Urban Land Cover Classification Algorithms and Simulation

Rishik Gannavarapu

Arizona State University

[1]*Received: 21 June 2024; Accepted: 31 July 2024; Published: 27 August 2024*

ABSTRACT

This study uses the UCI urban land cover dataset to propose a unique integrated discriminant model. Computers are used in remote sensing image classification to process features and classify models of images captured by satellite remote sensing. Current research primarily focuses on feature extraction from images and model training to accomplish feature-based classification for accurate land type classification. This study suggests a novel approach for the latter by training each set of data taken from the many coarser scales of the original image using the supervised machine learning algorithms Random Forest, SVM, Naïve Bayes, and KNN. The prediction accuracy of the algorithms is then compared. The model at that scale is the algorithm that works best with the matching scale data. Lastly, the ultimate result is obtained by weighing the predictions made by each model. Tests have demonstrated that the model outperforms current techniques on every scale.

INTRODUCTION

A. Overview

Traditional geography, environmental science, and geoscience studies depend on data collected by field visits or ground monitoring stations, which have costly, time-consuming, and long-cycle flaws. These aspects have been significantly enhanced by remote sensing; for example, Landsats can now acquire global surface data in over ten days, while shorter-cycle ocean satellites can obtain global pictures in a few hours.

Multi-phase data is crucial to the evolution study, and the time and expense of data collection are significantly decreased [1–5].

People use crowdsourced data to create global framework data resources, like road networks, town names, and attractions, which are crucial given the advancements in satellite remote sensing and navigation technologies function in quickly interpreting satellite imagery [6,7]. OpenStreetMap has been used as a basis map by international ERDAS firms to support a variety of data applications in their ERDAS TITAN 2009 products [8–10]. This research, which focuses on the advancement of remote sensing technology, reproduces the model under study using a small amount of data and refines it to suggest an improved model based on the same data [1].

B. What We Do

This article initially processes the outlier and verifies the data's integrity using the Urban Land Cover dataset that was acquired from UCI. A training and testing set is created by dividing the data sets by tenfold and grouping them to a coarser scale using the dimension reduction method due to the large number of characteristics. evaluating the model on prediction sets, assessing accuracy, and training on training sets using several supervised machine learning

[1] *How to cite the article:* Gannavarapu R., August 2024; Urban Land Cover Classification Algorithms and Simulation; *International Journal of Innovations in Scientific Engineering*, Jul-Dec 2023, Vol 20, 15-24

algorithms (RandomForest, SVM, Naïve Bayes, KNN) [11]. Using testing accuracy and the best model from a 10-fold cross-training, choose the best algorithm for every piece of data.

To match the coarser scale range of the data, seven ideal models were found [12]. This study determines the predicted weight of each class for each set of models using the confusion matrix under the various coarser scales. It then aggregates the prediction data from seven sets of models to produce the final verification results, determine the final accuracy, and contrast our model with the model in the reference paper [1].

C. New Data Intervention Prior to processing

Before being used, data must be pre-processed. Outlier analysis is the primary focus of preprocessing in this work. To find outliers, a box plot is necessary. Nevertheless, because the values of features under various coarser scale classes fluctuate greatly, so do the values among features within the same class.

First, data normalization is done. after the classification of normalized data by class and scale. This is an example of a box plot that we create using grouped data.

