observed utilizing the *unique()* function available in the python environment. By observing the output, the missing data values were located and accounted for considering one of two suitable methods elaborated below.

a) Dropping the samples consisting of missing data values

The entire row features which contain a missing value would be dropped from the dataset. As a result, a smaller dataset would be resulted.

b) Imputing the missing values

The data values which are missing would be replaced considering a suitable value. This value derived however may not entirely be the best estimate to that missing data sample and hence may affect the accuracy of the predictions carried out.

Usually, weather predictions are made considering over thousands of data samples. Since this level of accessibility was not available for the optimization process, dropping samples and hence reducing the size of the dataset was not considered. Hence, imputation was performed to account for the missing data values.

2.4. One Hot-Encoding

Majority of the machine learning models are incapable of working with categorical data directly. It was required to encode each of the available categorical data type in each categorical data feature into a binary number format.

Wind direction was the only categorical data required to be encoded in the new dataset generated. For each state observed in the wind direction feature, a new column feature is created. Then the presence of each state is considered and assigned a one in the new feature for samples in which the specific state is observed to be present. A zero is assigned to the newly created feature in all other samples where this specific state is not observed. After the creation of these new columns, the single column which accounted for the wind direction is dropped.

2.5. Feature Normalization

To shift the scale of values of the data given in a data column to a range in between 0 and 1, feature normalization was performed considering Eq. (1).

$$x(normalized) = \left(\frac{x - min(x)}{max(x) - \min(x)}\right) \quad (1)$$

2.6. Detecting and Eliminating the Outliers Utilizing box plots

After the one hot-encoding and normalization procedures were performed, boxplots were utilized to observe the presence of anomalies among the samples in a feature of focus. Once the anomalies were visually detected utilizing box plots, it is made possible to account for this anomaly and hence prevent them from effecting the overall performance of the model by expanding the ability of the model to generalize to a more satisfactory degree. There were two methods which were considered for handling outliers according to (Vidhya, 2021). Removing the outliers and replacing outliers with a suitable value Out of these two solutions, the later method was considered for implementation.

2.7. Andrew's plots

Variations in the hence processed data frame was observed by means of utilizing various plots such as Andrew's plots. In 1972, Andrew's suggested the idea of representing multivariate data by means of coding. Each multivariate observation can be transformed to a curve such that the coefficients of the Fourier Series is represented by the observations. The outliers appear as single Andrew's curves which are different from the rest. By utilizing these plots, it was hence made possible to visualize the degree to which the rectification of the outliers in the data set were carried out as illustrated by (Jaadi, 2021).

3 EXPERIMENTAL EVALUATIONS

3.1. Rectification of Outliers

Detection and the rectification of outliers is considered one of the most crucial steps in the procedure of data preprocessing. If the outliers in the dataset were not accounted for, the ability of the model to yield well generalized results to highly variant readings may not be feasible. The final data frame considered consisted of twenty dimensions. For each of these dimensions, box plots were generated to gauge the presence or the absence of outliers. (Dawson, 2011)

Outliers could have been accounted for by means of either removing the samples specific outliers, or by means of replacing the outliers with the closest quantile. By considering the later means, outlier values will be rounded up to or down to the nearest quantile value as observed from the box plots.

