

**Classifying Building Usages: A Machine Learning Approach on Building
Extractions**

A Thesis Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Samantha Eleanor Duchscherer

May 2018

ABSTRACT

This paper considers methods to infer building usage from the geographic and geometric spatial distribution of building extractions. Focusing on Knox County, TN, a Random Forest (RF) and Support Vector Machine (SVM) were used to classify a polygonized building map developed from a Convolutional Neural Network (CNN) based upon remote sensing imagery. The resulting classification metrics of nine building usages are then compared to the RF and SVM building usage classification of Knox County's LiDAR building footprints and CNN building extractions with removal of false positives. It is shown that the raw CNN building extractions have acceptable building usage classification accuracies. This result is a useful addition to our understanding of building usage because the best remote sensing data (LiDAR building footprints) are not always accessible and completing tedious editing work (CNN building extractions with removal of false positives) is not feasible. Using the methods developed here, the effect of increasing CNN building detection training data for Knox County for testing on Knox County is also investigated. This case study assists in the process of examining if training a model on all Knox County CNN building detections can classify building usages in the similar urban-rural geographic location of Hamilton County, TN. ArcMap and R programming are utilized in gathering the data to conduct the machine learning algorithms while the building usage is defined by CoreLogic Parcel Land - Use codes.

Keywords: Building Usage, Land - Use, building extractions, building footprints, supervised learning, classification, Convolutional Neural Networks (CNN), remote sensing imagery, CoreLogic, LiDAR, Random Forest (RF), and Support Vector Machine (SVM)

TABLE OF CONTENTS

1. INTRODUCTION	1
2. DATA AND METHODS	2
2.1 Overview	2
2.2 County Data.....	2
2.3 Datasets	3
2.4 Models.....	5
2.4.1 Random Forest (RF)	6
2.4.2 Support Vector Machine (SVM)	7
2.5 Specifics	9
2.5.1 Stratified k-Fold Cross - Validation	9
2.5.2 Features Selection.....	10
2.5.3 Imbalanced Data	13
2.6 Summary	14
3. RESULTS	15
3.1 Selected Features and Hyperparameter(s).....	17
3.2 Classifying LiDAR Building Footprints, Edited CNN, and Raw CNN in Knox County ...	18
3.3 Altering Training Data in Knox County and Testing in Hamilton County.....	23
4. DISCUSSION	27
LIST OF REFERENCES.....	29
APPENDIX.....	35
APPENDIX A	36
APPENDIX B	43
VITA.....	46

LIST OF TABLES

Table 1. Comparing Knox and Hamilton County total area, perimeter, and population.	3
Table 2. Knox and Hamilton County Parcel Land - Use code breakdowns.	3
Table 3. Knox County Land - Use breakdown for building footprints (LiDAR) and building detections (edited and raw).	4
Table 4. The 21 features used in learning algorithms.	10
Table 5. The performance measures used for this study.	15
Table 6. Knox County's raw CNN building extractions residential category.	16
Table 7. Selected features and hyperparameter(s) for the different datasets and algorithms.	19
Table 8. LiDAR derived building footprints (top) and raw CNN building extractions (bottom) confusion matrices calculated from RUS - RF. Percentages were calculated from each cell count divided by its total number of actual values (corresponding row sum).	21
Table 9. Confusion matrix for training RUS - RF with all of Knox and testing on Hamilton County. Percentages were calculated from each cell count divided by its total number of actual values (corresponding row sum).	25
Table A - 1. Knox County specific Land - Use codes for LiDAR building footprints, edited building detections (only the building detections that intersect a building footprint within a three - meter buffer), and raw building detections (all CNN building detections).	36
Table A - 2. Frequencies of Knox and Hamilton raw CNN building detection for specific Land - Use categories.	37
Table A - 3. Knox and Hamilton County building usage accuracies referring to the percent of raw CNN building extraction that intersect building footprints within a three-meter buffer.	38
Table A - 4. Comparing LiDAR building footprints, edited building detections (only the building detections that intersect a building footprint within a three-meter buffer), and raw building detections (all CNN building detections) with models trained on around 20% of Knox County and testing on the remainder of Knox County.	39
Table A - 5. Comparing metrics with of RF and RUS - RF with increasing Knox County's training data by 5%, 20%, and 50%.	40
Table A - 6. Comparing the metrics of training RF and RUS - RF on all Knox County and testing in Hamilton County.	41
Table A - 7. Comparing classification metrics of the nine - multiclass problem with removal of 80% of the residential class to the binary problem consisting of only residential and other (with no RUS). Results shown are calculated with a RF consisting of $m_{try} = 2$ and the following variables: total area of all building detections in a parcel, nearest building detection distance, total perimeter of all building detections in a parcel, and number of building detections in a parcel.	42

LIST OF FIGURES

Figure 1. Knox (top right) and Hamilton (bottom left) County.....	2
Figure 2. CoreLogic Parcel data (blue) and CNN output for single family resident building detections (red) in Knox County, TN.	4
Figure 3. Comparing a sample of LiDAR building footprint and raw CNN building detection residential areas m ²	5
Figure 4. Support vector classifier with linear decision boundary.	8
Figure 5. Some LiDAR building footprint (pale yellow) and CNN building detections (red) fall on two single family resident parcels (blue)	11
Figure 6. Comparing RF - RFE results for raw CNN and raw CNN with RUS.....	17
Figure 7. RF – RFE output for the LiDAR dataset with RUS.	18
Figure 8. F - score (%) on the nine building usages categories for the three datasets (LiDAR, edited CNN, and raw CNN) and four algorithms (RF, RUS - RF, SVM, RUS - SVM).	20
Figure 9. Commercial (top), recreational (middle) and industry (bottom) building usages.	22
Figure 10. Comparing the metrics of the four algorithms for the raw CNN building detections.	24
Figure 11. Comparing F - scores of RF and RUS - RF when implementing different training data.	25
Figure 12. Comparing the metrics of training RF and RF - RUS on all of Knox County to test on Hamilton County.....	26
Figure 13. Comparing the nearest building distance m ² predictor variable for Knox and Hamilton County.....	26

LIST OF ALGORITHMS

Algorithm 1. Random Forest.	6
Algorithm 2. Recursive Feature Elimination (RFE) with resampling pertaining to the resampling technique and subset size chosen in this case study.	12

1. INTRODUCTION

There has been consistent improvement in the use of remote sensing imagery to establish automated, up-to-date, and large-scale building maps (Li, Femiani, Xu, Zhang, & Wonka, 2015; Ngo, Mazet, Collet, & Fraipont, 2017; Rottensteiner & Briese, 2002; Yuan, 2016). Thus, it is now more feasible to produce a timely and reliable distinction in building usage across regions. Differentiating between building usages assists with environmental concerns, urban planning, population distribution modeling, and socio-economics studies (Jensen & Cowen, 2011; Xian, 2015; Zribi, 2016). Although there is a need for classifying building usages, there has been relatively little research on how to effectively implement the new established building maps to make this classification. Steiniger et al. (2008) suggested that a non-linear Support Vector Machine (SVM) classification problem based only on the characterization of building geometries and morphological measures has potential for classifying urban building structures, though that research used topographic data. Hecht et al. (2015) also used topographic databases to classifying different residential and nonresidential building footprints with a Random Forest (RF). Recently, remote sensing imagery has been implemented to develop a binary RF method that classifies regular and irregular residential settlement types (Jochem, Bird, & Tatem, 2018), but only residential data were used in training and testing the RF. This results in the extra step or dilemma of obtaining some prior knowledge for distinguishing residential and nonresidential before performing the building usage classification process.

The research reported here explores a more automated classification scenario. The classes in the training and testing datasets consist of all existing building usage categories. The building usage categories are defined by Core Logic Parcel Land - Use codes ("CoreLogic | Our Company," n.d.) grouped into nine broad categories: residential (100 -199), commercial (200 – 299), industrial (300 – 399), vacant (400 – 499), agriculture (500 – 599), public (600 – 699), recreational (700 – 799), transport (800 – 899), and NA - misc (" " and 999). By first focusing on training and testing in Knox County, TN, a Random Forest and Support Vector Machine are implemented to classify the nine unknown building usages for the three following building maps based on remote sensing imagery: Knox County's LiDAR derived building footprints, Convolutional Neural Network (CNN) building extractions with removal of false positives, and raw CNN building extractions. Classifying building usages when only provided the raw CNN building extractions is shown to have comparable classification accuracy as LiDAR (the best remote sensing building footprint) and the building extractions with removal of false positives (the better choice of the remote sensing building extractions). With validation for continuing with the raw CNN building extractions, the next task is exploring if training a model on 5%, 20% or 50% of Knox County significantly changes the classification outcome in the building usage categories. Therefore, when there is success in classifying building usage in Knox County with using only a portion of the raw CNN output there might arise a scalable objective. This paper investigates if the raw CNN output for all Knox County can train an algorithm (RF and/or SVM) to classify building usage in the urban-rural geographic location of Hamilton County, TN, which appears initially to have similar spatial building usage characteristics.

2. DATA AND METHODS

2.1 Overview

This chapter provides an overview of the data pertinent to the three objectives of comparing the building usage classification outcomes of the different datasets in Knox County, assesses the amount of Knox County's training data needed for each building usage category, and examines if training a model on all of Knox County raw CNN building detections can classify building usages in Hamilton County, TN. The methods used to meet the objective of implementing Random Forest (RF) and Support Vector Machine (SVM) are also described. Section 2.2 discusses why Knox and Hamilton County were selected and describes the parcel data found in both counties. An overview of LiDAR building footprints and the CNN building detections is found in Section 2.3. Section 2.4 gives a literature review and summary on RF and SVM while Section 2.5 unfolds the details of cross-validation, feature selection, and imbalance data that are encompassed in executing a RF and SVM.

2.2 County Data

Finding building usage for any building in the United States, more specifically in Tennessee, is not difficult due to the availability of information. However, this study was planned to be conducted in an environment for which all the information is known in order to validate any results. This project started as exploratory research so selecting familiar areas, Knox and Hamilton County, made it easier to sift through different locations and building usages to compare LiDAR or building detections to reality. More importantly, selecting locations that appear to have similar spatial building usage characteristics significantly helps in training a model on one area (Knox County) for classifying building usage on another area (Hamilton County). Figure 1 shows the initial overview of both county's size, proximity to each other, and the United States Census definition of urban (50,000 or more people) and rural (all population, housing, and territory not included within an urban area) areas.

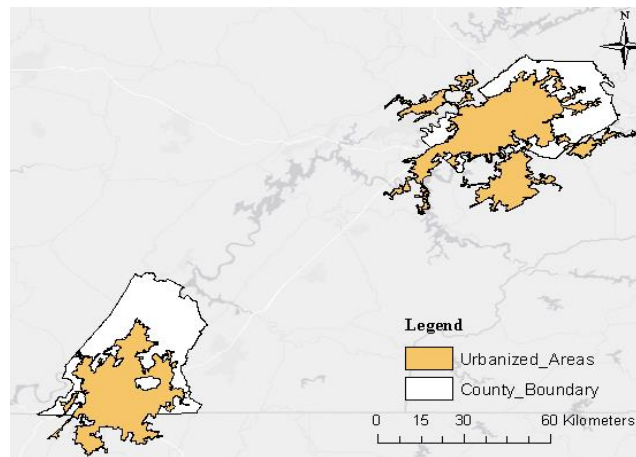


Figure 1. Knox (top right) and Hamilton (bottom left) County.

Associating these two Tennessee counties with numerical data, Table 1 gives the total US Census 2017 population, area, and perimeter (m²). Table 2 lists the breakdown of the parcel data

(Core Logic's Land - Use ID). Note, Table 2 groups the NA (" ") and Miscellaneous (999) category that occur as discrepancies when Core Logic nationalizes county parcel data.

Table 1. Comparing Knox and Hamilton County total area, perimeter, and population.

	Area m ²	Perimeter m ²	Population 2017
Knox County	136,362,665.90	226,811.95	461, 860
Hamilton County	149,135,952.36	198,371.48	361, 613

Table 2. Knox and Hamilton County Parcel Land - Use code breakdowns.

Land - Use Code	Description	Knox Freq	Knox %	Hamilton Freq	Hamilton %
100 - 199	Residential	154869	80.62%	120714	78.91%
200 - 299	Commercial	7448	3.88%	7754	5.07%
300 - 399	Industry	508	0.26%	772	0.50%
400 - 499	Vacant Land	21764	11.33%	20022	13.09%
500 - 599	Agricultural	2235	1.16%	6	0.00%
600 - 699	Public	1294	0.67%	628	0.41%
700 - 799	Recreational	222	0.12%	612	0.40%
800 - 899	Transport	424	0.22%	679	0.44%
" "-999	NA-Misc	3325	1.73%	1796	1.17%

While fully aware that the less populated Hamilton County is larger in size and rural area, overall the distributions of the parcel data in both counties are similar. It is important to clarify that only the distributions of the Land - Use parcel boundaries are compared here and not the building detections in the parcel data. When conducting a supervised learning classification, the building detections in the testing dataset are unknown, and thus comparing building detections before selection of a suitable test area is not an option. Showing similarity based on factors such as poverty, demographics, economy, culture, etc. will not be illustrated here because of the available parcel data, but such an examination should be conducted when information at the parcel level is not available.

2.3 Datasets

This research will compare building usage classification for three distinct datasets: LiDAR derived building footprints, an edited version of building extractions developed from a Convolutional Neural Network (CNN), and a raw version of CNN building extractions. LiDAR or light detection and ranging is a special case of remote sensing data that uses a laser to generating point cloud datasets, which are 3 - dimensional points that consist of elevation of the terrain and surface objects on the ground. The LiDAR derived building footprints in this study refer to the 2D footprints derived from 0.7 meter LiDAR point cloud data ("TNGIS - LiDAR," n.d.). Typically obtaining airborne LiDAR is more expensive and less accessible than imagery. Therefore, the Geographic Information Science and Technology (GIST) group at the Oak Ridge National Laboratory (ORNL) has developed up-to-date and large-scale building maps based solely on remote sensing imagery. More specifically, this method uses National Agriculture Imagery Program (NAIP) images to implement a Convolution Neural Network (CNN) framework resulting in an automatic extraction database of building footprints for the contiguous United States. Since NAIP imagery is acquired at a one-meter scale ("NAIP Imagery," n.d.), the

CNN polygonized building extractions are scaled at one-meter. Figure 2 provides an example of the CNN building extraction for a single family resident (SFR). For this study, the raw or unedited building extractions refer to the untouched CNN output. The edited or cleaner CNN dataset (CNN building extractions with removal of false positives or Type I errors) is defined as removing the building detections that do not intersect the closest form of ground truth, a LiDAR building footprint, within a three-meter buffer. Adding this three-meter buffer is a common technique to account for a minor shift in the datasets caused by temporal discrepancy: 2016 building footprints for Knox (“TNGIS - LiDAR,” n.d.), 2015 building footprints for Hamilton (“TNGIS - LiDAR,” n.d.), and the CNN algorithm used 2014 TN NAIP imagery (“NAIP Imagery,” n.d.).



Figure 2. CoreLogic Parcel data (blue) and CNN output for single family resident building detections (red) in Knox County, TN.

Table 3 provides the Knox County Land - Use breakdown for LiDAR building footprints, edited CNN, and raw CNN datasets in Knox County. Table A - 1 in the Appendix shows the more specific Land - Use frequencies (e.g. the different residential buildings in the broad residential category) of all three Knox County datasets and Table A - 2 in the Appendix shows the more specific Land - Use frequencies of the raw CNN dataset for both Knox and Hamilton County. It is expected that the raw CNN dataset shows more building detections than the raw CNN. This is due to the false positives in the raw CNN (see Table A - 3 in Appendix for the breakdown of the amount of raw CNN building detections that intersect building footprints). Unexpectedly, the edited CNN dataset also has more building detections than LiDAR for most of the building usages. Further investigation concludes that the CNN algorithm generates multiple building detections for one building.