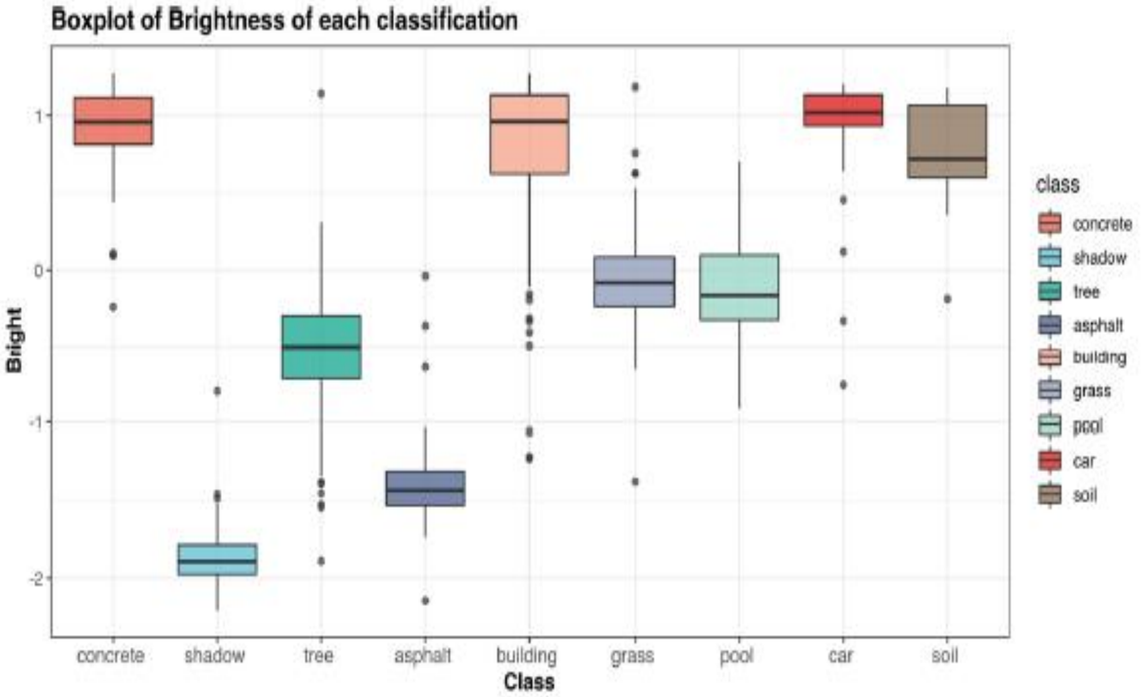

Figure 1: Box plot of asphalt in 20 coarser scale.

Figure 1 makes it clear that the distribution of data outside the quarter point contains a large number of discrete values, all of which are probably outliers.

Cook's distance is employed in this work to identify outlier values.

Cook Distance only searches for the most significant outliers in the data—that is, the sample that deviates the greatest from the model's predictions—in regression analysis in order to retain as much information as possible in the original data.

The precise technique for determining the Cook distance involves computing each sample's residuals and leverage value.

Both can be regarded as outliers if they are at a higher level, which raises the Cook distance.

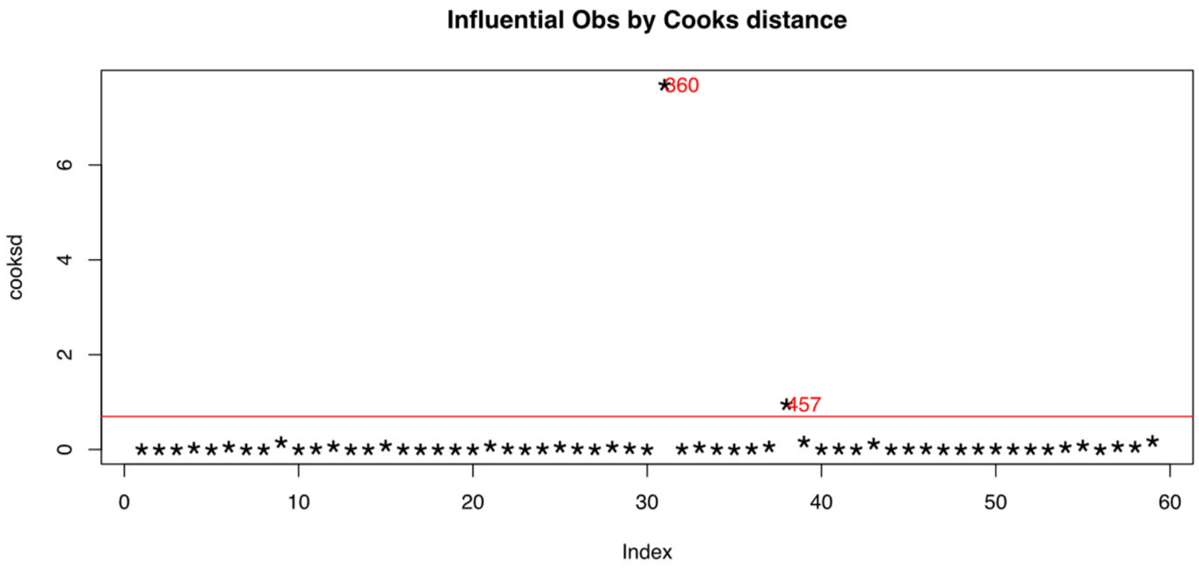

Figure 2 shows a scatter plot of asphalt's Cook's distance on a 20-coarse scale.