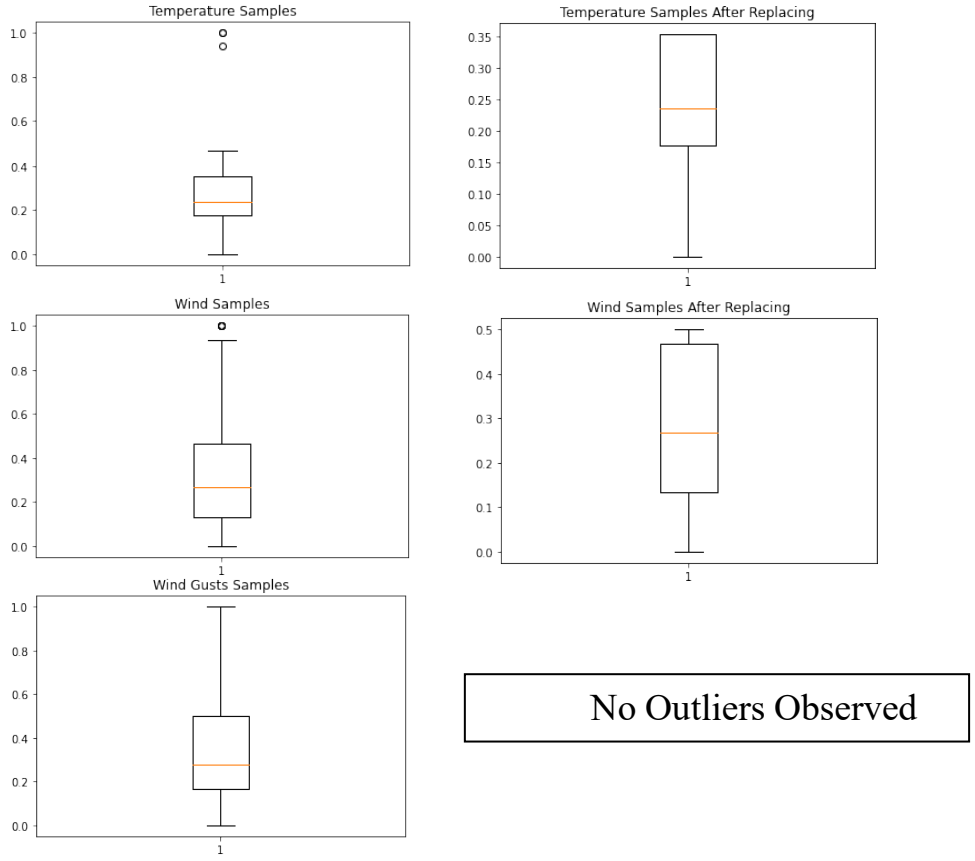

Figure 2. Box plots generated to visualize the presence of outliers in the dataset generated for Kandy

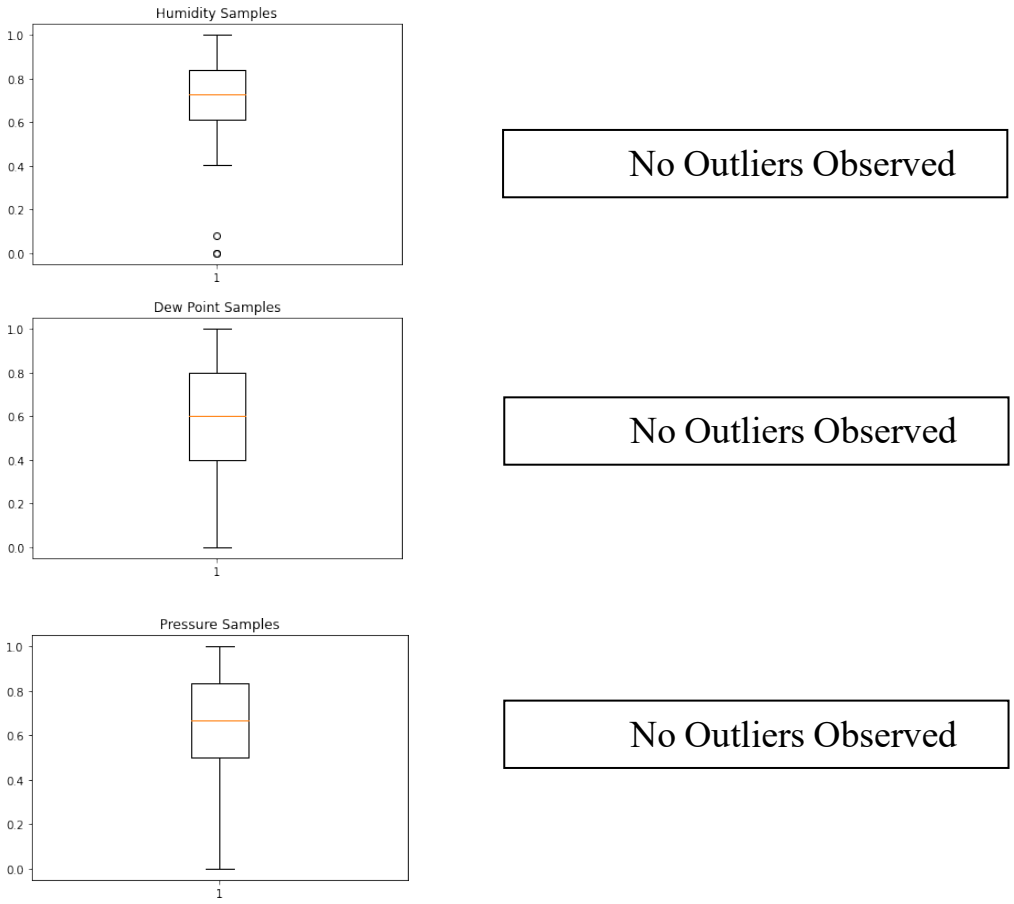

Figure 3. Box plots generated to visualize the presence of outliers in the dataset generated for Kandy

3.2. Identifying Variations in the Data Frame

Due to the considerably higher number of dimensions, it proved difficult to visualize the position of data points in all the dimensions. Scatter plots can only help in the visualization of data up to three dimensions and hence were not considered an efficient solution. For this purpose, Andrew's plots were used.