The raw CNN also does not have the same building area coverage as LiDAR. Figure 3 shows a comparison of residential areas m^2 for the polygonized raw CNN detection and LiDAR building footprint. When comparing a sample of both datasets for the same parcels, there appears to be a large difference in building coverage (e.g. the minimum building extraction area for raw CNN is $0.65 m^2$ while the minimum area for LiDAR building footprint is $72 m^2$). This pattern pertains to all building usage categories and predictor variables, but the nonresidential building usages do have a larger gap between LiDAR building footprints and raw CNN building extractions. The

multiple building detections used for one building and the difference in area coverage initially raise concerns, but the results show these are key aspects that aid in classifying the raw CNN building detections.

Table 3. Knox County Land - Use breakdown for building footprints (LiDAR) and building detections (edited and raw).

Land - Use ID	Description	LiDAR	LiDAR %	Edited	Edited %	Raw	Raw %
100 - 199	Residential	151460	88.75%	148932	86.38%	185143	80.84%
200 - 299	Commercial	8236	4.83%	11634	6.75%	18096	7.90%
300 - 399	Industry	806	0.47%	1225	0.71%	2074	0.91%
400 - 499	Vacant Land	2239	1.31%	2396	1.39%	6423	2.80%
500 - 599	Agricultural	3878	2.27%	3729	2.16%	8976	3.92%
600 - 699	Public	1416	0.83%	2002	1.16%	3074	1.34%
700 - 799	Recreational	232	0.14%	307	0.18%	681	0.30%
800 - 899	Transport	334	0.20%	432	0.25%	950	0.41%
" " - 999	NA-Misc	2055	1.20%	1761	1.02%	3597	1.57%

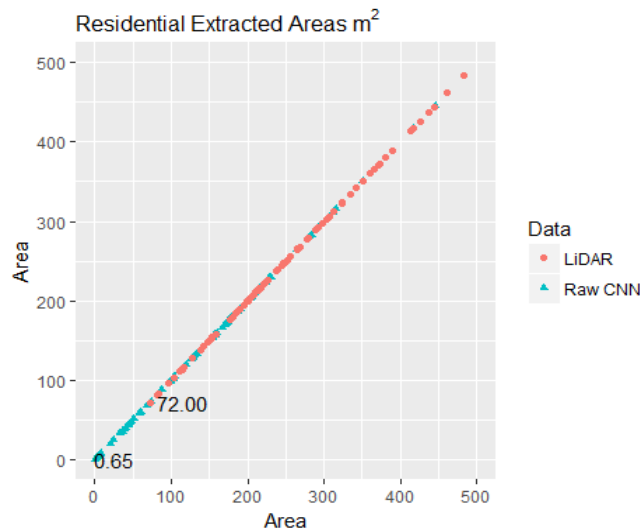


Figure 3. Comparing a sample of LiDAR building footprint and raw CNN building detection residential areas m^2 .

2.4 Models

This study concerns the situation when the classes (building usages) are known and the goal is to classify current and new observations into these classes with numerical predictor variables. This is considered a supervised learning classification problem, and methods under this machine learning umbrella include the Random Forest and Support Vector Machine learning algorithms. Comparison of at least two different algorithms is a standard practice because no single model will generally consistently work best for every problem, a phenomenon known as the “No Free Lunch” theorem (Wolpert & Macready, 1997). RF and SVM are shown to be among the top performing (Caruana & Niculescu-Mizil, 2006) and common supervised learning algorithms found in a variety of domains (Bégué et al., 2018; Granitto, Furlanello, Biasioli, & Gasperi,

2006; Han, Jiang, Zhao, Wang, & Yin, 2017; Hasan, Nasser, Pal, & Ahmad, 2014; Kumar & Thenmozhi, 2006; Ogutu, Piepho, & Schulz-Streeck, 2011; Qiu & Wang, 2017; Statnikov & Aliferis, 2007). These algorithms are overall easy to tune and can be used for both regression and classification. The R caret package (Kuhn, 2017) is utilized for implementing and comparing both algorithms.

2.4.1 Random Forest (RF)

To understand a Random Forest model, a simple classification decision tree needs to first be explained. A classification tree is composed of nodes and branches and the overall aim is to partition data into more homogenous groups. The tree starts with the root node and a nested if-then statement based on the Gini index (Breiman, Friedman, Stone, & Olshen, 1984) splits the root node into k internal node(s) where each of these contains a larger proportion of one class. The k internal independent node(s) then split with different if-else statements, and the process of increasing the tree depth continues until each terminal tree node or leaf has a class label, which for classification is determined by a majority vote of all the observations in that leaf (Kuhn & Johnson, 2013). Having one tree will often lead to high variation in classification results so building on this concept, a RF (Breiman, 2001) is an ensemble of independent classification trees. The concept of RF is represented in Algorithm 1 (Kuhn & Johnson, 2013).

Algorithm 1. Random Forest.

```

1  Select n trees to build
2  for i = 1 to n {
3      Generate a bootstrap sample (random samples with replacement) of the original data
4      Train a tree model on this sample
5      for each split {
6          If there are M total features, randomly select  $m \ll M$  features
7          Select the best predictors among the m variables and partition the data
8      } end loop
9      Tree is completed when stopping criteria is reached (each leaf has a majority vote)
10 } end loop
```

The algorithm for this study begins with selecting $n = 500$ bootstrap samples and an independent classification tree is built with each sample. If all features were used to create the trees, then this algorithm would be considered bagging classification trees. Bagging is the technique that uses bootstrap samples, random samples with replacement, and considers an aggregation of an ensemble of algorithms. Hence, bagging is also referred to bootstrap aggregating. Using all features in creating the bagged decision trees typically creates individual trees that are correlated (Kuhn & Johnson, 2013). Thus, random forest decorrelates the trees by using a random subset of features, which is always less than the total number of features, for each if-else statement split. Therefore, with the same subset of features at each node but randomizing the selection of which features are used, all 500 trees are fully grown until some stopping criteria is reached. This study's stopping criteria for an individual tree is kept at the R Caret default, where the label given to each terminal node has the majority vote of at least $1 - (1/\text{number of classes})$. To predict a new observation's class, the observation goes through each of the 500 trees in the forest and each tree votes for what class that new observation should belong to. The new observation is then classified as the class that had the most votes over all the trees in the forest. Random forest can also provide an overall prediction error for classifying an observation. With the use of bootstrap samples, each tree has its own set of observations from the original dataset that did not

appear in creating that particular tree. An out-of-bag (OOB) estimate is defined by the error of a classification tree predicting the observations that remained outside its bootstrap sample. The overall OOB training error reported in a RF averages the 500 OOB errors.

Choosing the random subset of features for developing the classification trees is defined through the tuning parameter, m_{try} . For a classification problem, Breiman suggest setting m_{try} to the square root of the number of predictors, but as in any machine learning algorithm tuning this hyperparameter based upon the provided data is almost always the better approach. Tuning m_{try} could range from one to one less than the total number of predictors. This case study estimated the m_{try} parameter using a method known as a grid search with cross-validation. Essentially, m_{try} goes through the values of 1 to 15 to see which m_{try} value generates a better model (see Section 2.5.2 for more details). While there are also other tuning parameters in a RF, only the m_{try} hyperparameter is tuned. This is because m_{try} has the most significant impact on results by allowing a RF to be more robust to outliers and correlation between the trees (Breiman, 2001). It is common to increase the tree amount to 1,000 but implementing this for the raw CNN dataset did not have a significant accuracy trade-off while doubling the computational time. Note that visualizing a single tree in the ensemble of 500 trees is not feasible with the provided datasets. For perspective, consider training a RF on the 20% of the raw CNN dataset. This nine - multiclass problem needs over 5,500 terminal nodes in each tree for each terminal node to have a label with the majority vote of $1 - (1/9) = 88.89\%$.

2.4.2 Support Vector Machine (SVM)

Although Vladimir Vapnik established Support Vector Machines in the mid-1960s, it wasn't until the 1990s that SVMs become the more flexible and effective algorithm for nonlinear classification problems (Kuhn & Johnson, 2013). Before discussing SVM, it will be helpful to initially consider a completely separable binary classification problem with a linear decision boundary consisting of p features or variables. A binary classification problem means there are two classes, $y = 1$ or $y = -1$. When there exists a completely separable linear decision boundary, a hyperplane (Equation 1) distinguishes the classes where all of one class ($y = 1$) is on one side of the hyperplane (Equation 2) and the other class ($y = -1$) is on the other side of the hyperplane (Equation 3) (James, Witten, Hastie, & Tibshirani, 2013). Classifying a new observation is essentially deciding which side of the hyperplane the observation falls.

$$\beta_0 + \beta_1 X_1 + \dots + \beta_p X_p = 0 \quad (1)$$

$$\text{If } y = 1 \text{ then } \beta_0 + \beta_1 X_1 + \dots + \beta_p X_p > 0 \quad (2)$$

$$\text{If } y = -1 \text{ then } \beta_0 + \beta_1 X_1 + \dots + \beta_p X_p < 0 \quad (3)$$

With completely separable data, there could essentially be infinite separating hyperplanes and the next task becomes finding the best separating hyperplane. Naturally, the maximal margin hyperplane is an optimal choice. This is the separating hyperplane with the largest margin, which is the smallest perpendicular distance from a given separating hyperplane to the closest data point. In this completely separable linear case, there will never be any observation in the margin. Now moving to a non-separable case, the maximal margin hyperplane is not possible, and for these cases the support vector classifier extends the concept of a separating hyperplane to create a hyperplane that almost separates the classes in the training data (James et al., 2013).

The support vector classifier or soft margin classifier allows for some error in classification. The support vectors can be on the wrong side of the margin or even on the wrong side of the hyperplane. Selecting the optimal hyperplane then becomes the hyperplane that maximizes the margin while minimizing the penalization caused from the support vectors. This optimization problem involves the tuning parameter C , which can be thought of as a budget for the cost of these allowed errors. A larger C means the model is more tolerant of violations implying a larger margin, a more complex model, and a less smooth decision boundary (James et al., 2013). Figure 4 from *An Introduction to Statistical Learning* is used to visual a support vector classifier when fit to a small dataset with two classes (James et al., 2013). In Figure 4, the solid line is the separating hyperplane and the margin is the dashed line. Observations 1, 2, 7, 8, 9, 11, and 12 are support vectors because they fall on the margin, inside the margin, and even on the wrong side of the hyperplane. The cost associated with each support vector varies on the placement of the hyperplane. In the example provided, the support vector 12 will have a highest cost value. Both 11 and 12 are on the wrong side of the hyperplane, but 12's perpendicular distance from the hyperplane is larger than 11. Also, support vector 1 will have a higher penalty than 2 because it is inside the margin while 2 is on the margin.

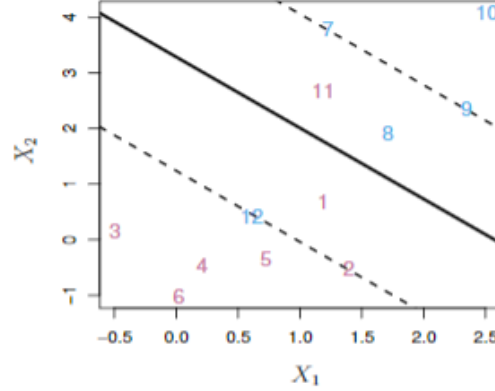


Figure 4. Support vector classifier with linear decision boundary.

A SVM uses the concepts from the support vector classifier but adds a kernel trick that allows for a nonlinear decision boundary. Essentially, a kernel maps the original data into a higher dimension feature space where the process of the linear support vector classifier is capable of being performed. This allows more flexibility in SVM because using the kernel representations of high dimensional feature mappings makes it possible for an efficient computation on a nonlinear decision boundary. This case study implements a SVM with a radial kernel, which can be thought of as a nonlinear support vector classifier (Equation 5) substituting in a Gaussian radial basis function (RBF) kernel (Equation 6). Three kernels were tested, and the radial kernel had better classification accuracies than both linear and polynomial kernels.

$$f(x) = \beta_0 + \sum_{i \in S} \alpha_i K(x, x_i) \quad (5)$$

$$K(x_i, x_{i'}) = \exp\left(-\gamma \sum_{j=1}^n (x_{ij} - x_{i'j})^2\right) \text{ where } \gamma > 0 \text{ and } j_1, \dots, j_n \in S \quad (6)$$

In the equations above, there are p variables and $\alpha_i, i = 1, \dots, p$ represents the training observation. For the α_i 's, Equation 5 only sums up the collection of the support vectors ($j_1, \dots, j_n$) because the α_i equals zero when the observation is not in the S collection. This is due

to there being zero cost related to a correctly classified observation. Also, considering only the S collection produces a more efficient equation with far fewer terms than p . The $K(x, x_i)$ is the radial kernel that signifies a generalization of the inner product between all pairs of training observations. In summary, to represent a nonlinear classifier $f(x)$ as well as computing its coefficients, only the inner product of all support vector pairs is needed. In a nonlinear classification problem, these needed inner products are generalized by a kernel (James et al., 2013).

As seen in Equation 6, using the radial kernel introduces another tuning parameter γ to the SVM algorithm. The hyperparameter γ inversely controls variance in the RBF kernel. Hence, tuning γ in SVM dictates how far a single training point influences the decision boundary. Intuitively, a high γ would entail a small variance so only a certain amount of nearby points would be considered in making the decision boundary resulting in a less smooth boundary. Just as in the RF section, tuning the hyperparameters SVM can make the algorithm more robust to outliers, but it is also very computationally expensive to have an extensive tuning grid search. This paper follows the suggestion of Hsu et al. (2003) and tunes C and γ using a grid search approach with stratified 5-fold cross-validation on the exponentially growing sequences of $(2^{-3}, 2^{-1}, 2^1, 2^3, 2^5, 2^7, 2^9)$ and $(2^{-6}, 2^{-4}, 2^{-2}, 2^0, 2^2, 2^4, 2^6)$ respectively (see Section 2.5.2 for more details). Also, recall that this case study is a multiclass classification problem, and thus a one - vs - all SVM approach is implemented. With $K = 9$ classes, K SVMs are fit where each time one of all the K classes is compared to the remaining $K - 1$ classes (Kuhn & Johnson, 2013).

2.5 Specifics

2.5.1 Stratified k-Fold Cross - Validation

The stratified k - fold cross - validation (CV) resampling technique is used in aiding training data for model selection, which is evaluating the performance from using different features and tuning parameters. Implementing a regular k -fold CV involves randomly dividing the training data into k equal subsets known as folds. However, this case study has an imbalance in the response variables since most of the building usages are residential. Using a normal k -fold CV on this imbalanced dataset would most likely create folds only consisting of residential building usage, which results in a biased trained model. Implementing a stratified k -fold CV approach ensures the distribution of all classes has the same representation of the initial data for each fold. The stratified rotation process proceeds by taking the equal folds to train and validate a model k times where each fold represents a validation set pertaining to a $k - 1$ trained model. The optimal model, with a corresponding subset of features and tuning parameter(s), is the $k - 1$ model that has the lowest associating validation error (Olson & Delen, 2008). In summary, a stratified k - fold CV is implemented by first isolating the training and testing dataset. The stratified k - fold CV is only conducted on the training dataset. Estimating the optimal subset of features and tuning parameter(s) is calculated inside the stratified k - fold CV process. The final trained RF and SVM models hold the features and tuning parameters(s) constant and run the algorithms with the full training data. Typically, $k = 10$ is a more reliable choice for reducing the variance and bias in the overall model (Kohavi, 1995; Kuhn & Johnson, 2013), but $k = 10$ was not practical in this case study due to the amount of data and algorithm runs. Although a stratified 5 - fold CV generally has less variance and bias than a regular 5 - fold CV (Kohavi, 1995), selecting $k = 5$ rather than $k = 10$ is considered a limitation.

