Predicting Road Visibility in Shenzhen Using Machine Learning

Anonymous Honglin wei submission

Paper ID 2022300013

Abstract

This study focuses on predicting road visibility in Shenzhen by leveraging real-time image data from the city's road monitoring system alongside meteorological data using machine learning methodologies. Initially, historical data were analyzed to identify correlations between factors such as precipitation, humidity, and wind speed with road visibility, with distribution maps illustrating these relationships. Subsequently, various machine learning algorithms, including Support Vector Regression (SVR) and Decision Tree Regression, were employed to construct predictive models. The performance of these models was evaluated using metrics such as Mean Squared Error (MSE), with results indicating that SVR provides better predictive accuracy compared to Decision Tree Regression.

This research offers an effective approach for predicting road visibility in Shenzhen and provides valuable insights for meteorological monitoring and traffic management in similar urban settings.

1. Introduction

In modern urban traffic management, the prediction and monitoring of road visibility are of paramount importance. Visibility, as a common indicator in daily life, plays a crucial role in driving and transportation. Especially in rapidly developing cities like Shenzhen, with high traffic density and a large number of vehicles, timely and accurate prediction of road visibility is particularly crucial.

1.1. Background

In Shenzhen, a thriving modern city in China, managing traffic safely amid rapid growth is crucial. Visibility on roads directly affects people's safety and daily lives.

Machine learning offers new ways to predict road visibility. By using weather data and machine learning techniques, we can accurately predict visibility, giving early warnings for accidents and guiding traffic management decisions.

Figure 1. Haze weather in Shenzhen. Retrieved from Internet

1.2. Existing methods

1. Hybrid Model of CNN and LSTM

Zhang, Ling, Zhao Yang, and Jian Zhang proposed a hybrid model combining Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks for road visibility estimation [6]. The CNN extracts spatial features from images, while the LSTM captures temporal dependencies in the data. This hybrid approach leverages both spatial and temporal information, leading to high accuracy in visibility prediction. It is particularly robust in dynamic and changing environments.

2. Real-time Visibility Estimation Using Deep Neural Networks

He, Yong, Wei Zhang, Lei Xu, and Xin Li developed a deep neural network model specifically designed for real-time visibility estimation in foggy conditions [2]. The model is trained on a large dataset of foggy weather images and demonstrates high accuracy and fast processing speed, making it suitable for real-time applications. The model effectively handles varying levels of fog density.

3. Visibility Estimation of Expressways Based on Deep Learning

Qi, Chao, Jingwen Zhang, Mingjie Huang, and Bo Wang introduced a deep learning model trained on expressway images to estimate visibility [4]. The model architecture focuses on capturing the features specific to highway environments. It achieves high accuracy in visibility estimation

on expressways, demonstrating the potential of deep learning in specialized transportation contexts.

4. Multi-scale Convolutional Neural Networks for Real-time Visibility Estimation

Shi, Qianqian, Yanfeng Han, and Xing Liu proposed a multi-scale CNN approach, where features are extracted at different scales to capture both fine and coarse details in the images [5]. This multi-scale approach enhances the model's ability to predict visibility accurately across various weather conditions and environments, providing robust real-time performance.

5. Foggy Road Visibility Estimation Using Generative Adversarial Networks (GANs)

Liu, Yiming, Qian Sun, and Jian Wang employed Generative Adversarial Networks (GANs) to improve the visibility of road images by generating de-fogged images, which are then used for visibility estimation [3]. The GAN-based method shows significant improvements in image clarity and visibility estimation accuracy. It excels in extremely foggy conditions where traditional methods may struggle.

6. Conclusion

However, common limitations include high computational demands, data specificity, and potential generalization issues. Future research should focus on developing more generalized models that can handle diverse weather conditions and road types while optimizing computational efficiency.

2. Machine learning methodology

In my study, I employ traditional machine learning algorithms (Support vector regression, Decision tree) . This selection allows for simpler implementation and less computational demand compared to deep learning methods.

2.1. Support Vector Regression

Support Vector Regression (SVR) is a type of regression technique that uses Support Vector Machines (SVM) to function as a regression estimator. The main principle behind SVR is to identify a hyperplane in a high-dimensional space that has the maximum margin with respect to the training data. In other words, SVR aims to find the hyperplane that minimizes the generalization error while still fitting the training data well.

The equation for SVR can be represented as:

$$y = w^T x + b$$

where: - y is the predicted output value - w is the weight vector - x is the input sample - b is the bias term

The goal of SVR is to find the optimal values of w and b that minimize the error between the predicted output and the true output while also maximizing the margin between the hyperplane and the training data.