Two points over the red line in Figure 2 are regarded as outliers, and the red number represents their sample number. The red line is four times the average of the sub-dataset's Cook's distance. These outlier values are treated as missing values in this work, and the missing values are filled using multilinear regression.

$$D_i = \frac{\sum_{j=1}^{n} \left(\hat{Y}_j - \hat{Y}_{j(i)}\right)^2}{p \times MSE}$$

Sample point i must be eliminated from the data in order to calculate the Cook distance, and the remaining data must then be used for regression analysis. figuring out the regression residuals for every sample. This operation illustrates how the sample points affect the model's predictions and the bias that results.

Class label is represented by j, coarser scale by i, and variable label by k.

Create scatter plots using the computed Cook's distance to obtain:

MODEL CONSTRUCTION

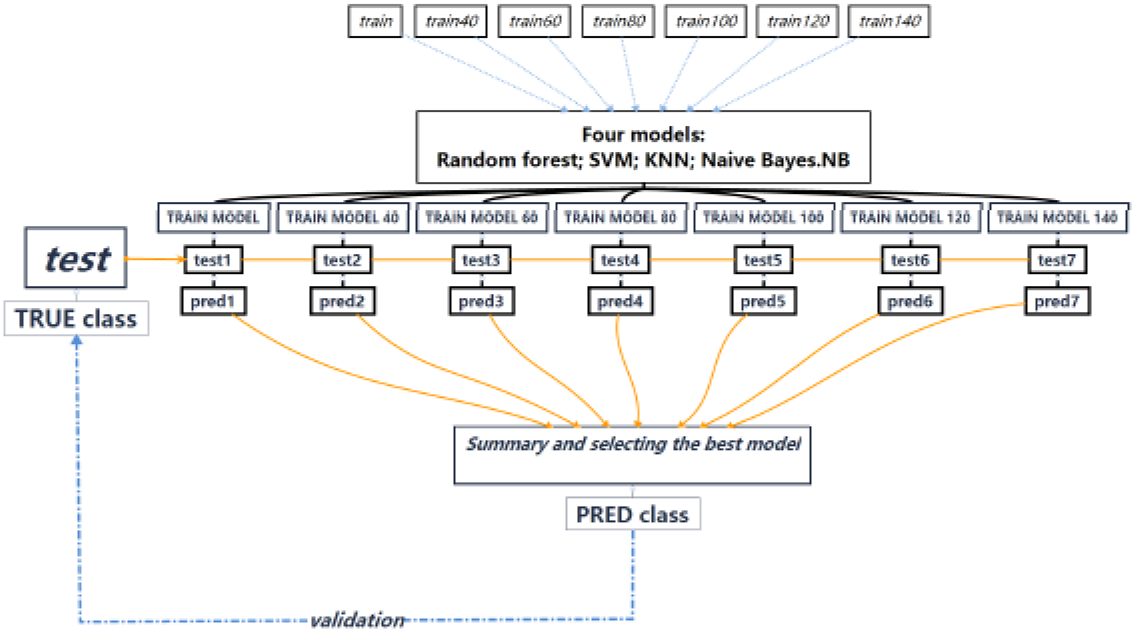

Figure 3: Flowchart of the model.

Figure 3 shows seven data groups arranged by scale at the top, which are then fed into four algorithms to provide the best methods for each scale. The forecast result is provided based on the predictions made by the seven groups. There are nine labels and 148 variables in all for urban land cover data. This study divides the characteristics into seven groups for supervised machine learning according to the coarser scale of remote sensing accuracy.

These four types of supervised machine learning algorithms—Random Forest, SVM, Naïve Bayes, and KNN—train each piece of data, compare each method's prediction accuracy, choose the best algorithm for each set of data, and then, The final predictions were obtained by voting on seven weighted sets of prediction results after each set's weight was adjusted.

A. Unprocessed Models

1) SVM

Among the supervised classification algorithms is SVM. Its basic premise is that the sample space has a variety of point kinds. Identifying a hyperplane that separates various sample types and dividing it should have the best generalization ability, meaning that each sample type should be closest to it and the sample points should be as far apart as possible.