2.5.2 Features Selection

It is important to reemphasize that the selected features are only the features that are obtainable from the CNN building detections. This reasoning is due to the hopes of implementing the methodology studies here in locations for which CNN building detections are the only available data. Although a feature such as a building's height would be extremely helpful to classify building usages, this study considers height to be an unattainable feature. ArcMap and R are used to obtain 21 geographic and geometric spatial distribution features that are believed to be helpful for this classification problem. These 21 features are gathered separately for the three datasets, meaning LiDAR, edited CNN, and raw CNN will have different values for each feature. For simplicity of explanation, consider only gathering the 21 features for the raw CNN building detection (Table 4). The building detections are first projected to their corresponding WGS_1984_UTM Zone. The building detections' inside centroids are then spatially joined with the Core Logic Parcel data to give each building detection a Land - Use code. This results in the features being calculated based on the building detections' centroids. The 21 features in Table 4 are calculated for all building detection in the raw CNN dataset, thus the final raw CNN dataset consists of the total number of building detections as the rows and the 21 features as the columns. The LiDAR and edited CNN will have the same matrix format, but again because the datasets consist of different and fewer polygons the values and row count are different.

Table 4. The 21 features used in learning algorithms.

Description	Code Name
Total area of all building detections in parcel m ²	p_bd_area
Total perimeter of all building detections in parcel m	p_bd_peri
Area of individual building detections m ²	s_bd_area
Perimeter of individual building detections m	s_bd_peri
Number of building detections within parcel	num_bd_in_parcel
The distance of the nearest building detection centroid m	nearestbddist
Number of building detection centroids in 20 m radius	num_bd_within20
Median perimeter of all the building detection centroids found in 20 m radius	medianperi20
Median area of all the building detection centroids found in 20 m radius	medianarea20
Number of building detection centroids in 40 m radius	num_bd_within40
Median perimeter of all the building detection centroids found in 40 m radius	medianperi40
Median area of all the building detection centroids found in 40 m radius	medianarea40
Number of building detection centroids in 60 m radius	num_bd_within60
Median perimeter of all the building detection centroids found in 60 m radius	medianperi60
Median area of all the building detection centroids found in 60 m radius	medianarea60
Number of building detection centroids in 80 m radius	num_bd_within80
Median perimeter of all the building detection centroids found in 80 m radius	medianperi80
Median area of all the building detection centroids found in 80 m radius	medianarea80
Number of building detection centroids in 100 m radius	num_bd_within100
Median perimeter of all the building detection centroids found in 100 m radius	medianperi100
Median area of all the building detection centroids found in 100 m radius	medianarea100

Deciding to spatially join the centroid to Core Logic Parcel data arose as a matter of discussion. Unfortunately, this causes data to be lost. Some centroids do not fall in a parcel, implying there are no existing Land - Use codes. Also, this method does not split the building detections that are located on multiple parcels, implying only one Land - Use code is given to a detection that

technically has more than one Land - Use code (e.g. consider shopping centers with different store building usages inside one building). However, working with the centroids is still a practicable approach because when there are multiple parcel Land - Use codes for one building detection they tend to fall in the same broad Land - Use categories defined by Table 2 (see Section 2.2). So, consider the shopping center example, all the Land - Use codes might be different for that one building extraction, but they would still all fall in the same classification group of the commercial category consisting of Land - Use codes 200 – 299.

An alternative method of clipping building detections to parcel data was considered. This approach would consider the different building usages found in one detection but implementing this technique in a location without parcel data would be extremely difficult. Also, there are many cases for which the actual building doesn't land on two parcels, but shadows and shifts found in the imagery make a small portion of a footprint or extraction land on a nearby parcel. Figure 5 shows a residential example. It was found that this method would increase the amount of building detections tremendously and develop too many individual areas and perimeters (m²) with miniscule values that would greatly alter the classification groups.

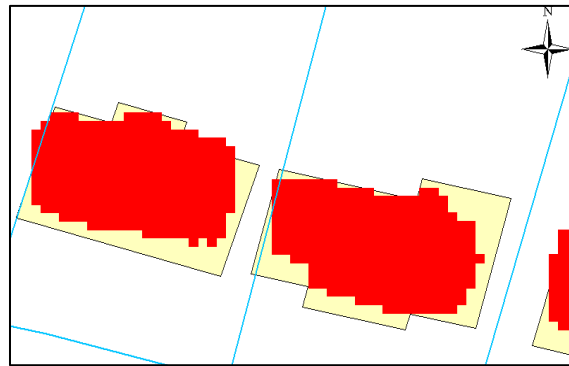


Figure 5. Some LiDAR building footprint (pale yellow) and CNN building detections (red) fall on two single family resident parcels (blue).

Considering 21 features in any machine learning algorithm is a substantial amount for which the curse of dimensionality becomes a concern (Xingquan, 2007). This makes any type of feature selection approach essential (Guyon & Elisseeff, 2003). The Recursive Feature Elimination (RFE) with resampling found in the R caret package (Kuhn, 2017) is the feature selection method used in this study. The RFE method implemented and shown in Algorithm 2 (Kuhn & Johnson, 2013) is a modification of the original RFE algorithm, which was actually introduced with the SVM algorithm (Guyon, Weston, Barnhill, & Vapnik, 2002). It was found that conducting an “outer” resampling loop on the original RFE process tremendously reduced the chance of overfitting the predictors because variation in the feature selection process can now be captured (Ambroise & McLachlan, 2002). Overall, this algorithm is not affected by correlation of predictor variables when preformed with both a RF (Gregorutti, Michel, & Saint-Pierre, 2017) and SVM (Xie, Hu, & Yu, 2006) algorithm. When the RFE process is conducted with these two algorithms, the notated is RF - RFE and SVM - RFE.

As stated in Section 2.5.1, this research uses a stratified 5 – fold CV meaning the rotation process will implement 5 iterations where each fold acts as a validation set. The first step in Algorithm 2

is to select one of the iterations in the stratified 5 – fold CV process. When isolating this iteration, the training data are split into a training set ($k - 1$ folds or $4/5$ training data) and validation set (1 fold or $1/5$ training data). Tuning and training the $k - 1$ folds with all 21 predictors is the next step. Recall that in this study RF tries m_{try} values between 1 to 15 and SVM tunes C and γ with $(2^{-3}, 2^{-1}, 2^1, 2^3, 2^5, 2^7, 2^9)$ and $(2^{-6}, 2^{-4}, 2^{-2}, 2^0, 2^2, 2^4, 2^6)$ values respectively. So, steps 3 - 5 in Algorithm 2 goes through all the hyperparameter(s) values in this set grid. RF has a grid length of 15 and SVM has a grid length of 47. The $k - 1$ folds now has a validation error for each hyperparameter(s) in the set grid. Thus, tuning the $k - 1$ model in the first iteration is selecting the hyperparameter(s) that produces lowest validation error for that k^{th} fold. Ranking all 21 variables based on their importance to the best tuned model is conducted next, and the training and tuning process is repeated but the step of recursively considering a smaller subset size of features is added. Finally, since this is a stratified 5 – fold CV, there are 5 validation errors associated with an optimal subset of variables and hyperparameter(s) and the last step is to select the $k - 1$ model that has the overall lowest validation error. Note, when considering a smaller subset size for RF (line 11 in Algorithm 2), the grid from 1 to 15 will change into 1 to S_i . A RF max m_{try} value is one less the total number of predictor values, but the R package will allow for m_{try} to equal the number of predictor variables. When the m_{try} variable does equal the total number of predictor variables, this implies that the tuning process is considering bagged decision trees.

Algorithm 2. Recursive Feature Elimination (RFE) with resampling pertaining to the resampling technique (stratified 5 – fold CV) and subset size chosen in this case study.

```

1  for  $i = 1$  to 5 {
2    Partition training data into training ( $4/5$  of data) and validation ( $1/5$  of data) sets
3    for  $i = 1$  to grid length of hyperparameter(s) {
4      Train a model using all predictors and the  $i^{th}$  hyperparameter(s)
5      Predict the validation set
6    } end loop
7    Select the model with hyperparameter(s) that has the lowest validation error
8    Calculate the variable ranking based on its importance to the model
9    for each subset size  $S_i$ ,  $i = 21, 20, 18, 16, 14, 12, 10, 8, 6, 4, 2$  {
10     for  $i = 1$  to grid length of hyperparameter (s) {
11       Train a model using  $S_i$  most important variables and the  $i^{th}$  hyperparameter(s)
12       Predict the validation set
13     } end loop
14     Select the model with  $S_i$  parameters and corresponding hyperparameter(s) that has the lowest
        validation error
15   } end loop
16 } end loop
17 Select the model with  $S_i$  and corresponding hyperparameter(s) that has the overall lowest validation error
    between the 5 iterations

```

Tuning the hyperparameter(s) and conducting feature selection inside a stratified $k -$ fold CV helps optimize an algorithm's performance but is very computationally expensive. Not only is selecting $k = 5$ in the process a limitation, but the tuning grid and selected subset sizes are also limitations. Again, with 21 variables, RF can have the m_{try} value between 1 to 20. However, setting the m_{try} grid from 1 to 20 resulted in an extensive amount of computation time, and thus the m_{try} grid was scaled down to only implement values between 1 and 15. The same dilemma was found in SVM when trying to expand the grid to account for more values for C and γ . The subset sizes selected in Algorithm 2 is also considered a limitation in this study because ideally

going through every predictor variable (1 to 21) is preferred. Yet, this is not practical with the amount of data presented.

2.5.3 Imbalanced Data

Conducting a RF and SVM on the data found in this case study gives back an overall approximate accuracy of 90%, resulting in the initial implication that these algorithms are fantastic for correctly classify building usages. It turns out however that the residential class takes up around 80% of the data and that overall accuracy only pertains to this class. This would be fine if the long-term users of this research agree that only classifying residential is needed and all other building usage categories are irrelevant. The 90% accuracy model would appear to still be a usable model because even though that accuracy only pertains to residential that is now the only class of interest. However, this is still an incorrect assumption made by an imbalanced dataset leading to the accuracy paradox (Xingquan, 2007).

By definition, an imbalanced dataset is where one or more classes have very low proportions compared to other classes (Kuhn & Johnson, 2013), and most classification problems are sensitive to this issue. A RF will create biased classification trees towards the majority class while a SVM produces skewed separating hyperplanes towards the minority classes. Both cases result in a low true negative rate or specificity (Kuhn & Johnson, 2013). When there is a low specificity, the proportion of negatives found in each class were not identified as being negative. Consider the case of only classifying residential buildings. The imbalanced data results in the residential category to have a high accuracy but low specificity. The imbalanced data trains the models to essentially classify everything as residential, and since residential contains most of the data of course the model will have a high overall accuracy.

Since many real-world datasets are imbalanced, many approaches have been developed to combat the issue (H. He & Garcia, 2009; Haibo He & Ma, 2013). Literature shows mixed results on the optimal solution, and so three methods considered here were the resampling method called SMOTE (Chawla, Bowyer, Hall, & Kegelmeyer, 2002), weighting the minority classes higher within the algorithm, and random under - sampling (RUS) the majority class. Synthetic Minority Over-Sampling Technique or SMOTE is a hybrid approach of random over - sampling and under - sampling. Rather than creating copies of the same observations, as in random over - sampling, SMOTE creates synthetic observations. Then with the minority classes having a larger sample size, the random under - sampling of the majority class will remove less, potentially useful, information. In this study, SMOTE immensely helped the specificity but significantly lowered the other metrics (precision, recall, and F-score).

Weighting the classes in an algorithm can adjust the importance of classes. It was found that increasing the minority classes' weights by ten, which implies the nonresidential classes would be 10 times more important in the RF and SVM models, had little impact on all the metrics. Due to this being a nine - multiclass problem, RUS outperformed both SMOTE and the technique of adjusting the class weights. Therefore, randomly under -sampling the majority class is essentially a guess and check method. There is potential for discarding observations that could have important information or make variables look like they have a higher variance (Haibo He & Ma, 2013). Randomly discarding 70%, 80%, and 90% of the residential class in training were all considered, but 80% had the better tradeoff in dramatically improving residential's specificity

while having little change on bias and variance in the corresponding test dataset. Randomly discarding 80% of residential and then conducting RF and SVM is notated in this paper as RUS - RF and RUS - SVM.

2.6 Summary

Knox and Hamilton County were selected for this study due to their similarities in size, coverage of rural and urban area, and parcel Land - Use breakdown. Thus, the input data pertinent to the three objectives are Knox County's building footprints and the CNN building detections that intersected the building footprint within a three-meter buffer as well as Knox and Hamilton raw CNN building detections. RF and SVM are the algorithms chosen for this classification problem because they are relatively understandable, simple (e.g. few tuning parameters) to implement, and more importantly they are among the top performing algorithms found in the literature. Before conducting a RF and SVM, feature selection and tuning hyperparameter(s) are vital because the initial features that are believed to be useful and the default hyperparameter(s) typically do not produce the best performance for either model. Therefore, this study uses the RFE technique (RF - RFE and SVM - RFE) with a stratified 5 - fold CV on the training datasets. For each fold, the RFE algorithm is recursively considering smaller and smaller sets of features while simultaneously using a grid search approach to tune the hyperparameter(s). This helps with model selection as well as reduces the chance of overfitting on the test dataset.

Once obtaining the optimal features and hyperparameter(s), the final RF and SVM holds both constant and conducts their algorithm with the full training data. Addressing the imbalanced data issue also helps a model's accuracy. Due to fact that the residential classification accounts for around 80% in Knox and Hamilton County, it is expected that the original data produce an algorithm with higher accuracy. However, essentially this model is classifying everything as residential and almost nothing as being nonresidential, and this can be validated by having a low specificity value (see Section 2.5.3 for definition of specificity and other metrics). RUS for 80% of the residential class in the training datasets is implemented in this study to combat this imbalanced data problem.

3. RESULTS

The long - term users of this research have deemed that 75% to 80% correctness in classifying building usage (residential, commercial, industry, vacant, agriculture, public, recreational, transport, and NA/misc) is an acceptable result. An aggregated classification metric of the overall model, such as accuracy or Kappa, will not be reported in this research. The goal is to see how each class did individually. Also, it is expected that the NA - misc class will have low classification results because this is CoreLogic's random collection of Land – Use codes. The highest combination of Table 5 metrics are used to compare, assess, and determine the optimal model. Note, that this table will use the following notation: TP = True Positive, TN = True Negative, FP = False Positive, and FN = False Negative.

Table 5. The performance measures used for this study.

Metric	Interpretation (e.g. residential building usage)	Formula
Precision/Positive Predictive Value (PPV)	For all buildings detections that were labeled as residential, how many actually are residential?	$\frac{TP}{(TP + FP)}$
Recall/Sensitivity/True Positive Rate (TPR)	Of all the building detections that truly are residential, how many did this model label?	$\frac{TP}{(TP + FN)}$
F – score	Harmonic average of precision and recall. This metric takes both FP and FN into account.	$\frac{2 * precision * recall}{precision + recall}$
Specificity	The fraction of buildings that are not residential that are correctly identified as negative by the model.	$\frac{TN}{(TN + FP)}$

This research was performed on a Dell computer equipped with an Intel Xeon processor E5 - 1650 v3 that has 32.0 GB of RAM running a 64 - bit version of Windows 7. The following recaps the three objects or results that are investigated in this study:

1. Are the RF and SVM accuracy of classifying building usages with the raw CNN building extractions comparable to classifying building usages using the best available data (LiDAR) or the edited CNN building extractions (building extractions with removal of false positives)?
2. Do different amounts of training data for the raw CNN building extractions in Knox County (5%, 20%, and 50%) have a significant impact on the building usage classification values in Knox County?
3. Can the raw CNN output for Knox County train an algorithm to predict the classification of building usage in the similar geographic location of Hamilton County?