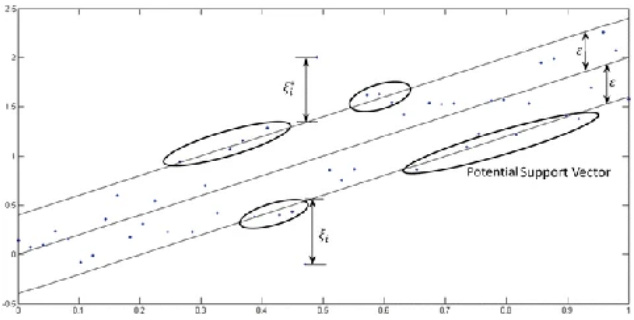

Figure 2. One-dimensional linear SVR . Adapted from [1]

Additionally, SVR uses a kernel function to map the input data into a higher-dimensional space in order to find a hyperplane that can separate the data points. The most commonly used kernel functions in SVR are the linear kernel, polynomial kernel, and radial basis function (RBF) kernel.

The optimization problem for SVR can be formulated as:

$$\min_{w,b,\xi,\xi^*} \frac{1}{2}||w||^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*)$$

subject to the constraints:

$$y_i - w^T x_i - b \leq \epsilon + \xi_i,$$

$$w^T x_i + b - y_i \leq \epsilon + \xi_i^*,$$

$$\xi_i, \xi_i^* \geq 0,$$

where: - N is the number of training samples - ξ_i, ξ_i^* are slack variables that allow for some deviation from the margin - C is the regularization parameter - ϵ is the margin of tolerance - w is the weight vector - b is the bias term

In summary, the principle of SVR involves finding the optimal hyperplane that minimizes the error while maximizing the margin with respect to the training data. The use of kernel functions allows SVR to handle non-linear relationships between the input data and the output, making it a powerful tool for regression tasks.

2.2. Decision tree Regression

A 1D regression with decision tree.

The decision trees is used to fit a sine curve with addition noisy observation. As a result, it learns local linear regressions approximating the sine curve.

We can see that if the maximum depth of the tree (controlled by the max-depth parameter) is set too high, the decision trees learn too fine details of the training data and learn from the noise, i.e. they overfit.

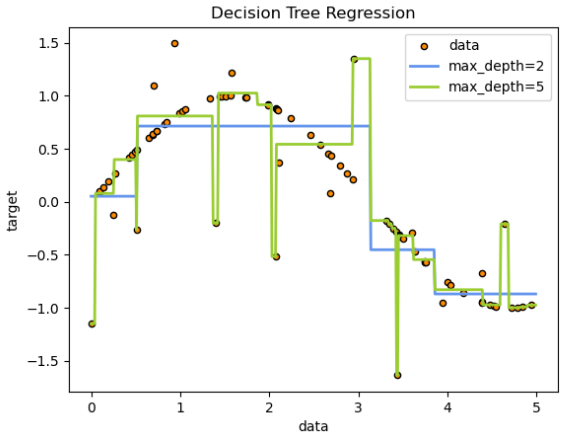

Figure 3. Decision tree.Retrieved from https://scikit-learn.org/

3. Experiments

3.1. Data Collection, Preprocessing and Analysis

Data Collection from the website:

https://opendata.sz.gov.cn/data/dataSet/toDataDetails/29200_00903518

This dataset contains hourly telemetry data from Shenzhen, with 3,730 records and 64 fields. The data types are primarily integers and strings. Some example fields include wind direction, cloud height, relative humidity, datetime, surface minimum temperature, grassland maximum temperature, automatic precipitation amount, minimum station pressure, maximum wind speed, and more.

Collection Time: The timestamps in the dataset range from August 9, 2015, to April 6, 2020, depending on the specific record.

1. We find that some characteristic sets have a lot of missing values, so we first delete these sets.

2. Since the processed dataset still has some missing values, we choose to replace the missing values with 0.

3. We calculate the correlation values between visibility and other characteristic sets.

4. We observe that the dataset has many characteristics, and many of them have low correlation with visibility. Therefore, we only select characteristics with a correlation value greater than 0.15.

5. We explore the distribution for every characteristic.

6. We calculate the mean and variance for each characteristic. The results are shown in Table 2.

Feature	Correlation
RELHUMIDITY	0.311094
MINRELHUMIDITY	0.306294
INSTANTWINDV	0.212944
HEXMAXWINDV	0.198315
WINDV10MS	0.162007
MAXWINDV10MS	0.156455
AUTOPRECIPAMOUNT	0.156363
GRASSLANDMAXTEMP	0.152857

Table 1. Correlation of various features with VISIBILITY.