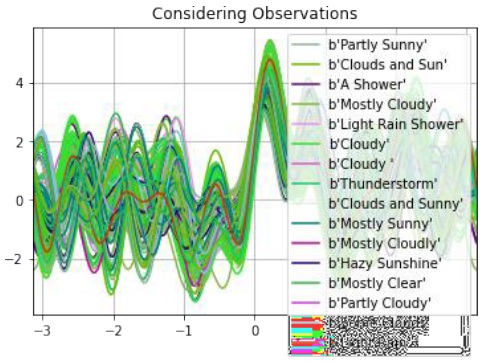

Figure 5. Andrew's plot for the labels in the Kandy dataset

Though there were considerable number of variations among the plot corresponding to each of the available output labels, it did not illustrate any curves which deviated from the rest to a considerable degree. All the curves were clustered together. Hence the outlier rectification procedure was therefore

verified to be successful. A downside observed in the plot were the considerably bad signal-to-ink-ratio observed due to the number of curves being overlaid exceeding the recommended amount. Nonetheless, the main purpose of verifying the absence of outliers was confirmed.

3.3. Diagnosing the Model

Initially, it was noted that the algorithm suffered from considerably high bias as it failed to perform well in neither the training nor the testing datasets. This was accounted for by increasing the sample size of the dataset.

Table 1. Results After Accounting for High Bias

	Kernel	C	Degree	Coef0	Gamma	Accuracy
Training Set	poly	10	1	0.01	auto	0.46
Testing Set	poly	10	1	0.01	auto	0.15

It was noted that both the overall accuracies when using either training or testing dataset decreased, but it was noted that the performance of the model for the training dataset was considerably higher compared to that of the testing dataset. The overall decrease in the performance was concluded to be due to the dataset undergoing significant changes and hence resulting in a different optimal parameter vector. This could have also been due to the considerably larger testing dataset resulted. The manual Grid Search and Random Search optimization methods were yet again carried out to yield the optimal results indicated in Table 2.

Table 2. Results After Optimizing for the Second Time

	Kernel	C	Coef0	Degree	Gamma	Accuracy
Training Set	poly	8	144	3	auto	0.86
Testing Set	poly	8	144	3	auto	0.26

The performance on the training set was considerably higher compared to that of the training set. According to the machine learning model diagnosis findings from (Andrew, 2011), it was concluded that the model was overfitting. One effective action recommended to be taken to account for overfitting was to reduce the number of features of the dataset. The dimensional size was then reduced to a size of twenty such that only the real time readings obtained from the sensors were used instead of also considering past values. The hence resulted dataset was utilized to retrain the model and obtain the optimal hyperparameters.

Table 3. Results After Optimizing for the Third Time

	Kernel	C	Coef0	Degree	Gamma	Accuracy
Training Set	poly	8	144	3	auto	0.92
Testing Set	poly	8	144	3	auto	0.57

Upon carrying out this downsizing of the label states, it was noted that the training accuracy yielded increased.

Table 4. Results After Reducing the Resolution of the Labels

Parameter	kernel	C	Coef0	Degree	gamma	Accuracy
Training Set	poly	8	144	3	auto	0.92
Testing Set	poly	8	144	3	auto	0.64

To maintain the time series nature of the dataset, the order of the samples was not randomized prior to using to train the model. Hence upon reobserving the data frame, it was noted that the label 'Clear' present in the training set was not observed in the training set. Hence, when the SVM saw this variable for the first time, it had considerable difficulty in identifying it as it was not initially trained to identify such a label. Hence, some samples which consisted of the label 'Clear' were randomly shifted to the training dataset. This was done due to the lack of data that was available to carry out the training procedure effectively.

Table 5. Results After Randomizing the Order of the Samples to a Certain Degree

Parameter	Kernel	C	Degree	Coef0	Gamma	Accuracy
Training Set	poly	3	3	144	auto	0.92
Testing Set	Poly	3	3	144	auto	0.86

3.4. Principal Component Analysis

The Principal Component Analysis was performed for both the Kandy and Badulla datasets generated. The results can be interpreted as follows.