Before conducting any analysis, all the features were normalized to have values between 0 and 1. Also, the amount of training data selected from each building usage category for the first objective is a tradeoff combination of obtaining a suitable sample size for each category but yet not requiring too much computation time for comparing different algorithms and different datasets. Continuing forward, the data science approach for objective one is first randomly selecting 20% from each building usage category of Knox County's raw CNN building extractions. Table 6 shows the residential breakdown of the 20% sample from Knox County's raw CNN building extractions. The selected 20% and RUS process considers as much as possible of the individual building usage categories found in the broad categories. In the

residential category, this ensures that single family residence (SFR) is not the only representation of the broad residential category.

Table 6. Knox County's raw CNN building extractions residential category.

Land - Use Code	Description	Total	Selected	Percentage Selected
100 - 199	RESIDENTIAL	185143	37029	20.00%
100	RESIDENTIAL(NEC)	9080	1842	20.29%
102	TOWNHOUSE/ROWHOUSE	138	30	21.74%
106	APARTMENT	176	34	19.32%
112	CONDOMINIUM	1337	265	19.82%
114	COMMONAREA	24	3	12.50%
116	MIDRISECONDO	52	12	23.08%
118	FRAT/SORORITYHOUSE	1	0	0.00%
131	MULTIFAMILY10UNITSPLUS	3272	631	19.28%
132	MULTIFAMILY10UNITSLESS	427	83	19.44%
133	MULTIFAMILYDWELLING	3017	608	20.15%
136	MOBILEHOMEPARK	2292	457	19.94%
137	MOBILEHOME	41	8	19.51%
138	MANUFACTUREDHOME	251	53	21.12%
142	MOTEL	47	15	31.91%
148	PUD	1673	344	20.56%
155	GROUPQUARTERS	252	60	23.81%
163	SFR	162664	32516	19.99%
164	TRANSIENTLODGING	399	68	17.04%

The Knox County's LiDAR building footprints and edited CNN building extractions training data are selected next. This selection uses the same Parcel IDs in the 20% raw CNN building extractions implying that the training data used for the LiDAR building footprints and the edited CNN building extractions is not necessarily 20% of its own dataset. However, this gives a comparison of the same locations or Parcel IDs, and again considers as much as possible of the individual Land - Use codes for all three datasets.

The RFE algorithm with a stratified 5 - fold CV and a grid search tactic of finding the optimal hyperparameter(s) is conducted on each of the 20% selected datasets: building footprints, CNN building extractions with removal of false positives, and the raw CNN building extractions. This part takes an extensive amount of time. RF - RFE and RF - RFE on one dataset can take between 2 to 3 days respectively. However, the next step of conducting the RF and SVM algorithms, with the best subset of features and optimal hyperparameter(s) held constant, and then predicting the classification of the corresponding test set takes minutes. Appendix B shows the code that implements the R caret functions to preform RF - RFE on the raw CNN building detections. The code also shows how the RF - RFE results are used to predict the test dataset with the RF algorithm having the best subset of features and optimal m_{try} parameter held constant. The code involving SVM calculations is almost identical.

RUS 80% of residential is separately conducted on the three training datasets and the same process is repeated. The RUS process follows the same process of selecting the 20% by eliminating as much as possible of the individual building usage categories found in the broad categories. Note, by the design of stratified k - fold CV, conducting RUS outside or inside this CV process gives the same output. This results in a total of six models that are analyzed. Showing that the raw CNN building extractions can comparably train a model to classify building usages, gives motivation for investigating objective two. Hence, the better performing algorithm (RF original, RUS - RF, SVM original, or RUS - SVM) on the 20% raw CNN building extractions in Knox County is selected and compared with training that algorithm on 5% and 50% of Knox County. Finally, with promising results from objective two, that same model is used to train all of Knox County's CNN building extractions to hopefully classify building usage in Hamilton County, TN.

3.1 Selected Features and Hyperparameter(s)

Recall that the stratified 5 - fold CV process was not repeated in this study, and that CV is only validating outcomes based on the provided training dataset. Hence, the RFE results must be cautiously analyzed because initial results show very different optimal feature subsets and tuning parameter(s) for the various datasets. For example, performing RF - RFE on 20% of the raw CNN and then on 20% of the raw CNN with removing 80% of the residential class gives two different solutions, and as it turns out this is not from the imbalanced data. Recall that the stratified k - fold CV process in RFE gives a validation accuracy and tuning parameter (s) for each subset size. Thus, examining the validation accuracy of the different variable subset sizes in the optimal k - 1 fold model, the raw CNN (RUS residential) with 21 features and $m_{try} = 8$ gives very comparable results to the raw CNN with four features and $m_{try} = 2$ (Figure 6). When having very equivalent classification training accuracy, a common machine learning approach is to select the more simplistic model, which is the model with less predictor variables, so that the chance of overfitting in the test dataset is reduced. More significantly, the optimal RF - RFE feature output when using 50% of the raw RUS CNN dataset resulted in the same four variables.

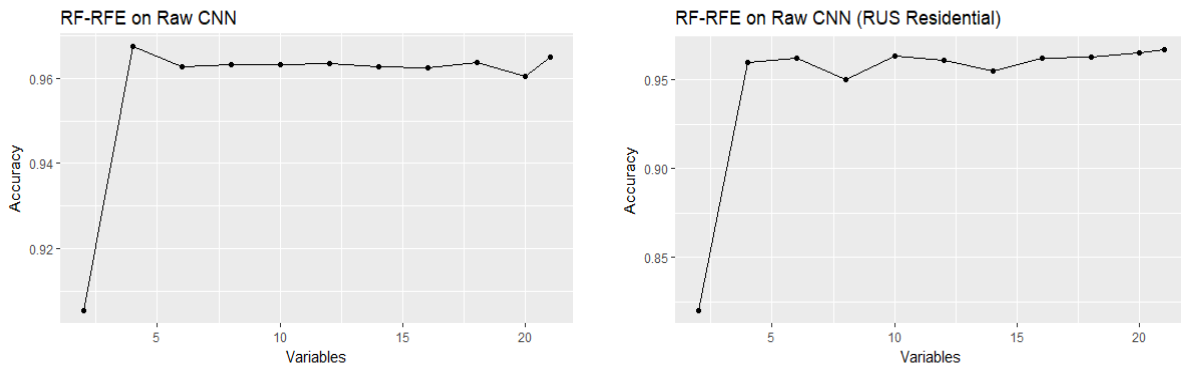


Figure 6. Comparing RF - RFE results for raw CNN and raw CNN with RUS.

Since the edited CNN building detection dataset is just the cleaner version of the raw CNN dataset (removed false positives), the datasets had the same 4 features for both algorithms: total area of all building detections in a parcel (`p_bd_area`), nearest building detection distance (`nearestbddist`), total perimeter of all building detections in a parcel (`p_bd_peri`), and number of building detections in a parcel (`num_bd_in_parcel`). As stated in Section 2.5.2, the values of

these features use the inside centroid of the building detections and the unit of measure is in meters. The LiDAR dataset RFE output was different than the CNN datasets. Figure 7 shows that the LiDAR RFE output was 21 variables for the RF – RFE with RUS the residential class, but the more simplistic model consist of 8 features. These 8 features have the same 4 features as the edited and raw CNN RFE output plus the following: number of building footprints within 100 m (num_bd_within100), the area of the individual building footprint (s_bd_area), the perimeter of the individual building footprint (s_bd_peri), and the median area of the building footprints within a 100 m radius (medianarea100).

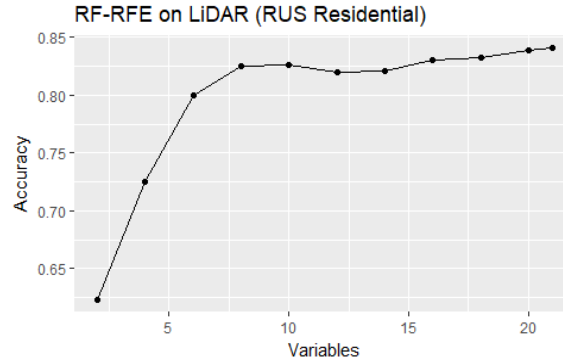


Figure 7. RF – RFE output for the LiDAR dataset with RUS.

When considering an ensemble of trees or a mapping in a higher dimensional feature space, interpreting why the selected features are important for this nine - multiclass problem is difficult. However, some of the features are intuitively understandable. For example, the individual area and perimeter of the building footprint or detection has importance in the LiDAR data and not CNN because the LiDAR derived building footprints have different building coverage, resulting in vastly difference values for the individual area and perimeter features. Even with RUS, the residential class takes up most of the data. Thus, comparing residential with nonresidential, residential typically has less building detections or footprints inside a parcel and a more distinct distance pattern when considering its nearest neighbor. Table 7 summarizes the features and hyperparameter(s) used for the different datasets and algorithms. Note, the Boruta R package (Kursa & Rudnicki, 2017), which has shown to be a superior feature selection method for RF (Degenhardt, Seifert, & Szymczak, n.d.), was conducted too. When training a RUS-RF with 20% of the raw CNN dataset, Boruta gave the same result as the optimal RFE output and claimed all 21 features were significant.

3.2 Classifying LiDAR Building Footprints, Edited CNN, and Raw CNN in Knox County

The initial reason for comparing the three datasets was to verify that the raw CNN dataset produced similar classification results for LiDAR and the cleaner version of the CNN building detections. It was assumed that LiDAR would have the best classification accuracies, but results showed the exact opposite. Figure 8 compares the F – scores of the three datasets for all nine building usages. F – score was the metric selected for visual comparison because unlike precision and recall it considers both false positives (FP) and false negatives (FN). However, Table A - 4 in the Appendix shows the precision, recall, and specificity values for the three datasets.

Table 7. Selected features and hyperparameter(s) for the different datasets and algorithms.

Algorithms and Datasets	Hyperparameter(s)	Features (Ranked)
RF and RUS - RF for LiDAR	$m_{try} = 2$	s_bd_area, s_bd_peri, num_bd_in_parcel, medianarea100, num_bd_within100, p_bd_area, nearestbddist, p_bd_peri,
SVM and RUS - SVM for LiDAR	$C = 16$ and $\gamma = 8$	num_bd_within100, p_bd_area, medianarea100, nearestbddist, p_bd_peri, s_bd_area, s_bd_peri, num_bd_in_parcel,
RF and RUS - RF for edited CNN and raw CNN	$m_{try} = 2$	p_bd_area, nearestbddist, p_bd_peri, num_bd_in_parcel
SVM and RUS - SVM for edited CNN and raw CNN	$C = 128$ and $\gamma = 4$	p_bd_area, p_bd_peri, num_bd_in_parcel, nearestbddist

The residential category outcome was better than expected. All three datasets have almost the exact same residential F - scores. When considering the nonresidential categories for the edited and raw CNN building detections, the difference between F - scores is small. Even though the false positives in the raw CNN dataset do not land on a building, they do land on a parcel, meaning they pertain to a building usage category. Therefore, the slightly higher F - score between these two datasets implies that the false positives found in the raw CNN building detections are helping in the classification process. Comparing LiDAR and raw CNN for the nonresidential categories is puzzling because it was unexpected to find that the raw automatic building extractions outperform LiDAR. The F - scores in the LiDAR datasets are significantly lower for industry, public, recreational, and transport building usages. Randomly selecting a bad representation of the LiDAR building footprints is not the cause of this occurrence because the random 20% training data selected for the three datasets were considering the same parcels. LiDAR seems to misclassify the nonresidential categories significantly more than the raw CNN dataset, and this is shown in the confusion matrices. Table 8 shows that a RUS - RF on LiDAR classifies most of the industry, public, recreational, and transport building usages as commercial or residential. Although the confusion matrix shows that the raw CNN has misclassification too, most of the classifications are for the correct building usage category.

Figure 9 compares LiDAR, edited CNN, and raw CNN for commercial, recreational, and industry building usages. The raw CNN building detections for commercial has distinctly different building area coverage than LiDAR as well as multiple building extractions for one building. The commercial building usage category in the raw CNN output appears to have its own signature that provides a clear distinction between other building usage categories. Figure 9 also shows how the false positives and building coverage aid in the classification of raw CNN. The recreational and industry building usage examples provided in Figure 9 are both classified as residential in the LiDAR algorithm but are correctly classified in the raw CNN algorithm. Thus, a strong case can be made that the faults found in the raw CNN, which are multiple building extractions for one building footprint, large numbers of false positives, and having a smaller building area coverage, help classify the nonresidential building usages. This is a significant outcome in this classification study, and results will continue with only the raw CNN building extractions. However, with the faults in the raw CNN datasets, classification results should cautiously be interpreted.

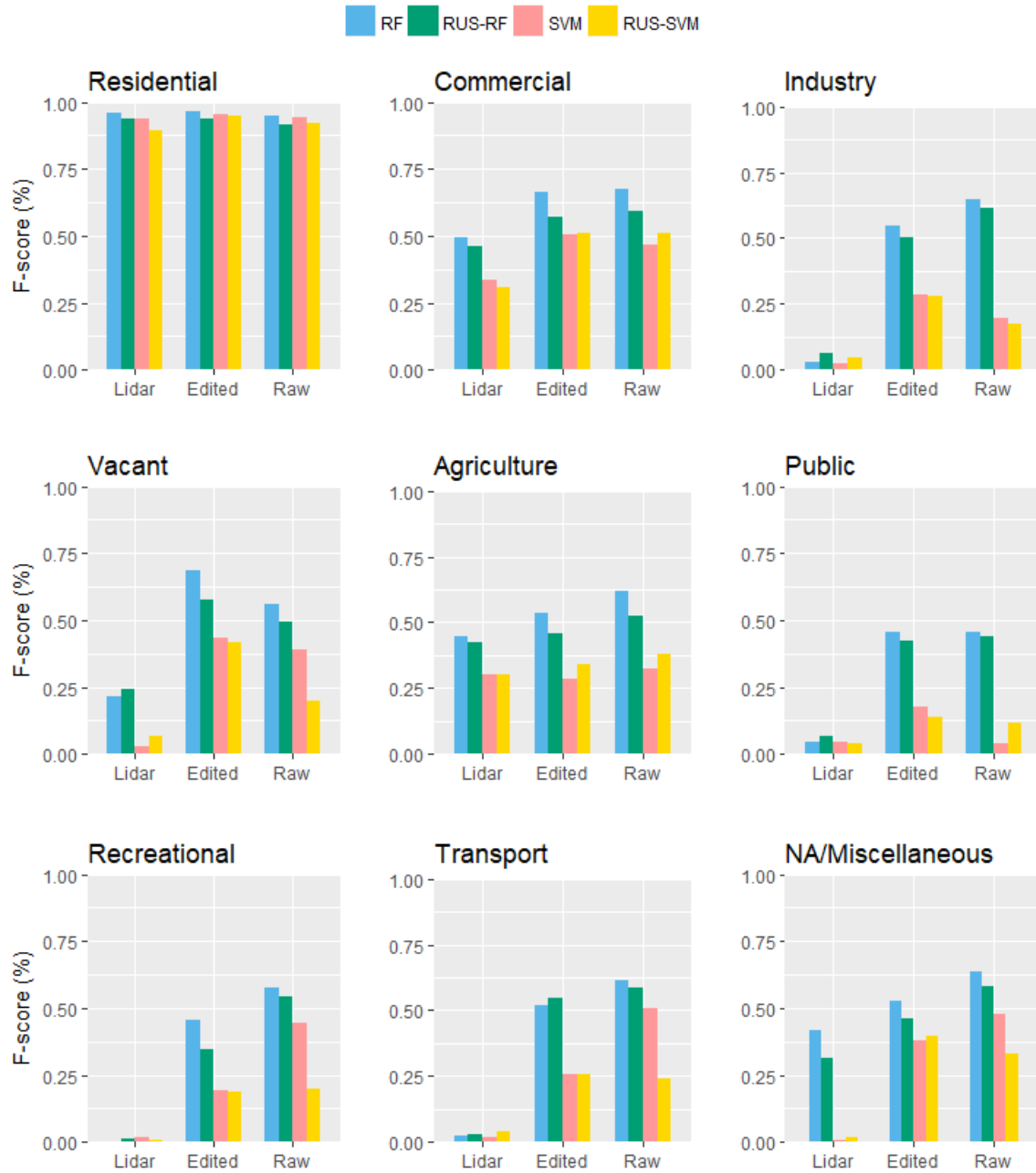


Figure 8. F - score (%) on the nine building usages categories for the three datasets (LiDAR, edited CNN, and raw CNN) and four algorithms (RF, RUS - RF, SVM, RUS - SVM).