2) The Random Forest

Random forests are praised as "methods that represent the high level of integrated learning techniques" due to their remarkable effectiveness in classification and regression, ease of implementation, low computing overhead, and simplicity.

3) KNN

One of the most basic machine learning algorithms is the KNN (K-Nearest Neighbor) approach, which was first put forth by Cover and Hart in 1968 and is a classification process in supervised learning. The formula is straightforward and easy to understand: if a sample falls into a category in the K most similar (or closest to the feature space) sample in the feature space, then the sample likewise falls into that category.

4) The Naïve Bayes

The classification technique known as Naïve Bayes is predicated on the Bayes theorem and conditional independent assumptions. The naïve Bayes approach is frequently used in applications including text categorization, spam filtering, and natural language processing because it is easy to implement and has very high learning and prediction efficiency.

B. The K-fold Model Parameter and Cross-Validation Loop of Self-Adaptation

Usually, cross-validation is employed to assess a machine learning model's performance. Cross-validation is more frequently employed when choosing a model. Although cross-validation has been used for a long time, there are still several issues with cross-validation research. How to select K in the most basic K-fold cross-validation is an intriguing subject. More intriguingly, cross-validation is frequently employed to ascertain parameters in other algorithms, such figuring out K's value in K-neighbor methods. Therefore, in K-fold cross-validation, we must first choose K.

K-fold cross-validation is the process of splitting the training data into K parts, training the model with the (K-1) parts, and assessing the model's quality with the remaining 1 part. K pieces of data are subjected to a sequential cycle of the procedure, and the K assessment outcomes are then merged, for example, by voting or average.

For instance, the training data D is split into nine training set copies and one test set copy for the 10-fold cross-validation. After the cycle, all of the evaluation findings are averaged. As is well known, various models contain a wide range of parameters, which affects the predicted accuracy of the models after training. (For example, Random Forests can change how many random trees are used, Naïve Bayes can choose internal functions, while SVM and KNN can alter the kernel function.

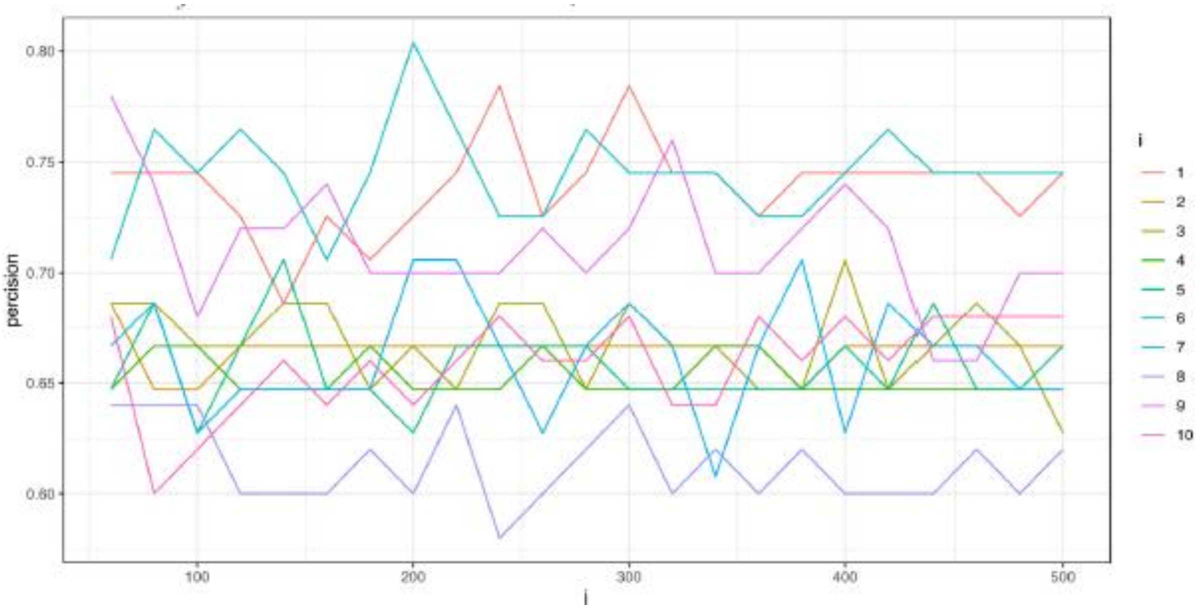

Figure 4: K-fold cross-validation with change of number of trees

In Figure 4, i denotes K-fold cross-validation and j is the number of random trees. Using the sixth fold as a prediction set, 200 is the random tree's maximum accuracy prediction; hence, in the optimal model selection, select it as the random forest's ideal parameter under the appropriate scale.