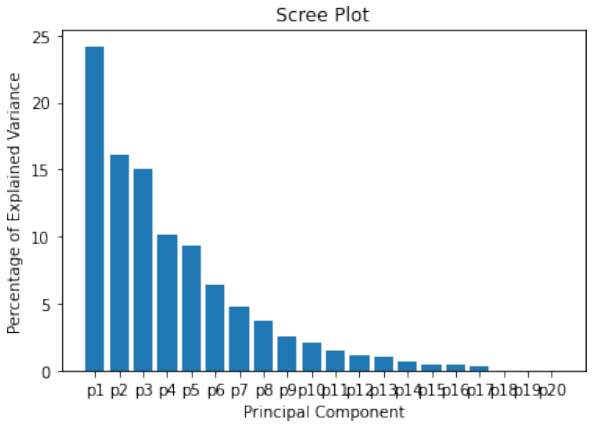

Figure 6. Scree Plot Resulted for the Kandy dataset

Each principal component contributed some level of information of the data and by leaving out principal components, data can be lost. If the first few PCs have caught majority of the information, the rest is negligible. An ideal plot should bend at an 'elbow' and then flattens out. However, as observed from Figure 6, the plot is far from ideal.

If there existed more than 3 principal components, as is the case with both plots above, according to (Ngo, 2018), PCA is not recommended. Therefore, the results of the PCA were not considered when carrying out the feature reduction process and the features were selected considering the physical implementations of the weather nodes. The inadequacy of taking the results of the principal component analysis into account is better highlighted when observing the results of the PCA biplots for the data frame.

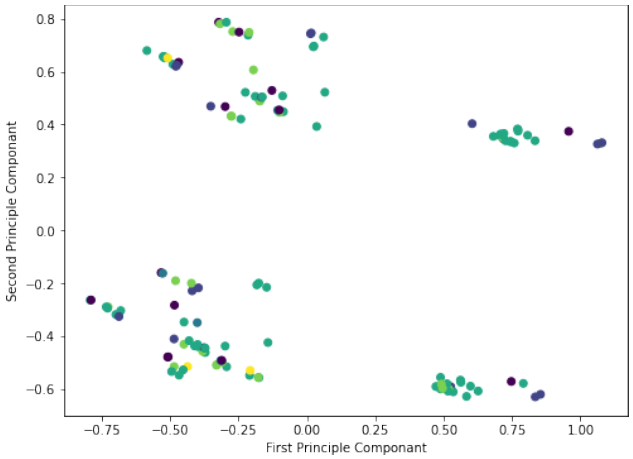

Figure 7. Biplot resulted for the Kandy dataset

Instead of discarding any samples, PCA reduces the number of dimensions by constructing principal components, which in turn describe variation and accounts for the varied influences of original characteristics. These influences are then backtracked from the plot to find what produces the differences among clusters. As observed above, the different components are not effectively clustered. PCA was not considered a viable candidate for model optimization for this specific instance.

3.5. Performance Evaluation

Confusion Matrix of the SVM

```
array([[14,  0,  0,  0,  0,  0],
       [ 0, 16,  0,  1,  0,  0],
       [ 0,  0,  3,  0,  0,  0],
       [ 1,  0,  0, 59,  1,  0],
       [ 1,  1,  0,  5, 17,  0],
       [ 0,  0,  0,  0,  0,  3]], dtype=int64)
```

Confusion Matrix of the KNN Implementation

```
array([[ 9,  1,  0,  1,  0,  0],
       [ 1, 11,  0,  1,  2,  0],
       [ 2,  0,  0,  0,  0,  0],
       [ 5,  4,  0, 35,  2,  0],
       [ 2,  2,  0,  3, 11,  0],
       [ 2,  0,  0,  1,  0,  0]], dtype=int64)
```

Upon glance, it is evident that the true positive and the true negative rate of the confusion matrix corresponding to the SVM is considerably higher than that of the confusion matrix corresponding to the KNN implementation.

4 CONCLUSIONS

The SVM performed better due to it being able to perceive time series forecasts to a considerably higher degree. In addition to this, the SVM had a considerably higher number of parameters which were adjustable, which indicated that the SVM was more flexible to optimization procedures.

Table 6. Final Optimal Performance Comparison

Model	**KNN**	**SVM**			
Optimal Parameters	K = 27	Kernel: Poly	C = 3	Degree = 3	Coef0 = 144
Accuracy	61%	86%			

The accuracy reached by the SVM, for the time span of consideration, exceeded the general accuracy observed in weather prediction models in the local industry. It should be noted that deep learning models would have a considerably higher level of accuracy, however due to the lack of data that was available, this implementation was not considered a viable candidate.