Table 8. LiDAR derived building footprints (top) and raw CNN building extractions (bottom) confusion matrices calculated from RUS - RF. Percentages were calculated from each cell count divided by its total number of actual values (corresponding row sum).

	agr	com	ind	NA/misc	pub	rec	res	tra	vac
agr	0.66	0.10	0.00	0.00	0.00	0.00	0.22	0.00	0.01
com	0.04	0.61	0.00	0.01	0.01	0.00	0.30	0.00	0.03
ind	0.05	0.57	0.03	0.01	0.01	0.00	0.25	0.00	0.07
NA/misc	0.09	0.19	0.00	0.24	0.00	0.00	0.37	0.00	0.11
pub	0.06	0.47	0.00	0.00	0.04	0.00	0.40	0.00	0.02
rec	0.15	0.28	0.00	0.00	0.01	0.01	0.44	0.00	0.12
res	0.03	0.04	0.00	0.00	0.00	0.00	0.92	0.00	0.01
tra	0.13	0.53	0.02	0.01	0.02	0.00	0.25	0.02	0.02
vac	0.03	0.15	0.00	0.04	0.00	0.00	0.55	0.00	0.22

	agr	com	ind	NA/misc	pub	rec	res	tra	vac
agr	0.67	0.07	0.00	0.01	0.01	0.00	0.20	0.00	0.04
com	0.03	0.71	0.00	0.00	0.01	0.00	0.22	0.00	0.02
ind	0.05	0.32	0.51	0.00	0.01	0.00	0.10	0.00	0.01
NA/misc	0.09	0.16	0.00	0.50	0.01	0.00	0.17	0.00	0.07
pub	0.05	0.32	0.00	0.00	0.34	0.00	0.27	0.00	0.02
rec	0.07	0.26	0.01	0.00	0.01	0.43	0.18	0.00	0.04
res	0.03	0.05	0.00	0.00	0.00	0.00	0.89	0.00	0.02
tra	0.05	0.19	0.02	0.00	0.01	0.00	0.19	0.46	0.08
vac	0.06	0.08	0.00	0.01	0.00	0.00	0.21	0.00	0.62



Figure 9. Commercial (top), recreational (middle) and industry (bottom) building usages.

When analyzing which is the better model for classifying the raw CNN dataset, it is helpful to compare all four metrics. Figure 10 shows that RF and RUS - RF outperforms SVM and RUS - SVM. The SVM algorithms could fine - tune its hyperparameters further to obtain higher results but conducting SVM is already more time consuming than RF and fine - tuning C and γ would just tack on more time. Thus, RF and RUS - RF are now the models of focus. The residential class is the only building usage that has precision, recall, and F - score above the classification goal of 75% - 80%, but at the same time this class has the lowest specificity. RUS - RF is the better model for classifying residential because its specificity is 79% while RF specificity is only 62%. Yet, RF has a higher F - score for all nonresidential building usages. Higher precision and recall metrics vary between these two algorithms.

3.3 Altering Training Data in Knox County and Testing in Hamilton County

Continuing with Knox County and only conducting the RF and RUS - RF algorithms on the raw CNN building extractions, the next objective is to alter the amount of training data. Figure 11 explores how the F - scores change in Knox County when the raw CNN training data for RF and RUS - RF is altered to 5%, 20%, and 50%. The F - score in the residential building usage category has very little change when training both algorithms with drastically different training data amounts. The specificity of residential using a RUS - RF on 5% training data is also comparable with a value of 77%. The other eight building usages F - scores increased extensively when training the models on 50% rather than 5% of their class (see Table A - 5 in Appendix for other metric scores).

From these results, it is anticipated that there will only be useful residential building classification metric values when extrapolating the RF and RF - RUS methods for classification of Hamilton County. To provide the most data for the nonresidential classes, all of the raw CNN building detections in Knox County are used to classify the raw CNN building detections in Hamilton County. Table A - 6 in the Appendix lists the different building usage metrics for testing in Hamilton County, and it is verified that nonresidential building usages simply cannot train a RF or RUS - RF in Knox County to effectively learn the nonresidential building usages in Hamilton County. Table 9 shows the confusion matrix of Hamilton. It appears that the nonresidential categories are being classified as commercial or residential, and that residential is being classified more as vacant. Focusing on the residential building usage results, Figure 12 shows that the precision, recall, and F - score values are acceptable while the specificity value is unfortunately subpar.

Increasing the RUS amount to discard 90% of the residential would increase the specificity from 62% to 70%, but at the same time the F - score decreased from 81% to 70%.

Knox and Hamilton County have different Land - Use codes that generate the broad Land - Use categories (see Table A - 2 in Appendix), but the specificity in residential did not improve when constructing the nine class to contain the exact same Land - Use codes. By investigating the distribution differences of each predictor variables for both counties, the nearest building distance feature was the only one that may be problematic. Figure 13 shows the log distribution of each building detection's distance to its nearest building detection neighbor for all of Knox and Hamilton County residential class. It was found that removing this variable from the classification algorithm decreased all metrics.

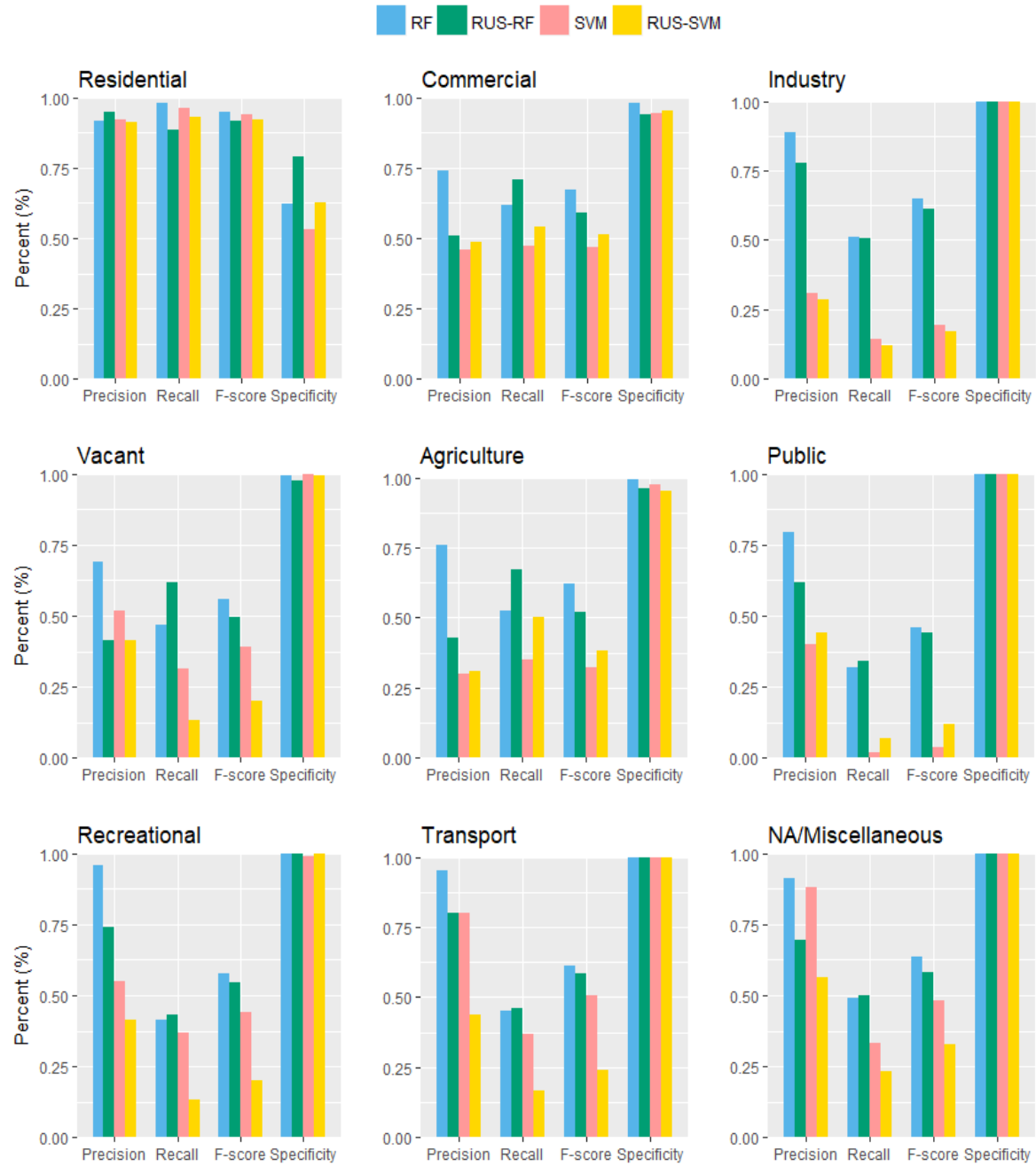


Figure 10. Comparing the metrics of the four algorithms for the raw CNN building detections.

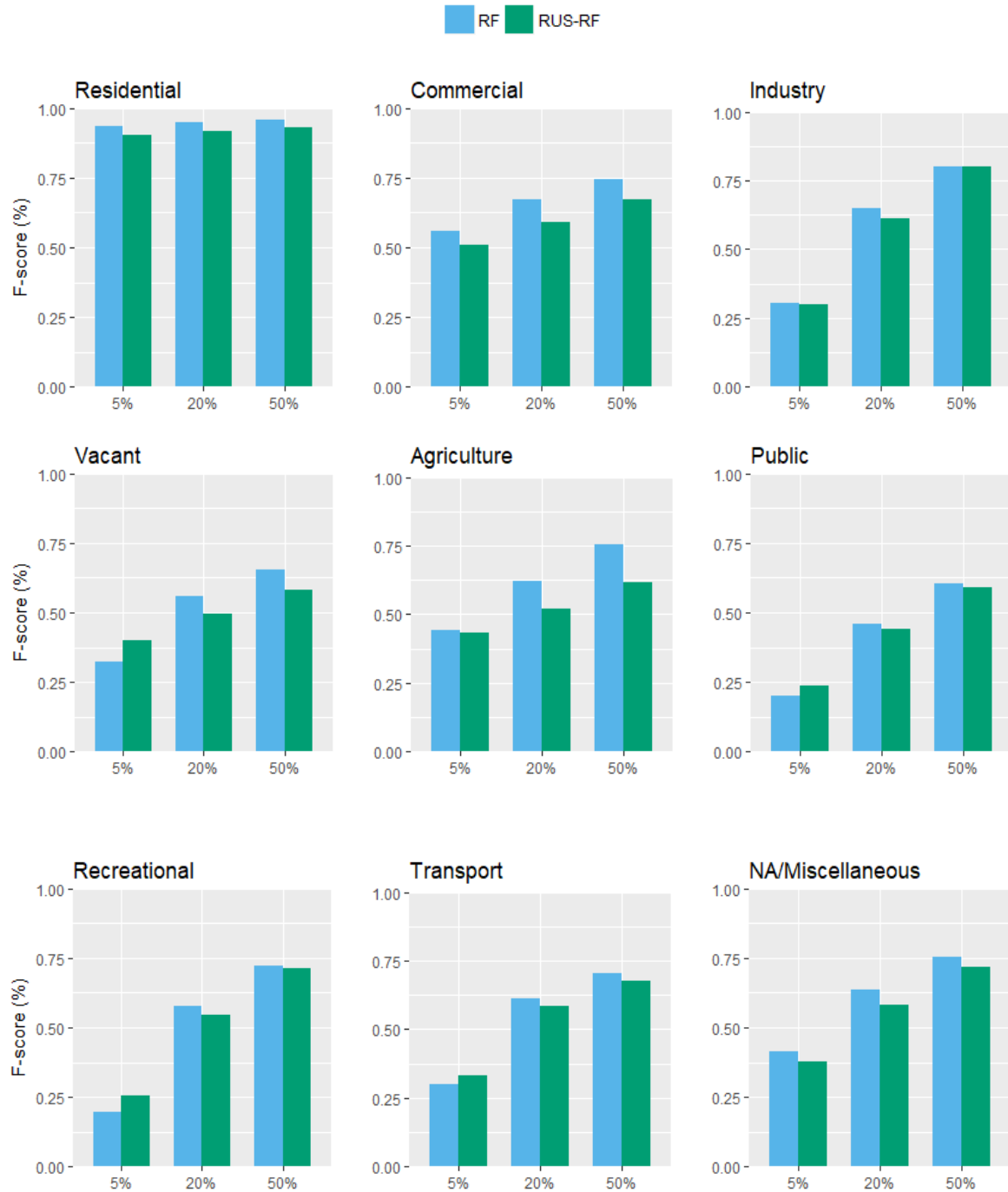


Figure 11. Comparing F - scores of RF and RUS - RF when implementing different training data.

Table 9. Confusion matrix for training RUS - RF with all of Knox and testing on Hamilton County. Percentages were calculated from each cell count divided by its total number of actual values (corresponding row sum).

	agr	com	ind	NA/misc	pub	rec	res	tra	vac
agr	0	0.67	0.27	0	0	0	0.05	0	0.02
com	0.06	0.45	0.01	0.02	0.02	0.01	0.39	0	0.05
ind	0.05	0.6	0.03	0.02	0.07	0	0.18	0.01	0.05
NA/misc	0.02	0.37	0	0.01	0.02	0	0.49	0	0.09
pub	0.02	0.58	0.01	0.08	0.02	0	0.24	0	0.04
rec	0.11	0.27	0.05	0.08	0.06	0.01	0.3	0.02	0.11
res	0.02	0.05	0	0.01	0	0	0.74	0	0.17
tra	0.08	0.44	0.01	0.02	0.03	0	0.34	0	0.08
vac	0.14	0.08	0	0.01	0	0	0.57	0	0.19

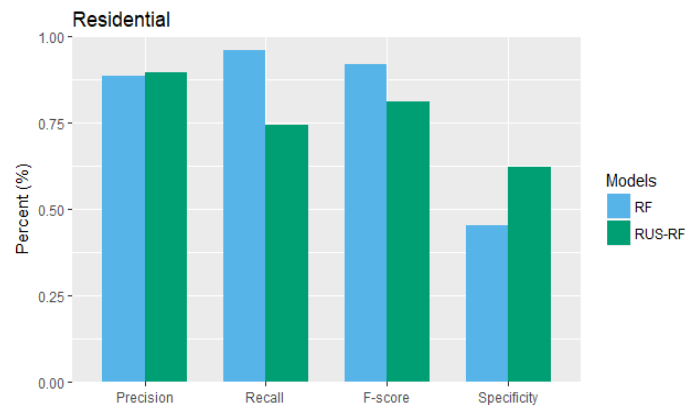


Figure 12. Comparing the metrics of training RF and RF - RUS on all of Knox County to test on Hamilton County.