For instance, the training data D is split into nine training set copies and one test set copy for the 10-fold cross-validation. After the cycle, all of the evaluation findings are averaged. As is well known, various models have a large number of parameters.

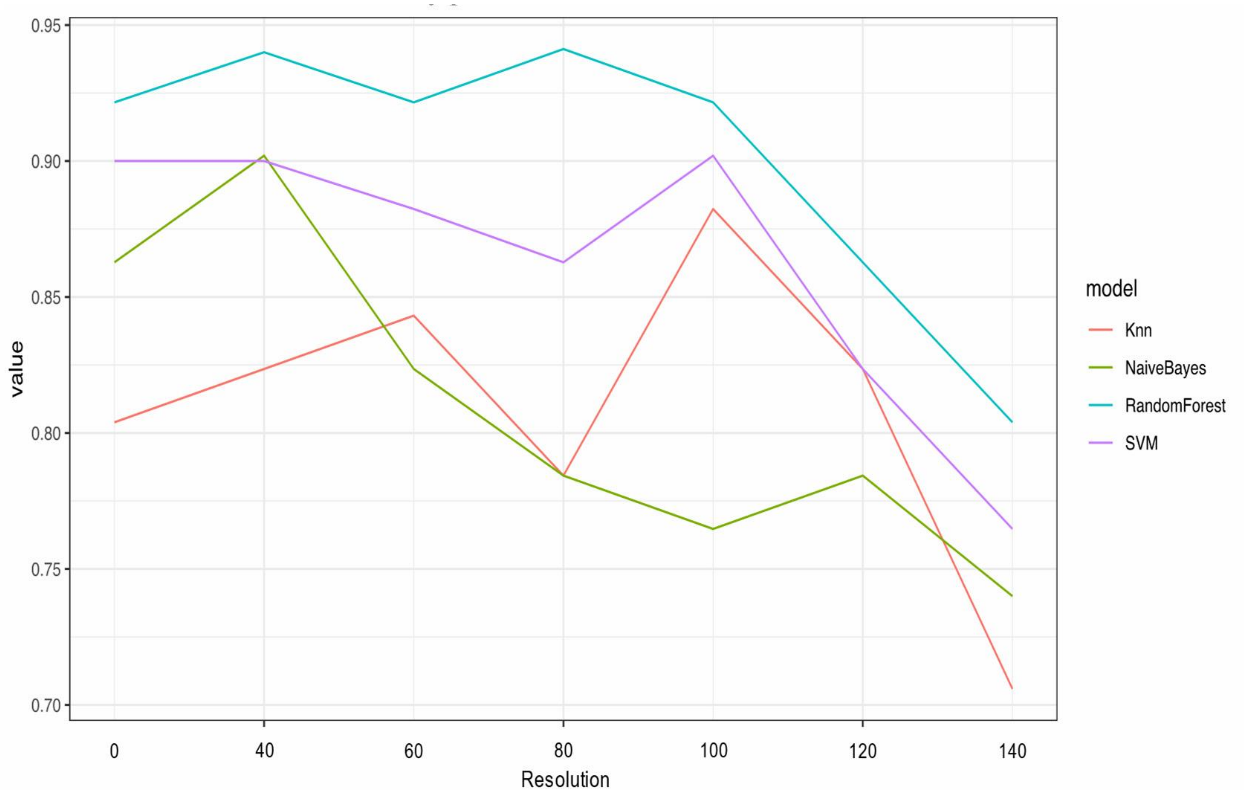

Figure 5: Optimal model selection

resulting in varying prediction accuracy following model training.

(For example, Naïve Bayes can choose internal functions, SVM and KNN can alter the kernel function, and Random Forests can modify the number of random trees.)