It can also be noted that the use of considerably smaller dataset and how the dataset was divided as 10% for the testing set and 90% for the training set may not have allowed the model to generalize to a satisfactory degree. This short come can be accounted for by means of utilizing a bigger dataset to train the model. Hence, as future work, the concept of making seasonal weather predictions based on a model trained using a seasonal dataset consisting of data corresponding to the same season spanning back numerous years and having a different model for each season can be further proved to be effective over the use of a model trained with an annual wise dataset.

REFERENCES

Aguileta, A., Brena, R., Mayora, O., & Molino-Minero-Re, E. (2019, September 3). NCBI- WWW Error Blocked Diagnostic. NCBI. Retrieved November 24, 2021, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6749203/%3E/

Andrew, N. (2011). Lecture 10.4 — Advice For Applying Machine Learning | Diagnosing Bias Vs Variance — [Andrew Ng]. (2017, January 1). YouTube. Retrieved July 2, 2021, from https://www.youtube.com/watch?v=fDQkUN9yw44

Andrew, NG. (2017 January 01). Advice For Applying Machine Learning | Diagnosing Bias vs Variance. YouTube. Retrieved June 01, 2021, from https://www.youtube.com/watch?v=fDQkUN9yw44

Atwell, C. (2021, March 18). What is sensor fusion? Fierce Electronics. Retrieved August 11, 2021, from https://www.fierceelectronics.com/sensors/what-sensor-fusion

Behera, B. Kumari, S. Kumari, A. and Kumar, A. (2020). APPLICATION OF IoT AND WEATHER PREDICTION FOR ENHANCEMENT OF AGRICULTURAL PRODUCTIVITY
Researchgate. Retreated September 16, 2021, from https://www.researchgate.net/publication/347916525_APPLICATION_OF_IoT_AND_WEATHER_PREDICTION_FOR_ENHANCEMENT_OF_AGRICULTURAL_PRODUCTIVITY

Bhandari, A. (2021, July 23). Confusion Matrix for Machine Learning. Analytics Vidhya.
Retrieved September 2, 2021, from https://www.analyticsvidhya.com/blog/2020/04/confusion-matrix-machine-learning/

Dawson, R. (2011). How Significant is a Boxplot Outlier. Journal of Statistics Education, volume (19). Retrieved September 8, 2021, from http://jse.amstat.org/v19n2/dawson.pdf Development Team Scikit-learn. (2007). 1.13. Feature selection. Scikit-Learn. Retrieved November 20, 2021, from https://scikit-learn.org/stable/modules/feature_selection.html

F. (2021, April 12). Hyperparameter Tuning | Evaluate ML Models with Hyperparameter Tuning. Analytics Vidhya. Retrieved October 19, 2021, from https://www.analyticsvidhya.com/blog/2021/04/evaluating-machine-learning-models-hyperparameter-tuning/

Hewage, P. (2020, June 22). Deep learning-based effective fine-grained weather forecasting model. SpringerLink. Retrieved November 14, 2021, from https://link.springer.com/article/10.1007/s10044-020-00898- 1?error=cookies_not_supported&code=d06d0643-f8a7-4aee-a94a-6cf282ca6588

Isabel, M. 2021.Working with higher dimensional data. 2021. Medium. Retrieved June 05, 2021, from.https://medium.com/working-with-high-dimensional-data/working-with-high-dimensional- data-9e556b07cf99#:~:text=Some%20of%20the%20basic%20algorithms,a%20classification%20or%20regression%20problem

Jaadi, Z. (2021, December 1). A Step-by-Step Explanation of Principal Component Analysis (PCA). Built In. Retrieved December 29, 2021, from https://builtin.com/data-science/step-step- explanation principal-component-analysis

Nautiyal, D. (2021, August 20). ML | Underfitting and Overfitting. GeeksforGeeks. Retrieved November 23, 2021, from https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/

Ngo, L. (2020, September 21). How to read PCA biplots and scree plots. BioTuring's Blog. Retrieved August 23, 2021, from https://blog.bioturing.com/2018/06/18/how-to-read-pca-biplots-and-scree-plots/

Steiger, J. H. (2015, February 16). Principal Components Analysis. Statpower. Retrieved September 4, 2021, from http://www.statpower.net/Content/312/R%20Stuff/PCA.html