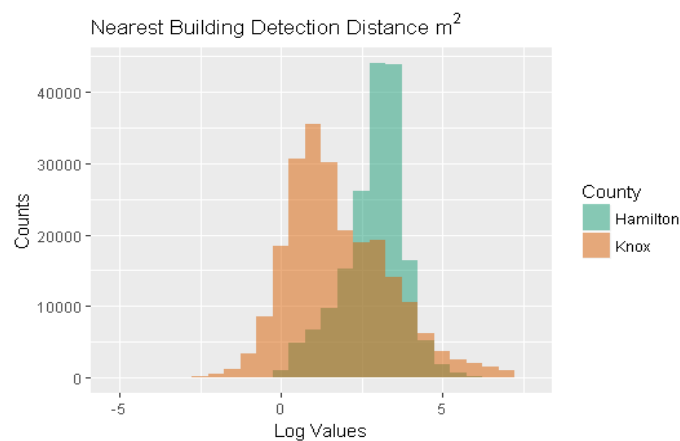


Figure 13. Comparing the nearest building distance m^2 predictor variable for Knox and Hamilton County.

4. DISCUSSION

This paper explored classification of the following broad building usage categories: residential, commercial, industrial, vacant, agriculture, public, recreational, transport, NA/misc. Using only the geographic and geometric spatial distribution of building extractions, four algorithms were implemented to classify the nine building usages for LiDAR, CNN building extractions with removal of false positives, and raw CNN building extractions in Knox County, TN. When comparing the RF and SVM algorithms on the original data and these algorithms with 80% of the residential class removed (RUS - RF and RUS - SVM), the RUS algorithms insignificantly altered the precision, recall and F - score for residential but dramatically improved its specificity.

For the three datasets, RF and RUS - RF outperformed SVM and RUS - SVM, and the raw CNN building extractions with four features (i.e. total area of all building detections in a parcel m^2 , nearest building detection distance m , total perimeter of all building detections in a parcel m , and number of building detections in a parcel) produced very comparable residential classification results to the LiDAR and edited CNN. Furthermore, the raw CNN output having multiple detections for one building footprint, numerous false positives, and different building area coverage, helped produce higher classification results than LiDAR and edited CNN for the other eight building usages. With LiDAR, the best remote sensing data available, having the lowest classification accuracies, there was justification to continue with classifying the raw CNN dataset. Examining the change in classification outcomes when obtaining different amounts of raw CNN training data was the next objective. Results showed that when a RF or RUS - RF algorithm is implemented on the raw CNN building detections, with the same four features, only 5% of Knox County is needed to successfully classify residential for the remaining Knox County. The other eight building usage categories need a training dataset of 50% or more. Finally, this study showed that it is difficult to extrapolate the RF classification algorithm, which was trained on all of Knox County, to the geographic area of Hamilton, TN. It was expected to see the very low nonresidential classification accuracies, but unfortunately even with RUS the specificity in the residential class is also very low.

Overall, classifying building extractions from a raw CNN building map did have some success, but considering false positives (detecting a building when the building does not exist) and not even considering the false negatives (not detecting a building when the building does exist) the results should be interpreted very carefully. In the future, a smaller multiclass problem might be investigated. Nine classes were selected because they fit the data that were available, and unfortunately the confusion matrixes are sporadic in term of building usages being misclassified as other building usages. Thus, finding an optimal subset of building usage classes that fits thematically and mathematically will be an extensive data mining objective. Simply classifying residential vs nonresidential was considered but underperformed the nine - multiclass problem (see Table A - 7 in Appendix).

An additional data mining objective could be investigating if different Land - Use areas produce similar results. For example, with the provided datasets, it is still unclear how an algorithm would classify agriculture if the focus area consisted of 80% agriculture rather than 80% residential. Training and testing with different Land - Use areas would help in understanding if the misclassification of the nonresidential categories is from a lack of data or from the possibility

that the features and algorithms selected simply cannot produce clear nonresidential groups. As with any type of modeling, there is also a strong interest to incorporate a level of uncertainty. Uncertainty could be studied in the future by either calculating a standard deviation from training and testing the algorithms multiple times. Perhaps a better approach is investigating how to incorporate uncertainty in the algorithms from the predictor variables. To conclude, further research will be exploring interactions between predictor variables (e.g. can the area and perimeter interaction variable have more importance to a model than having both variables), diving into a deep learning approach (CNN algorithm using classification), and eventually expanding outside the United States and explore if CoreLogic parcel information can be substituted with OpenStreetMap (OSM) and Points of Interest (POI).

LIST OF REFERENCES

- Ambroise, C., & McLachlan, G. J. (2002). Selection bias in gene extraction on the basis of microarray gene-expression data. *Proceedings of the National Academy of Sciences*, 99(10), 6562–6566. <https://doi.org/10.1073/pnas.102102699>
- ArcMap | ArcGIS Desktop. (n.d.). Retrieved March 11, 2018, from <http://desktop.arcgis.com/en/arcmap/>
- Bégué, A., Arvor, D., Bellon, B., Betbeder, J., de Aballeyra, D., P. D. Ferraz, R., ... R. Verón, S. (2018). Remote Sensing and Cropping Practices: A Review. *Remote Sensing*, 10(1), 99. <https://doi.org/10.3390/rs10010099>
- Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), 5–32. <https://doi.org/10.1023/A:1010933404324>
- Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and Regression Trees*. Taylor & Francis.
- Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *J. Artif. Int. Res.*, 16(1), 321–357.
- CoreLogic | Our Company. (n.d.). Retrieved February 3, 2018, from <http://www.corelogic.com/about-us/our-company.aspx>
- Degenhardt, F., Seifert, S., & Szymczak, S. (n.d.). Evaluation of variable selection methods for random forests and omics data sets. *Briefings in Bioinformatics*. <https://doi.org/10.1093/bib/bbx124>
- Granitto, P. M., Furlanello, C., Biasioli, F., & Gasperi, F. (2006). Recursive feature elimination with random forest for PTR-MS analysis of agroindustrial products. *Chemometrics and Intelligent Laboratory Systems*, 83(2), 83–90. <https://doi.org/10.1016/j.chemolab.2006.01.007>

- Gregorutti, B., Michel, B., & Saint-Pierre, P. (2017). Correlation and variable importance in random forests. *Statistics and Computing*, 27(3), 659–678.
<https://doi.org/10.1007/s11222-016-9646-1>
- Guyon, I., & Elisseeff, A. (2003). An Introduction to Variable and Feature Selection. *J. Mach. Learn. Res.*, 3, 1157–1182.
- Han, T., Jiang, D., Zhao, Q., Wang, L., & Yin, K. (2017). Comparison of random forest, artificial neural networks and support vector machine for intelligent diagnosis of rotating machinery. *Transactions of the Institute of Measurement and Control*, 0142331217708242. <https://doi.org/10.1177/0142331217708242>
- Hasan, M. A. M., Nasser, M., Pal, B., & Ahmad, S. (2014). Support Vector Machine and Random Forest Modeling for Intrusion Detection System (IDS). *Journal of Intelligent Learning Systems and Applications*, 2014. <https://doi.org/10.4236/jilsa.2014.61005>
- He, H., & Garcia, E. A. (2009). Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9), 1263–1284.
<https://doi.org/10.1109/TKDE.2008.239>
- He, Haibo, & Ma, Y. (2013). *Imbalanced learning: foundations, algorithms, and applications*. Retrieved from <http://site.ebrary.com/id/10716627>
- Hecht, R., Meinel, G., & Buchroithner, M. (2015). Automatic identification of building types based on topographic databases – a comparison of different data sources. *International Journal of Cartography*, 1(1), 18–31. <https://doi.org/10.1080/23729333.2015.1055644>
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning: with Applications in R*. Springer Science & Business Media.

- Jensen, J. R., & Cowen, D. C. (2011). Remote Sensing of Urban/Suburban Infrastructure and Socio-Economic Attributes. In Martin Dodge, R. Kitchin, & C. Perkins (Eds.), *The Map Reader* (pp. 153–163). John Wiley & Sons, Ltd.
<https://doi.org/10.1002/9780470979587.ch22>
- Jochem, W. C., Bird, T. J., & Tatem, A. J. (2018). Identifying residential neighbourhood types from settlement points in a machine learning approach. *Computers, Environment and Urban Systems*. <https://doi.org/10.1016/j.compenvurbsys.2018.01.004>
- Kohavi, R. (1995). A Study of Cross-validation and Bootstrap for Accuracy Estimation and Model Selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2* (pp. 1137–1143). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. Retrieved from <http://dl.acm.org/citation.cfm?id=1643031.1643047>
- Kuhn, M. (2017). caret: Classification and Regression Training (Version 6.0-78). Retrieved from <https://cran.r-project.org/web/packages/caret/index.html>
- Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling*. Springer Science & Business Media.
- Kumar, M., & Thenmozhi, M. (2006). *Forecasting Stock Index Movement: A Comparison of Support Vector Machines and Random Forest* (SSRN Scholarly Paper No. ID 876544). Rochester, NY: Social Science Research Network. Retrieved from <https://papers.ssrn.com/abstract=876544>
- Kursa, M. B., & Rudnicki, W. R. (2017). Boruta: Wrapper Algorithm for All Relevant Feature Selection (Version 5.2.0). Retrieved from <https://cran.r-project.org/web/packages/Boruta/index.html>

- Li, E., Femiani, J., Xu, S., Zhang, X., & Wonka, P. (2015). Robust Rooftop Extraction From Visible Band Images Using Higher Order CRF. *IEEE Transactions on Geoscience and Remote Sensing*, 53(8), 4483–4495. <https://doi.org/10.1109/TGRS.2015.2400462>
- NAIP Imagery. (n.d.). [page]. Retrieved March 11, 2018, from <https://www.fsa.usda.gov/programs-and-services/aerial-photography/imagery-programs/naip-imagery/>
- Ngo, T. T., Mazet, V., Collet, C., & Fraipont, P. de. (2017). Shape-Based Building Detection in Visible Band Images Using Shadow Information. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(3), 920–932. <https://doi.org/10.1109/JSTARS.2016.2598856>
- Ogut, J. O., Piepho, H.-P., & Schulz-Streeck, T. (2011). A comparison of random forests, boosting and support vector machines for genomic selection. *BMC Proceedings*, 5(S3), S11. <https://doi.org/10.1186/1753-6561-5-S3-S11>
- Olson, D. L., & Delen, D. (2008). *Advanced Data Mining Techniques*. Springer Science & Business Media.
- Qiu, S., & Wang, J. (2017). The prediction of food additives in the fruit juice based on electronic nose with chemometrics. *Food Chemistry*, 230, 208–214. <https://doi.org/10.1016/j.foodchem.2017.03.011>
- R Core Team (2013). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <http://www.R-project.org/>. (n.d.).
- Rottensteiner, F., & Bries, C. (2002). A new method for building extraction in urban areas from high-resolution LiDAR data. Retrieved from

<http://citeseerx.ist.psu.edu/viewdoc/citations;jsessionid=C2DFF7DBA410D3B3022E36F4B0BF3467?doi=10.1.1.111.4853>

- Statnikov, A., & Aliferis, C. F. (2007). Are Random Forests Better than Support Vector Machines for Microarray-Based Cancer Classification? *AMIA Annual Symposium Proceedings, 2007*, 686–690.
- Steiniger, S., Lange, T., Burghardt, D., & Weibel, R. (2008). An Approach for the Classification of Urban Building Structures Based on Discriminant Analysis Techniques. *T. GIS*, 12, 31–59. <https://doi.org/10.1111/j.1467-9671.2008.01085.x>
- TNGIS - LiDAR. (n.d.). Retrieved February 3, 2018, from <http://www.tngis.org/lidar.htm>
- Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1), 67–82. <https://doi.org/10.1109/4235.585893>
- Xian, G. Z. (2015). *Remote Sensing Applications for the Urban Environment*. CRC Press.
- Xie, Z.-X., Hu, Q.-H., & Yu, D.-R. (2006). Improved Feature Selection Algorithm Based on SVM and Correlation. In *Advances in Neural Networks - ISNN 2006* (pp. 1373–1380). Springer, Berlin, Heidelberg. https://doi.org/10.1007/11759966_204
- Xingquan, Z. (2007). *Knowledge Discovery and Data Mining: Challenges and Realities: Challenges and Realities*. Idea Group Inc (IGI).
- Yuan, J. (2016). Automatic Building Extraction in Aerial Scenes Using Convolutional Networks. *ArXiv:1602.06564 [Cs]*. Retrieved from <http://arxiv.org/abs/1602.06564>
- Zribi, M. (2016). *Land Surface Remote Sensing: Environment and Risks*. Elsevier.

APPENDIX

APPENDIX A

Table A - 1. Knox County specific Land - Use codes for LiDAR building footprints, edited building detections (only the building detections that intersect a building footprint within a three - meter buffer), and raw building detections (all CNN building detections).

Land-Use Code	Description	Lidar Freq	Edited Freq	Unedited Freq	Land-Use Code	Description	Lidar Freq	Edited Freq	Unedited Freq
0	NA	1514	1076	1887	286	WHOLESALE	212	385	600
100	RESIDENTIAL(NEC)	5481	5000	9080	300	INDUSTRIAL(NEC)	2	5	7
102	TOWNHOUSE/ROWHOUSE	89	134	138	308	CHEMICAL	4	10	17
106	APARTMENT	87	151	176	309	TEXTILE/CLOTHES/CARPETINDUST	3	5	5
112	CONDOMINIUM	961	1268	1337	310	PAPER&ALLIEDINDUSTRY	33	98	152
114	COMMONAREA	37	19	24	316	FOODPROCESSING	15	49	73
116	MIDRISECONDO	41	39	52	323	INDUSTRIALPLANT	6	5	13
118	FRAT/SORORITYHOUSE	1	1	1	324	LIGHTINDUSTRIAL	44	93	161
131	MULTIFAMILY10UNITSPLUS	2054	2493	3272	326	LUMBERYARD	34	37	67
132	MULTIFAMILY10UNITSLESS	401	382	427	331	METALPRODUCT	56	149	211
133	MULTIFAMILYDWELLING	2722	2574	3017	336	MINIWAREHOUSE	146	172	223
136	MOBILEHOMEPARK	726	669	2292	344	PETROLEUM	117	124	187
137	MOBILEHOME	14	14	41	349	MINE/QUARRY	33	65	218
138	MANUFACTUREDHOME	162	153	251	353	TECHNOLOGICALINDUSTRY	23	77	113
142	MOTEL	18	27	47	358	STORAGE	290	336	627
148	PUD	1398	1617	1673	400	VACANTLAND(NEC)	851	702	4083
155	GROUPQUARTERS	116	186	252	421	COMMONLAND	1385	1690	2332
163	SFR	137027	133965	162664	453	OPENSOURCE	3	4	8
164	TRANSIENTLODGING	125	240	399	500	AGRICULTURAL(NEC)	3623	3509	8059
200	COMMERCIAL(NEC)	893	1525	2486	511	FARMS	60	71	303
204	AUTOREPAIR	287	333	538	540	FOREST	195	149	614
205	AUTOSALES	104	129	311	605	POLICE/FIRE/CIVILDEFENSE	9	8	13
209	BUSINESSPARK	2115	2591	3982	606	USPOSTALSERVICE	4	4	8
210	CARWASH	12	14	23	615	FEDERALBUILDING	5	7	13
211	COMMERCIALBUILDING	148	181	258	640	POSSESSORYINTEREST	8	6	12
212	CEMETERY	18	13	173	652	NURSERY/SCHOOL	11	15	23
214	CONVALESCENTHOSPITAL	31	38	57	660	EDUCATIONALSERVICE	208	329	535
217	DEPARTMENTSTORE	3	14	16	665	PUBLICSCHOOL	9	13	21
220	STOREFRANCHISE	2	2	2	670	PUBLICSERVICE	7	11	12
221	FASTFOODFRANCHISE	1	1	1	675	RELIGIOUS	1132	1561	2359
223	FINANCIALBUILDING	249	323	471	680	UNIVERSITY	23	47	75
225	FUNERALHOME	5	8	11	700	RECREATIONAL(NEC)	90	118	258
230	HOSPITAL	2	16	16	709	AMUSEMENTPARK	14	22	50
237	MEDICALBUILDING	172	238	328	721	BOWLINGALLEY	3	7	12
240	LAUNDROMAT	6	4	5	725	CLUB	80	93	230
244	OFFICEBUILDING	726	984	1560	728	HEALTHCLUB	5	11	17
249	MISCIMPROVEMENTS	4	6	6	750	GYMNASIUM	1	1	1
251	MISCBUILDING	420	731	1111	757	PARK	11	8	15
252	PARKINGLOT	30	57	210	770	TENNISCLUB	22	32	82
255	PARKINGSTRUCTURE	7	19	26	780	SKATINGRINK	3	5	6
261	RESTAURANTBUILDING	33	46	86	798	LIBRARY/MUSEUM	3	10	10
266	SERVICESTATION	7	9	10	801	COMMUNICATIONFACILITY	44	57	141
269	MISCCOMMERICALSERVICES	361	434	687	806	AIRCRAFTFACILITY	16	21	31
270	SHOPPINGCENTER	15	25	43	839	MARINEFACILITY	15	24	51
276	APPAREL	39	71	86	860	RAILROADFACILITY	1	3	21
278	STOREBUILDING	10	11	17	880	UTILITIES	258	327	706
282	RETAILTRADE	2219	3271	4720	999	MISCELLANEOUS	541	685	1710
283	SUPERMARKET	15	20	49	601	TAXEXEMPT	0	1	3
284	FOODSTORES	90	135	207					

Table A - 2. Frequencies of Knox and Hamilton raw CNN building detection for specific Land - Use categories.