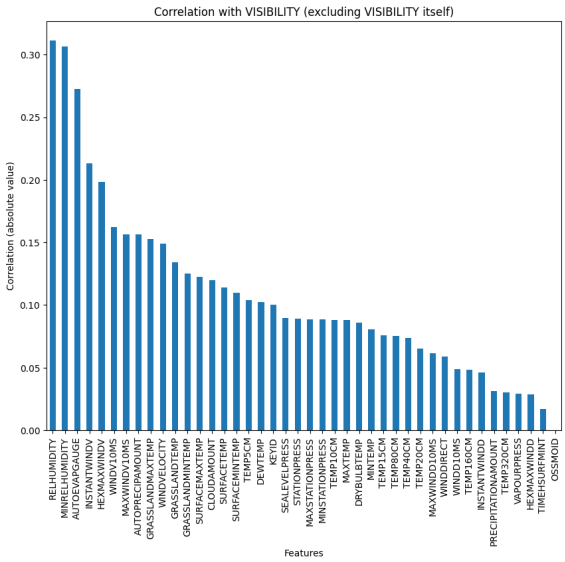

Figure 4. Correlation with Visibility

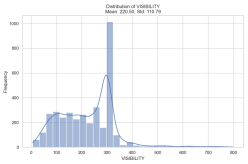

Figure 5. VISIBILITY

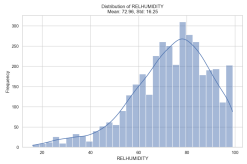

Figure 6. RELHUMIDITY

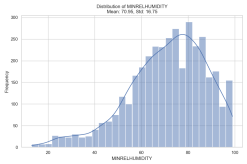

Figure 7. MINRELHUMIDITY

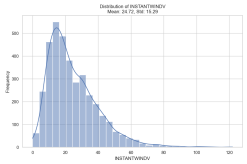

Figure 8. INSTANTWINDV

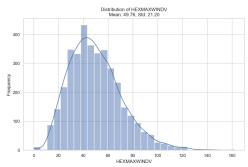

Figure 9. HEXMAXWINDV

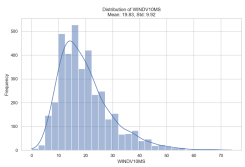

Figure 10. WINDV10MS

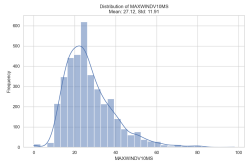

Figure 11. MAXWINDV10MS

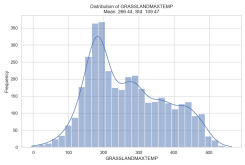

Figure 12. GRASSLANDMAXTEMP

Table 2. Mean and Standard Deviation

Feature	Mean	Standard Deviation
VISIBILITY	220.496	110.7851
RELHUMIDITY	72.96193	16.2497
MINRELHUMIDITY	70.95121	16.75499
INSTANTWINDV	24.71689	15.29268
HEXMAXWINDV	49.76139	21.20455
WINDV10MS	19.82949	9.918758
MAXWINDV10MS	27.12172	11.91181
GRASSLANDMAXTEMP	266.4373	109.4651

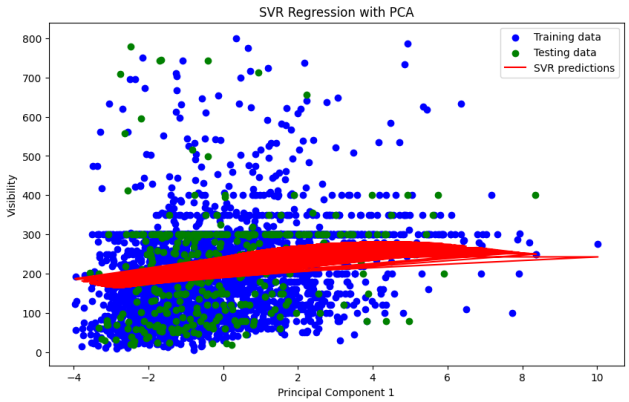

Figure 13. SVR Regression with PCA

3.2. Support Vector Regression

To develop an SVR (Support Vector Regression) algorithm to predict visibility using the given features, we will follow these steps:

1. Load the dataset from the provided Excel file.
2. Extract one principal component using PCA.

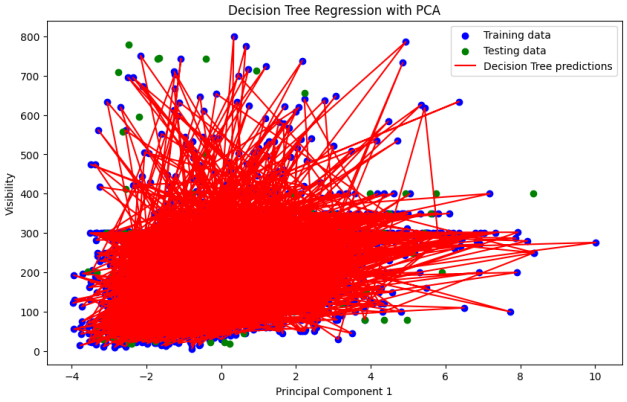

Figure 14. Decision Tree Regression with PCA

The SVR model predictions form a smoother curve through the data, indicating that the model is capturing a general trend rather than fitting the noise. The red line follows a trend with less variance compared to the Decision Tree model, suggesting a more generalized model.

3. Split the data into training and testing sets.
4. Train the SVR model using the training set.
5. Evaluate the model using 10-fold cross-validation on the training set and compute the average mean squared error (MSE).
6. Print the learned model and the average MSE.

3.3. Decision tree aggression

To develop an Decision tree aggression algorithm to predict visibility using the given features, we will follow these steps:

1. Load the dataset from the provided Excel file.
2. Extract one principal component using PCA.
3. Split the data into training and testing sets.
4. Train the Decision Tree model using the training set.
5. Evaluate the model using 10-fold cross-validation on the training set and compute the average mean squared error (MSE).
6. Print the learned model and the average MSE.

The Decision Tree model's predictions create a dense, crisscrossed web of red lines connecting the actual visibility values with the predicted values. This suggests that the Decision Tree model has a high variance, fitting the training data very closely (possibly overfitting). The model captures a lot of noise from the training data.

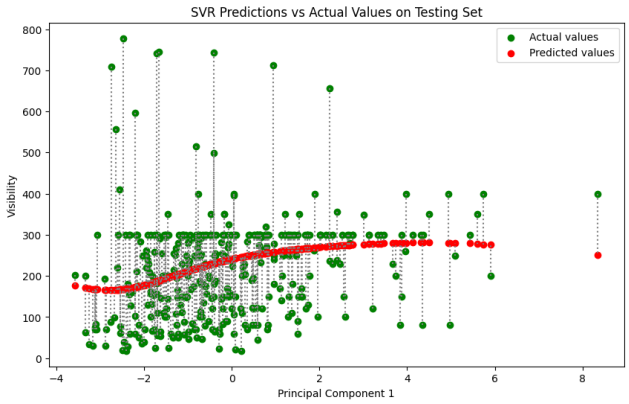

Figure 15. SVR Predictions vs Actual Values on Testing Set

4. Results and Analysis

4.1. Support Vector Regression

Metric	MSE
Training Mean Squared Error (MSE)	11006
Testing Mean Squared Error (MSE)	15326

Table 3. Mean Squared Error (MSE) for Training and Testing in svr

The MSE is higher on the testing set compared to the training set, indicating that the SVR model performs better on the data it was trained on and shows some degree of overfitting. The difference between training and testing MSE suggests that while the model generalizes reasonably well, there is still a noticeable performance drop on unseen data.

Trend Line: The SVR predictions (red dots) show a smoother, more continuous trend line, which indicates that the SVR model captures a general trend or pattern in the data.

Error Distribution: The predicted values are generally closer to the actual values (green dots), especially in the middle range of the principal component values. The spread of the predictions appears narrower compared to the Decision Tree model.

Outliers: There are fewer extreme outliers in the SVR predictions compared to the Decision Tree model. The SVR model seems to handle outliers better, keeping most predictions within a reasonable range from the actual values.

4.2. Decision tree aggression

Similar to the SVR model, the Decision Tree model shows a higher MSE on the testing set compared to the training set, indicating overfitting. However, the MSE values are higher overall compared to the SVR model, suggesting that the Decision Tree model may not be as effective at capturing the underlying patterns in the data.