C. Choosing a Model

Following the preceding step's selection of the models' ideal parameters under various scales, we compared the testing accuracy under each grouped data set to determine which models were best under each scale. Selecting the top seven random forest models for this article is evident from Figure 5, which shows that the model of random forests performs exceptionally well at all coarser scales.

D. Confusion Matrix and Parameter for Weighting

When we compare our confusion matrix across seven distinct scale groups, we find that the prediction errors are focused in the following areas: soil to concrete, grass to concrete, grass to tree, concrete to building, and shadow to asphalt. To increase the accuracy of the predictions even more, this paper

Table 1: Weighting Results.

	0	40	60	80	100	120	140	final
1	car	car	car	car	building	building	building	car
2	concrete	concrete	concrete	concrete	concrete	concrete	concrete	concrete
3	concrete	concrete	building	building	building	building	building	building
4	concrete	concrete	concrete	concrete	concrete	concrete	building	concrete
5	concrete	concrete	concrete	concrete	concrete	concrete	concrete	concrete
6	grass	grass	tree	grass	grass	grass	grass	grass

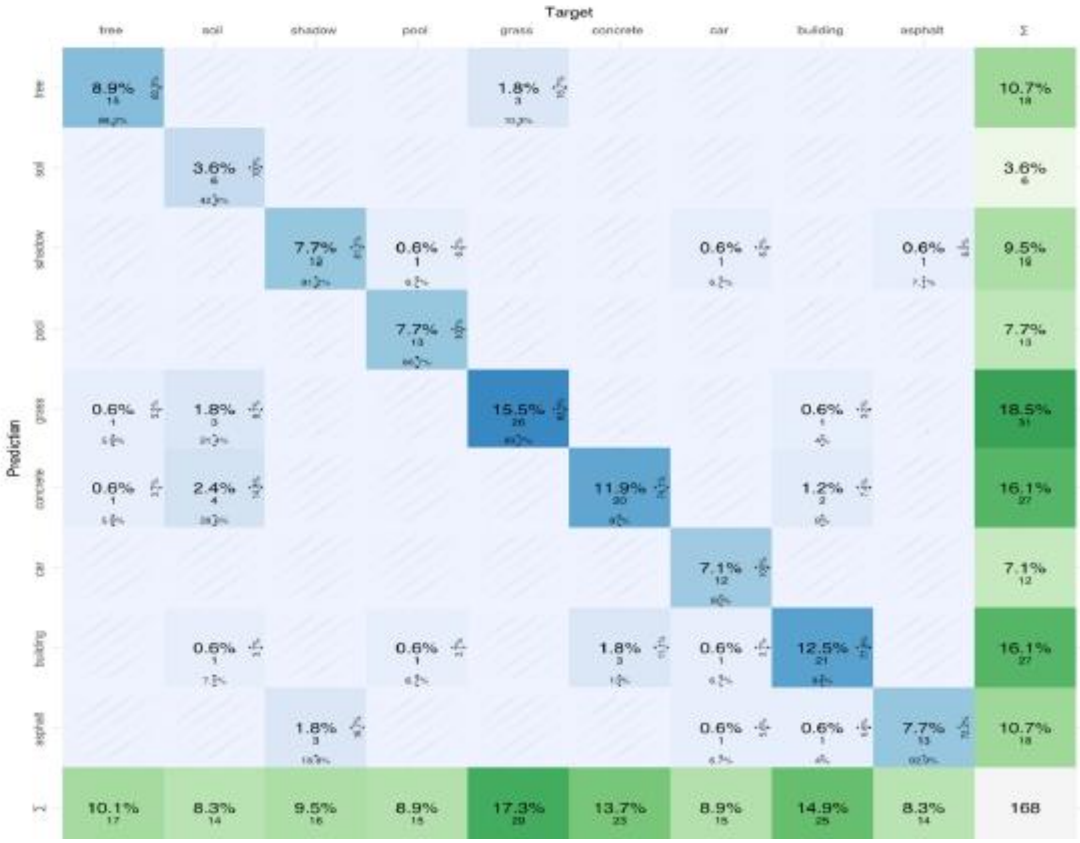

Figure 6: Confusion Matrix of final prediction results

We examine the aforementioned situations and choose the best prediction model for every prediction object. For instance, we give the model more weight when voting for asphalt since it is the best prediction model under the coarser scale of 80.