Land Use Code	Description	Knox Freq	Hamilton Freq	Land Use Code	Description	Knox Freq	Hamilton Freq
0	NONE	1887	882	806	AIRCRAFTFACILITY	31	15
100	RESIDENTIAL(NEC)	9080	2641	860	RAILROADFACILITY	21	104
112	CONDOMINIUM	1337	2091	880	UTILITIES	706	690
131	MULTIFAMILY10UNITSPLUS	3272	2733	999	MISCELLANEOUS	1710	445
132	MULTIFAMILY10UNITSLESS	427	496	115	DUPLEX	NA	4167
133	MULTIFAMILYDWELLING	3017	150	135	MOBILE HOME LOT	NA	1972
136	MOBILEHOMEPARK	2292	1515	157	NURSING HOME	NA	3
137	MOBILEHOME	41	37	165	TRIPLEX	NA	186
142	MOTEL	47	4	208	AUTO WRECKING	NA	40
155	GROUPQUARTERS	252	60	222	FIN/INSURANCE/REAL ESTATE	NA	267
163	SFR	162664	126387	239	LABORATORY	NA	64
164	TRANSIENTLODGING	399	471	243	BAR	NA	92
200	COMMERCIAL(NEC)	2486	1214	262	RESTAURANT DRIVE IN	NA	560
204	AUTOREPAIR	538	55	320	HEAVY INDUSTRIAL	NA	34
205	AUTOSALES	311	155	328	LUMBER MILL	NA	90
210	CARWASH	23	94	364	WAREHOUSE	NA	1234
211	COMMERCIALBUILDING	258	42	435	INDUSTRIAL ACREAGE	NA	13
212	CEMETERY	173	137	440	INDUSTRIAL LOT	NA	8
217	DEPARTMENTSTORE	16	180	600	PUBLIC (NEC)	NA	7
223	FINANCIALBUILDING	471	182	650	SCHOOL	NA	431
230	HOSPITAL	16	36	654	HIGH SCHOOL	NA	160
237	MEDICALBUILDING	328	647	656	VOCATIONAL/TRADE SCHOOL	NA	18
240	LAUNDROMAT	5	1	671	CHARITABLE ORGANIZATION	NA	18
244	OFFICEBUILDING	1560	1003	690	NATURE FACILITY	NA	44
251	MISCBUILDING	1111	4	706	AMUSEMENT ARCADE	NA	35
252	PARKINGLOT	210	323	712	AUDITORIUM	NA	17
261	RESTAURANTBUILDING	86	1	742	GOLF COURSE	NA	407
266	SERVICESTATION	10	3117	754	LAKE/RIVER/BEACH	NA	5
269	MISCCOMMERCIALSERVICES	687	3846	755	MARINA FACILITY	NA	255
270	SHOPPINGCENTER	43	430	796	THEATER	NA	14
278	STOREBUILDING	17	2	808	AIRPORT	NA	164
282	RETAILTRADE	4720	3382	818	ELECTRICAL FACILITY	NA	183
283	SUPERMARKET	49	10	877	TRANSPORT FACILITY	NA	407
284	FOODSTORES	207	255	102	TOWNHOUSE/ROWHOUSE	138	NA
286	WHOLESALE	600	868	106	APARTMENT	176	NA
300	INDUSTRIAL(NEC)	7	271	114	COMMONAREA	24	NA
308	CHEMICAL	17	646	116	MIDRISECONDO	52	NA
309	TEXTILE/CLOTHES/CARPETINDUST	5	203	118	FRAT/SORORITYHOUSE	1	NA
310	PAPER&ALLIEDINDUSTRY	152	167	138	MANUFACTUREDHOME	251	NA
316	FOODPROCESSING	73	350	148	PUD	1673	NA
324	LIGHTINDUSTRIAL	161	775	209	BUSINESSPARK	3982	NA
336	MINIWAREHOUSE	223	526	214	CONVALESCENTHOSPITAL	57	NA
344	PETROLEUM	187	33	220	STOREFRANCHISE	2	NA
358	STORAGE	627	2	221	FASTFOODFRANCHISE	1	NA
400	VACANTLAND(NEC)	4083	5260	225	FUNERALHOME	11	NA
500	AGRICULTURAL(NEC)	8059	56	249	MISCIMPROVEMENTS	6	NA
540	FOREST	614	4	255	PARKINGSTRUCTURE	26	NA
601	TAXEXEMPT	3	390	276	APPAREL	86	NA
605	POLICE/FIRE/CIVILDEFENSE	13	43	323	INDUSTRIALPLANT	13	NA
606	USPOSTALSERVICE	8	61	326	LUMBERYARD	67	NA
652	NURSERY SCHOOL	23	121	331	METALPRODUCT	211	NA
660	EDUCATIONALSERVICE	535	711	349	MINE/QUARRY	218	NA
675	RELIGIOUS	2359	47	353	TECHNOLOGICALINDUSTRY	113	NA
680	UNIVERSITY	75	127	421	COMMONLAND	2332	NA
700	RECREATIONAL(NEC)	258	23	453	OPENSACE	8	NA
721	BOWLINGALLEY	12	11	511	FARMS	303	NA
725	CLUB	230	14	615	FEDERALBUILDING	13	NA
728	HEALTHCLUB	17	62	640	POSSESSORYINTEREST	12	NA
750	GYMNASIUM	1	4	665	PUBLICSCHOOL	21	NA
757	PARK	15	286	670	PUBLICSERVICE	12	NA
770	TENNISCLUB	82	7	709	AMUSEMENTPARK	50	NA
780	SKATINGRINK	6	1	798	LIBRARY/MUSEUM	10	NA
801	COMMUNICATIONFACILITY	141	186	839	MARINEFACILITY	51	NA

Table A - 3. Knox and Hamilton County building usage accuracies referring to the percent of raw CNN building extraction that intersect building footprints within a three-meter buffer.

	Core Logic ID	Description	Detection Accuracy	Detections that Intersect a Footprint	Total Detections
Knox County	100 - 999	OVERALL	75.62%	173993	230077
	100 - 199	RESIDENTIAL	80.77%	149963	185670
	200 - 299	COMMERCIAL	64.56%	11656	18054
	300 - 399	INDUSTRY	59.62%	1240	2080
	400 - 499	VACANT	40.05%	2715	6779
	500 - 599	AGRICULTURE	41.84%	3802	9087
	600 - 699	PUBLIC	65.21%	2032	3116
	700 - 799	RECREATIONAL	46.77%	326	697
	800 - 899	TRANSPORT	45.80%	436	952
	" " - 999	NA-MISCELLANEOUS	50.05%	1823	3642
Hamilton County	100 - 999	OVERALL	75.77%	134197	177105
	100 - 199	RESIDENTIAL	81.63%	116930	143250
	200 - 299	COMMERCIAL	60.92%	10428	17117
	300 - 399	INDUSTRY	57.68%	2508	4348
	400 - 499	VACANT	20.08%	1172	5838
	500 - 599	AGRICULTURE	66.67%	42	63
	600 - 699	PUBLIC	62.77%	1381	2200
	700 - 799	RECREATIONAL	35.33%	412	1166
	800 - 899	TRANSPORT	34.39%	605	1759
	" " - 999	NA-MISCELLANEOUS	52.71%	719	1364

Table A - 4. Comparing LiDAR building footprints, edited building detections (only the building detections that intersect a building footprint within a three-meter buffer), and raw building detections (all CNN building detections) with models trained on around 20% of Knox County and testing on the remainder of Knox County.

	LiDAR Building Footprints				Edited Building Detections				Raw Building Detections			
	RF	SVM	RF-RUS	SVM-RUS	RF	SVM	RF-RUS	SVM-RUS	RF	SVM	RF-RUS	SVM-RUS
Overall												
Hyperparameters	mtry = 2	C = 16, sigma = 8	mtry = 2	C = 16, sigma = 8	mtry = 2	C = 128, sigma = 4	mtry = 2	C = 128, sigma = 4	mtry = 2	C = 128, sigma = 4	mtry = 2	C = 128, sigma = 4
Number of Features	8	8	8	8	4	4	4	4	4	4	4	4
(train, test) data	(42303, 135803)	(42303, 135803)	(14283, 135803)	(14283, 135803)	(42097, 137726)	(42097, 137726)	(14301, 137726)	(14301, 137726)	(45807, 183207)	(45807, 183207)	(16183, 183207)	(16183, 183207)
Residential (100 - 199)												
(train, test) data	(35025, 120743)	(35025, 120743)	(7005, 120743)	(7005, 120743)	(34744, 118441)	(34744, 118441)	(6948, 118441)	(6948, 118441)	(37029, 148114)	(37029, 148114)	(7405, 148114)	(7405, 148114)
Precision (%)	0.937	0.952	0.957	0.968	0.943	0.928	0.960	0.949	0.917	0.920	0.947	0.913
Recall (%)	0.977	0.919	0.922	0.830	0.985	0.980	0.918	0.943	0.982	0.962	0.887	0.931
F-Score (%)	0.957	0.935	0.939	0.894	0.964	0.953	0.939	0.946	0.948	0.941	0.916	0.922
Specificity (%)	0.478	0.625	0.670	0.779	0.633	0.535	0.765	0.687	0.623	0.532	0.791	0.627
Commercial (200 - 299)												
(train, test) data	(3408, 6110)	(3408, 6110)	(3408, 6110)	(3408, 6110)	(3459, 9464)	(3459, 9464)	(3459, 9464)	(3459, 9464)	(3620, 14476)	(3620, 14476)	(3620, 14476)	(3620, 14476)
Precision (%)	0.531	0.236	0.369	0.205	0.714	0.562	0.481	0.469	0.740	0.461	0.509	0.487
Recall (%)	0.466	0.569	0.608	0.639	0.616	0.457	0.706	0.566	0.616	0.473	0.709	0.542
F-Score (%)	0.496	0.334	0.459	0.310	0.662	0.504	0.572	0.513	0.672	0.467	0.592	0.513
Specificity (%)	0.981	0.913	0.951	0.883	0.982	0.974	0.944	0.953	0.981	0.942	0.941	0.951
Industry (300 - 399)												
(train, test) data	(389, 619)	(389, 619)	(389, 619)	(389, 619)	(388, 1037)	(388, 1037)	(388, 1037)	(388, 1037)	(415, 1659)	(415, 1659)	(415, 1659)	(415, 1659)
Precision (%)	0.263	0.102	0.304	0.071	0.823	0.348	0.642	0.348	0.887	0.310	0.779	0.288
Recall (%)	0.016	0.015	0.034	0.036	0.412	0.240	0.413	0.231	0.509	0.142	0.505	0.123
F-Score (%)	0.030	0.025	0.061	0.047	0.549	0.284	0.502	0.278	0.647	0.195	0.613	0.172
Specificity (%)	1.000	0.999	1.000	0.998	0.999	0.997	0.998	0.997	0.999	0.998	0.999	0.997
Vacant (400 - 499)												
(train, test) data	(578, 1967)	(578, 1967)	(578, 1967)	(578, 1967)	(641, 2081)	(641, 2081)	(641, 2081)	(641, 2081)	(1285, 5138)	(1285, 5138)	(1285, 5138)	(1285, 5138)
Precision (%)	0.331	0.077	0.260	0.088	0.843	0.587	0.540	0.511	0.690	0.520	0.413	0.412
Recall (%)	0.158	0.019	0.224	0.055	0.579	0.342	0.357	0.312	0.469	0.312	0.619	0.131
F-Score (%)	0.214	0.031	0.241	0.068	0.686	0.432	0.578	0.420	0.558	0.390	0.496	0.199
Specificity (%)	0.995	0.997	0.991	0.992	0.998	0.996	0.992	0.995	0.994	1.000	0.975	0.995
Agriculture (500 - 599)												
(train, test) data	(1490, 3076)	(1490, 3076)	(1490, 3076)	(1490, 3076)	(1456, 2947)	(1456, 2947)	(1456, 2947)	(1456, 2947)	(1796, 7180)	(1796, 7180)	(1796, 7180)	(1796, 7180)
Precision (%)	0.489	0.350	0.313	0.217	0.652	0.384	0.359	0.273	0.761	0.302	0.427	0.307
Recall (%)	0.414	0.267	0.663	0.497	0.458	0.230	0.627	0.463	0.525	0.350	0.672	0.503
F-Score (%)	0.448	0.303	0.425	0.302	0.538	0.288	0.456	0.344	0.621	0.324	0.522	0.381
Specificity (%)	0.990	0.988	0.959	0.959	0.995	0.992	0.975	0.973	0.993	0.973	0.963	0.954
Public (600 - 699)												
(train, test) data	(583, 1046)	(583, 1046)	(583, 1046)	(583, 1046)	(585, 1630)	(585, 1630)	(585, 1630)	(585, 1630)	(615, 2459)	(615, 2459)	(615, 2459)	(615, 2459)
Precision (%)	0.243	0.056	0.170	0.029	0.811	0.409	0.597	0.383	0.795	0.400	0.617	0.441
Recall (%)	0.026	0.041	0.041	0.078	0.319	0.112	0.330	0.085	0.320	0.020	0.342	0.068
F-Score (%)	0.047	0.047	0.066	0.043	0.458	0.175	0.425	0.139	0.457	0.038	0.440	0.117
Specificity (%)	0.999	0.995	0.998	0.980	0.999	0.998	0.997	0.998	0.999	0.998	0.997	0.999
Recreational (700 - 799)												
(train, test) data	(118, 170)	(118, 170)	(118, 170)	(118, 170)	(113, 249)	(113, 249)	(113, 249)	(113, 249)	(137, 544)	(137, 544)	(137, 544)	(137, 544)
Precision (%)	0.000	0.024	0.050	0.006	0.757	0.363	0.390	0.403	0.957	0.550	0.739	0.415
Recall (%)	0.000	0.012	0.006	0.006	0.325	0.133	0.313	0.124	0.414	0.370	0.432	0.131
F-Score (%)	0.000	0.016	0.011	0.006	0.455	0.194	0.347	0.190	0.578	0.442	0.545	0.199
Specificity (%)	1.000	0.999	1.000	0.999	1.000	1.000	0.999	1.000	1.000	0.989	1.000	0.999
Transport (800 - 899)												
(train, test) data	(158, 263)	(158, 263)	(158, 263)	(158, 263)	(153, 364)	(153, 364)	(153, 364)	(153, 364)	(190, 760)	(190, 760)	(190, 760)	(190, 760)
Precision (%)	0.231	0.071	0.333	0.053	0.905	0.429	0.754	0.462	0.950	0.801	0.799	0.436
Recall (%)	0.011	0.011	0.015	0.030	0.365	0.184	0.429	0.181	0.453	0.370	0.461	0.166
F-Score (%)	0.022	0.020	0.029	0.039	0.521	0.258	0.546	0.260	0.613	0.506	0.584	0.240
Specificity (%)	1.000	1.000	1.000	0.999	1.000	0.999	1.000	0.999	1.000	0.997	1.000	0.999
None - misc (" " and 999)												
(train, test) data	(554, 1809)	(554, 1809)	(554, 1809)	(554, 1809)	(558, 1513)	(558, 1513)	(558, 1513)	(558, 1513)	(720, 2877)	(720, 2877)	(720, 2877)	(720, 2877)
Precision (%)	0.615	0.014	0.459	0.019	0.726	0.644	0.568	0.578	0.911	0.881	0.695	0.564
Recall (%)	0.315	0.005	0.238	0.016	0.415	0.270	0.390	0.301	0.489	0.330	0.501	0.231
F-Score (%)	0.417	0.007	0.313	0.017	0.528	0.381	0.463	0.396	0.637	0.480	0.582	0.328
Specificity (%)	0.997	0.995	0.996	0.989	0.998	0.998	0.997	0.998	0.999	1.000	0.996	0.997

Table A - 5. Comparing metrics with of RF and RUS - RF with increasing Knox County's training data by 5%, 20%, and 50%.