Metric	MSE
Training Mean Squared Error (MSE)	21697
Testing Mean Squared Error (MSE)	23787

Table 4. Mean Squared Error (MSE) Results in decision tree

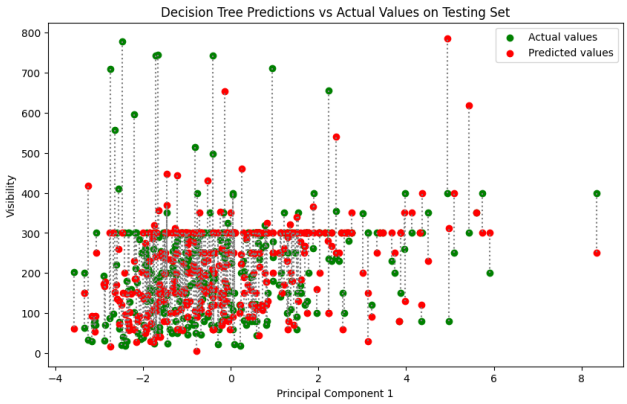

Figure 16. Decision Tree Predictions vs Actual Values on Testing Set

Trend Line: The Decision Tree predictions (red dots) do not form a clear trend line. Instead, they appear more scattered around the actual values.

Error Distribution: The predicted values are spread out more widely around the actual values. This suggests that the Decision Tree model may be overfitting, capturing more noise in the training data which leads to more variability in predictions.

Outliers: There are more extreme outliers and a wider spread of predictions compared to the SVR model. This indicates that the Decision Tree model is less robust to noise and outliers in the data.

4.3. Comparison of Two Methods

Overall, the SVR model demonstrates better performance in terms of MSE for both training and testing sets compared to the Decision Tree model. While both models exhibit overfitting, the SVR model's lower MSE values indicate it is more effective at modeling the data. However, the larger generalization gap in the SVR model suggests that further tuning or regularization might be needed to improve its generalization capability.

The SVR model captures the underlying trend more smoothly, has a narrower error distribution, and handles outliers more effectively. The Decision Tree model, on the other hand, shows signs of overfitting with more scattered predictions and a higher number of extreme outliers.

4.4. Comparison with existing methods

Decision Tree Regression with PCA and SVR with PCA are simpler and more interpretable models but tend to have higher error rates and may overfit or underfit depending on the data complexity.

Deep Learning Approaches (e.g., Hybrid CNN-LSTM, multi-scale CNNs, GANs) demonstrate superior performance in capturing complex patterns, handling various weather conditions, and providing robust real-time performance. These methods, however, require more computational resources and larger datasets for training.

In scenarios where computational resources and training data are abundant, deep learning models significantly outperform traditional methods like Decision Trees and SVR in terms of accuracy and robustness, especially in dynamic and challenging environments.

By comparing the results, it is evident that deep learning models offer substantial improvements in visibility estimation tasks over traditional machine learning approaches, albeit at the cost of increased complexity and resource requirements.

5. Conclusion

In this study, we explored the prediction of road visibility in Shenzhen using machine learning methods. We collected and analyzed meteorological data to identify key factors affecting visibility, such as humidity and wind speed. Two machine learning algorithms, Support Vector Regression (SVR) and Decision Tree Regression, were implemented to create predictive models.

Our results showed that SVR outperformed Decision Tree Regression in terms of predictive accuracy, with lower Mean Squared Error (MSE) on both training and testing datasets. While both models exhibited some degree of overfitting, the SVR model captured the underlying trends more effectively and demonstrated better generalization to new data.

Overall, this research highlights the potential of traditional machine learning techniques for predicting road visibility, providing valuable insights for urban traffic management and meteorological monitoring. Future work could explore more advanced models and larger datasets to further improve predictive performance.

References

[1] M. Awad, R. Khanna, M. Awad, et al. Support vector regression. In *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*, pages 67–80. Apress, New York, NY, 2015. 2

[2] Yong He, Wei Zhang, Lei Xu, and Xin Li. Real-time visibility estimation in foggy weather using deep neural networks. *IEEE Access*, 8:36212–36223, 2020. 1

[3] Yiming Liu, Qian Sun, and Jian Wang. A novel approach for foggy road visibility estimation using generative adversarial networks. *Applied Soft Computing*, 107:107379, 2022. 2

[4] Chao Qi, Jingwen Zhang, Mingjie Huang, and Bo Wang. Visibility estimation of expressways based on deep learning. *Journal of Transportation Safety Security*, 13(2):191–210, 2021. 1

[5] Qianqian Shi, Yanfeng Han, and Xing Liu. Multi-scale convolutional neural networks for real-time visibility estimation. *Neural Computing and Applications*, 33:4371–4382, 2021. 2

[6] Ling Zhang, Zhao Yang, and Jian Zhang. Road visibility estimation using a hybrid model of convolutional neural networks and long short-term memory. *Journal of Intelligent Transportation Systems*, 25(4):345–358, 2021. 1