E. Last Prediction

The following Table 1 illustrates how the overall forecast is weighted to the largest and final forecast with the majority of the voting, based on the model weights determined above.

The confusion matrix produced by the final prediction result is shown in Figure 6. The total forecast accuracy is 82.7381%, which is 4% higher than that of earlier studies [1].

COMPARISON

Our models under the same remote sensing are significantly more accurate when compared to the article's results. Compared to what the model in the original research indicated, the forecast correctness rate is noticeably greater.

Note: The correctness of [1]'s classification.

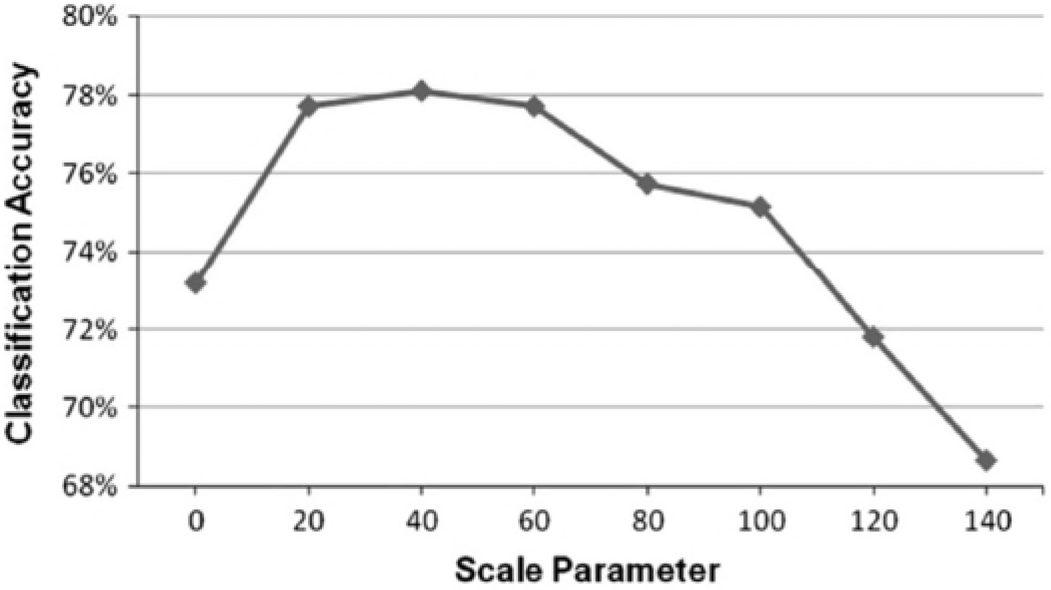

Figure 7: The outcome of the reference.

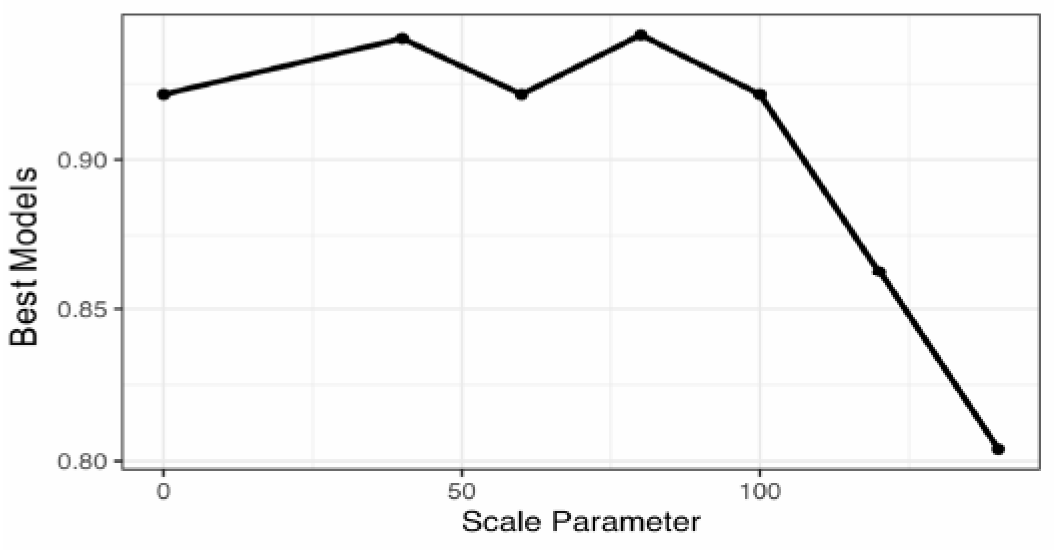

Figure 8: Our classification accuracy findings.