	RF			RUS-RF		
	5%	20%	50%	5%	20%	50%
Overall						
Hyperparameters	mtry = 2	mtry = 2	mtry = 2	mtry = 2	mtry = 2	mtry = 2
Number of Features	4	4	4	4	4	4
(train, test) data	(11455, 217559)	(45807, 183207)	(114509, 114505)	(4048, 217559)	(16183, 183207)	(40451, 114505)
Residential (100 - 199)						
n	(9258, 175885)	(37029, 148114)	(92572, 92571)	(1851, 175885)	(7405, 148114)	(18514, 92571)
Precision (%)	0.891	0.917	0.928	0.941	0.947	0.953
Recall (%)	0.978	0.982	0.989	0.867	0.887	0.909
F-Score (%)	0.933	0.948	0.958	0.902	0.916	0.930
Specificity (%)	0.497	0.623	0.678	0.769	0.791	0.810
Commercial (200 - 299)						
(train, test) data	(905, 17191)	(3620, 14476)	(9048, 9048)	(905, 17191)	(3620, 14476)	(9048, 9048)
Precision (%)	0.609	0.740	0.840	0.423	0.509	0.606
Recall (%)	0.518	0.616	0.672	0.639	0.709	0.755
F-Score (%)	0.560	0.672	0.746	0.509	0.592	0.673
Specificity (%)	0.972	0.981	0.989	0.925	0.941	0.958
Industry (300 - 399)						
(train, test) data	(104, 1970)	(415, 1659)	(1037, 1037)	(104, 1970)	(415, 1659)	(1037, 1037)
Precision (%)	0.778	0.887	0.958	0.471	0.779	0.920
Recall (%)	0.188	0.509	0.689	0.218	0.505	0.709
F-Score (%)	0.303	0.647	0.802	0.298	0.613	0.801
Specificity (%)	1.000	0.999	1.000	0.998	0.999	0.999
Vacant (400 - 499)						
(train, test) data	(322, 6101)	(1285, 5138)	(3211, 3211)	(322, 6101)	(1285, 5138)	(3211, 3211)
Precision (%)	0.623	0.690	0.794	0.328	0.413	0.484
Recall (%)	0.217	0.469	0.557	0.508	0.619	0.721
F-Score (%)	0.322	0.558	0.655	0.399	0.496	0.579
Specificity (%)	0.996	0.994	0.996	0.970	0.975	0.978
Agriculture (500 - 599)						
(train, test) data	(449, 8527)	(1796, 7180)	(4488, 4488)	(449, 8527)	(1796, 7180)	(4488, 4488)
Precision (%)	0.575	0.761	0.898	0.346	0.427	0.526
Recall (%)	0.358	0.525	0.654	0.577	0.672	0.750
F-Score (%)	0.441	0.621	0.756	0.432	0.522	0.618
Specificity (%)	0.989	0.993	0.997	0.956	0.963	0.972
Public (600 - 699)						
(train, test) data	(154, 2920)	(615, 2459)	(1537, 1537)	(154, 2920)	(615, 2459)	(1537, 1537)
Precision (%)	0.695	0.795	0.924	0.356	0.617	0.758
Recall (%)	0.116	0.320	0.449	0.179	0.342	0.482
F-Score (%)	0.199	0.457	0.604	0.238	0.440	0.589
Specificity (%)	0.999	0.999	0.999	0.996	0.997	0.998
Recreational (700 - 799)						
(train, test) data	(35, 646)	(137, 544)	(341, 340)	(35, 646)	(137, 544)	(341, 340)
Precision (%)	0.986	0.957	0.966	0.526	0.739	0.853
Recall (%)	0.108	0.414	0.576	0.170	0.432	0.615
F-Score	0.195	0.578	0.722	0.257	0.545	0.715
Specificity (%)	1.000	1.000	1.000	1.000	1.000	1.000
Transport (800) - 899						
(train, test) data	(48, 902)	(190, 760)	(475, 475)	(48, 902)	(190, 760)	(475, 475)
Precision (%)	0.782	0.950	0.989	0.507	0.799	0.827
Recall (%)	0.187	0.453	0.545	0.245	0.461	0.573
F-Score (%)	0.302	0.613	0.703	0.330	0.584	0.677
Specificity (%)	1.000	1.000	1.000	0.999	1.000	1.000
None - misc (" " and 999)						
(train, test) data	(180, 3417)	(720, 2877)	(1799, 1798)	(180, 3417)	(720, 2877)	(1799, 1798)
Precision (%)	0.775	0.911	0.968	0.423	0.695	0.812
Recall (%)	0.281	0.489	0.618	0.338	0.501	0.641
F-Score (%)	0.412	0.637	0.755	0.376	0.582	0.716
Specificity (%)	0.999	0.999	1.000	0.993	0.996	0.998

Table A - 6. Comparing the metrics of training RF and RUS - RF on all Knox County and testing in Hamilton County.

	Raw Building Extractions	
	RF	RUS-RF
Overall		
Hyperparameters	mtry = 2	mtry = 2
Number of Features	4	4
(train, test) data	(229014, 175987)	(80899, 175987)
Residential (100 - 199)		
n	(185143, 142913)	(37028, 142913)
Precision (%)	0.883	0.894
Recall (%)	0.957	0.742
F-Score (%)	0.919	0.811
Specificity (%)	0.454	0.620
Commercial (200 - 299)		
(train, test) data	(18096, 17007)	(18096, 17007)
Precision (%)	0.429	0.357
Recall (%)	0.339	0.446
F-Score (%)	0.379	0.396
Specificity (%)	0.952	0.914
Industry (300 - 399)		
(train, test) data	(2074, 4331)	(2074, 4331)
Precision (%)	0.128	0.201
Recall (%)	0.018	0.030
F-Score (%)	0.031	0.052
Specificity (%)	0.997	0.997
Vacant (400 - 499)		
(train, test) data	(6423, 5281)	(6423, 5281)
Precision (%)	0.121	0.039
Recall (%)	0.056	0.194
F-Score (%)	0.077	0.065
Specificity (%)	0.987	0.852
Agriculture (500 - 599)		
(train, test) data	(8976, 60)	(8976, 60)
Precision (%)	0.000	0.000
Recall (%)	0.000	0.000
F-Score (%)	0.000	0.000
Specificity (%)	0.984	0.971
Public (600 - 699)		
(train, test) data	(3074, 2178)	(3074, 2178)
Precision (%)	0.025	0.040
Recall (%)	0.007	0.022
F-Score (%)	0.011	0.028
Specificity (%)	0.997	0.993
Recreational (700 - 799)		
(train, test) data	(681, 1141)	(681, 1141)
Precision (%)	0.371	0.041
Recall (%)	0.011	0.006
F-Score	0.022	0.011
Specificity (%)	1.000	0.999
Transport (800) - 899		
(train, test) data	(950, 1749)	(950, 1749)
Precision (%)	0.000	0.000
Recall (%)	0.000	0.000
F-Score (%)	0.000	0.000
Specificity (%)	0.999	0.999
None - misc (" " and 999)		
(train, test) data	(3597, 2178)	(3597, 2178)
Precision (%)	0.005	0.005
Recall (%)	0.004	0.009
F-Score (%)	0.004	0.006
Specificity (%)	0.994	0.986

Table A - 7. Comparing classification metrics of the nine - multiclass problem with removal of 80% of the residential class to the binary problem consisting of only residential and other (with no RUS). Results shown are calculated with a RF consisting of $m_{try} = 2$ and the following variables: total area of all building detections in a parcel, nearest building detection distance, total perimeter of all building detections in a parcel, and number of building detections in a parcel.

	Training RF with 20% of Knox County to test on 80% of Knox County		Training RF with all of Knox County to test on Hamilton County	
Residential (100 - 199)	Nine Multiclass Problem (RUS Residential)	Binary Problem of Residential vs Other	Nine Multiclass Problem (RUS Residential)	Binary Problem of Residential vs Other
(train, test) data	(7405, 148114)	(229014, 175987)	(37028, 142913)	(229014, 175987)
Precision (%)	0.947	0.848	0.894	0.632
Recall (%)	0.887	0.679	0.742	0.518
F-Score (%)	0.916	0.754	0.811	0.569
Specificity (%)	0.791	0.971	0.620	0.930

APPENDIX B

```
# Packages for RF
library(caret)
library(parallel)
library(randomForest)
# Library(UBL) # Package for RUS

# Read in Rdata of Knox County Raw CNN Building Detection
load("Rdata_bd_orignal_unedits/knoxallvaluesbroad.RData")

# Partition Data
set.seed(1)
trainIndex <- createDataPartition(knoxRawCNN$overallllanduse, p=.20,
list=FALSE, times = 1)
knox_final_train <- knoxRawCNN[trainIndex,]

newvariables<-
c("p_bd_area","p_bd_peri","s_bd_area","s_bd_peri","num_bd_in_parcel","
nearestbddist", "num_bd_within20", "medianperi20",
"medianarea20","num_bd_within40", "medianperi40", "medianarea40",
"num_bd_within60","medianperi60", "medianarea60", "num_bd_within80" ,
"medianperi80", "medianarea80", "num_bd_within100",
"medianperi100","medianarea100", "overallllanduse")

knox_final_train<-knox_final_train[newvariables]

# normalize function
normalize <- function(x) {
  return ((x - min(x)) / (max(x) - min(x)))
}

# make data on 0 to 1 scale
x<-as.data.frame(normalize(knox_final_train[,1:21]))
y<-as.factor(knox_final_train[,22])
knox_final_train<-cbind(x,"overallllanduse" =y)

# Conduct RUS on residential where only 20% of this class is kept
# set.seed(1)
# C.perc = list(agr=1, com=1, ind=1,non_misc=1 , pub=1, rec=1,
res=.20, tra=1, vac=1)
```

```

# knox_final_train <- RandUnderClassif(overallllanduse~,
knox_final_train, C.perc)

# set up RFE function in parallel
set.seed(1)
# 5 fold CV so set seed to k + 1 = 6 (will go through 5 folds then run
the last model on the best fold)
seeds <- vector(mode = "list", length = 6)

# For the 5 folds try 15 values for mtry
for(i in 1:5) seeds[[i]]<- sample.int(n=1000, 15)

# Last fold will just have 1 mtry value
seeds[[6]]<-sample.int(1000, 1)

# Parallel
no_cores <- detectCores() - 1
cl <- makeCluster(no_cores, type='PSOCK')
registerDoParallel(cl)

# See R caret package for rfe function details
set.seed(1)
rfRFE<-rfe(overallllanduse~,
data = knox_final_train,
method = "rf",
sizes = c(2, 4, 6, 8, 10, 12, 14, 16, 18, 21), # subset of vars
tuneGrid =expand.grid(.mtry=c(1:15)), # mtry grid
rfeControl = rfeControl(method = "cv", number = 5, functions = caretFu
ncs, seeds = seeds)) # stratified 5 - fold CV

stopCluster(cl)
rfRFE # accuracy of each subset variable in a best k - 1 fold
rfRFE$optVariables # optimal variables
rfRFE$fit # tuning parameter

# Will reload data and run RF on all training data using optimal featu
res and mtry value
# Read in Rdata of Knox County Raw CNN Building Detection
load("Rdata_bd_orignal_unedits/knoxallvaluesbroad.RData")

# Partition Data
set.seed(1)
trainIndex <- createDataPartition(knoxRawCNN$overallllanduse, p=.20, li
st=FALSE, times = 1)
knox_final_train <- knoxRawCNN[trainIndex,]

```

```

knox_final_test <- knoxRawCNN[-trainIndex,]

newvariables<-c("p_bd_peri", "p_bd_area", "nearestbddist", "num_bd_in_
parcel", "overalllanduse")
knox_final_train<-knox_final_train[newvariables]
knox_final_test<-knox_final_test[newvariables]

# Make data on 0 to 1 scale
x<-as.data.frame(normalize(knox_final_train[,1:4]))
y<-as.factor(knox_final_train[,5])
knox_final_train<-cbind(x,"overalllanduse" =y)
x1<-as.data.frame(normalize(knox_final_test[,1:4]))
y1<-as.factor(knox_final_test[,5])
knox_final_test<-cbind(x1,"overalllanduse" =y1)

# Conduct RUS on residential where only 20% of this class is kept
# set.seed(1)
# C.perc = list(agr=1, com=1, ind=1,non_misc=1 , pub=1, rec=1, res=.20
, tra=1, vac=1)
# knox_final_train <- RandUnderClassif(overalllanduse~., knox_final_tr
ain, C.perc)

# Run RF
# method = "none" means using all of training data
control<-trainControl(method = "none")

# set mtry value
fitrf <- train(overalllanduse~., knox_final_train, method='rf', trCon
trol=control,tuneGrid = expand.grid(.mtry=c(2)))

# Predict test data
predictionrf_ordinal<-predict(fitrfr,knox_final_test)
cmrf<-confusionMatrix(predictionrf_ordinal, knox_final_test$overalllan
duse)
# Confusion Matrix
cmrf
# Metrics
cmrf$byClass

```

VITA

Samantha Duchscherer attended the University of North Georgia from August 2010 to December 2013. In three and half years, while having a part time job at XPO Logistics in Gainesville, GA during her last semester, she obtained a Bachelor of Science in mathematics along with a statistics and business administration minor. After graduating, she moved to Knoxville, TN and became a research associate for the Geographic Information Science and Technology (GIST) group at Oak Ridge National Laboratory (ORNL). The GIST group consistently challenged Samantha to learn data science methodologies resulting in her to become proficient in various software programs. However, she had the urge to travel before continuing her education further. Samantha had the opportunity to take a three month break from ORNL to volunteer with the International Volunteer HQ (IVHQ) organization in Guatemala, Ecuador, and Peru. Once she returned to the GIST group, she graciously accepted their graduate research assistantship at the University of Tennessee, Knoxville. She graduated in May 2018 with a Master of Science degree in mathematics and a minor in statistics. Samantha is eager to optimize the application of her skills and education to obtain a career in a data science related position.