As can be seen from the comparison of Figures 7 and 8, our model's classification accuracy outperforms the findings of the reference at every scale. When compared to the reference in the total forecast, our model's accuracy was 82.7381% [1].

CONCLUSION

In this study, a novel architecture for urban land cover categorization is proposed, utilizing models for the properties of urban land cover data at various scales and four machine learning algorithms: Random Forest, SVM, Naïve Bayes, and KNN.

This work demonstrates that the architecture has superior prediction capacity at all scales by comparing it to the early research in this field. The comprehensive prediction accuracy is 82.7381%.

Future studies could concentrate further on improving predictive capacities by reducing certain class prediction errors.

REFERENCE

[1] Johnson, B., Xie, Z., 2013. Classifying a high-resolution image of an urban area using super-object information. ISPRS Journal of Photogrammetry and Remote Sensing, 83, 40-49.

[2] Johnson, B., 2013. High resolution urban land cover classification using a competitive multi-scale object-based approach. Remote Sensing Letters, 4 (2), 131-140.

[3] Hegde, Gaurav et al. "Urban Land Cover Classification Using Hyperspectral Data." ISPRS Technical Commission VIII Symposium, translated by Isprs Tech Commission et al., vol. 40-8, 2014

[4]"Urban Land Cover Classification Using Hyperspectral Data." ISPRS Technical Commission VIII Symposium, translated by Isprs Tech Commission et al., vol. 40-8, 2014

[5] Tong, Xiaohua et al. "Urban Land Cover Classification with Airborne Hyperspectral Data: What Features to Use?" Ieee Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 7, no. 10, 2014, pp. 3998-4009, doi:10.1109/jstars.2013.2272212.

[6] Li, Su et al. "A Novel Approach Based on the Combination Image of Fraction Image and Normalized Mnf Image to Urban Land Use/Cover Mapping." IEEE International Geoscience and Remote Sensing Symposium (IGARSS), translated by Ieee, 2007

[7] Li, Xinwu et al. "Urban Land Cover Classification with High-Resolution Polarimetric Sar nterferometric Data." Canadian Journal of Remote Sensing, vol. 36, no. 3, 2010, pp. 236-47, doi:10.5589/m10-046.

[8] Lu, Dengsheng et al. "Land Cover Classification in a Complex Urban-Rural Landscape with Quickbird Imagery." Photogrammetric Engineering and Remote Sensing, vol. 76, no. 10, 2010, pp. 1159-68, doi:10.14358/pers.76.10.1159.

[9]Novack, T. and H. J. H. Kux. "Urban Land Cover and Land Use Classification of an Informal settlement Area Using the Open-Source Knowledge-Based System Interimage." Journal of Spatial Science, vol. 55, no. 1, 2010, pp. 23-41, doi:10.1080/14498596.2010.487640.

[10]Qiu, Xiaomin et al. "Incorporating Road and Parcel Data for Object-Based Classification of Detailed Urban Land Covers from Naip Images." Giscience & Remote Sensing, vol. 51, no. 5, 2014, pp. 498-520, doi:10.1080/15481603.2014.963982.

[11]Zheng, Gan et al. A Random Forest Based Method for Urban Object Classification Using Lidar Data and Aerial Imagery. 2015. 2015 23rd International Conference on Geoinformatics. Proceedings.

[12]Wentz, Elizabeth A. et al. "Land Use and Land Cover Mapping from Diverse Data Sources for an and Urban Environments." Computers Environment and Urban Systems, vol. 30, no. 3, 2006, pp. 320-46, doi:10.1016/j.compenvurbsys.2004.07.002.

[13] https://scholar.google.com/citations?user=xh9HeqgAAAAJ&hl=en

[14] https://scholar.google.com/citations?user=n8yVDWQAAAAJ&hl=en

[15] https://scholar.google.com/citations?user=MOfCYLwAAAAJ&hl=en