environments

MDPI

Article

Evaluation of Light Gradient Boosted Machine Learning Technique in Large Scale Land Use and Land Cover Classification

Dakota Aaron McCarty [1], Hyun Woo Kim [1,*] and Hye Kyung Lee [2]

[1] Department of Urban Policy & Administration, Incheon National University, 119 Academy-ro, Yeonsu-gu, Incheon 22012, Korea; dakota.mccarty@inu.ac.kr

[2] School of Urban Planning and Real Estate Studies, Dankook University, 153 Jukjeon-ro, Suji-gu, Yongin-si, Gyeonggi-do 16890, Korea; hklee@dankook.ac.kr

* Correspondence: kimhw@inu.ac.kr; Tel.: +82-32-835-8874

Received: 21 August 2020; Accepted: 2 October 2020; Published: 3 October 2020

Abstract: The ability to rapidly produce accurate land use and land cover maps regularly and consistently has been a growing initiative as they have increasingly become an important tool in the efforts to evaluate, monitor, and conserve Earth's natural resources. Algorithms for supervised classification of satellite images constitute a necessary tool for the building of these maps and they have made it possible to establish remote sensing as the most reliable means of map generation. In this paper, we compare three machine learning techniques: Random Forest, Support Vector Machines, and Light Gradient Boosted Machine, using a 70/30 training/testing evaluation model. Our research evaluates the accuracy of Light Gradient Boosted Machine models against the more classic and trusted Random Forest and Support Vector Machines when it comes to classifying land use and land cover over large geographic areas. We found that the Light Gradient Booted model is marginally more accurate with a 0.01 and 0.059 increase in the overall accuracy compared to Support Vector and Random Forests, respectively, but also performed around 25% quicker on average.

Keywords: land use; land cover; light gradient boosted; machine learning

1. Introduction

The classification of images acquired by remote sensing in the context of land use mapping consists of assigning each pixel of the image to a class. The set of classes selected to represent the scene of interest constitutes a taxonomy, or nomenclature, which varies according to the needs of the end user. The attribution of these classes is based on the visual analysis specific to the pixel and can be based on its visual description [1]. Existing works all use supervised classification methods [2], based on machine learning algorithms. The supervised term comes from the training step of the algorithm, which consists of modeling the classes in play from a reference data set. After training the data, one can then infer the unlabeled data classes by applying the model. The reference dataset is a set of pixels, described by attributes (observations), whose classes are known in advance. Each class is thus modeled in relation to these attributes, whether they are generative or discriminative methods.

It is important to note that the quality of a classification depends on (a) the choice of attributes to best discriminate between classes and (b) the training data set. Until very recently, discriminative methods based on Support Vector Machines or Random Forests have been preferred over generative methods. They are more flexible as they directly link the attributes with the labels provided via the learning set, thus directly solving the problem that is being evaluated [3]. Random Forest (RF) has become more popular due to its ability to be used for both classification of satellite images as well as regression analysis [4]. Due to its flexibility, RF is perfect for both continuous as well as categorical variables [5].

RF is also quite durable, it has been used in a wide range of uses from modeling forest coverage [6], land use/land cover (LULC), and also more complex object-based image analysis [7]. However, with its flexibility, it also has great accuracy. Rodriguez-Gailano [8] was able to reach an incredible 92% accuracy in his study of RF. Support vector machines (SVM) have also been shown to have the ability to generalize complex classifications, with previous research being able to achieve a 95.2% accuracy [9,10]. Similar to our study, they also chose to utilize Sentinel-2 data.

However, while there have been newer algorithms developed, such as Light Gradient Boosting Machine (LightGBM), not many researchers have utilized them. For a while now, RF and SVM have been adequate enough to perform the tasks assigned, however, as the study area increases in size, it becomes increasingly costly in time and effort to perform the functions. With LightGBM, there is the ability to have rapid and accurate results with the ability to reach accuracy rates of 92.07% [11] when categorizing crop types, a task that is not easily accomplished by older methods.

Additionally, newer open sourced tools, such as AutoML() developed by mljar for use in Python [12], aid in performing feature preprocessing and engineering, algorithm training, and hyper-parameters selection to create less coding need. Utilizing this open source tool allowed us to perform hundreds of different algorithms rapidly and cost effectively and gave us the best tuned parameters for our models.

Motivation and Objectives

Recently, we have been seeing increased amounts of deforestation, agricultural expansion, and overall environmental degradation being pushed by urbanization [13,14]. As well, with the growth of ecosystem engineering changing the landscapes and ecosystems [15], it is increasingly important to have accurate and up-to-date LULC maps to better track these changes and target the ones that could be dangerous or produce negative effects.

We chose to use Sentinel-2 satellite images for multiple reasons: (a) it is a publicly available source provided at no cost, (b) with a spatial resolution of 10m, it is relatively high when compared to Landsat or MODIS images, and (c) with four red-edge bands to measure vegetation is ideal for LULC classification. As well, currently, there are no other studies to our knowledge that utilize this data for performance assessment of old and new machine learning algorithms for LULC classification purposes at larger scales such as regions or country wide.

Thus, this work involves assessing the ability to provide an accurate LULC classification map covering over a 10,000 km^2 area of land, at 10 m spatial resolution. The objective and the evaluation are thus operational in the sense that we seek to propose an alternative solution to Random Forest and Support Vector Machine analysis with a limited configuration and calculation time but still high accuracy while using publicly available, no cost, satellite images with a resolution range of 10 m.

2. Literature Review

Multiple researchers have analyzed and discussed issues related to accuracy in remote-sensing research. For example, Khatami et al. used a supervised technique that included the addition of texture information, which allowed a machine learning algorithm to access additional spatial information to more accurately classify land-cover [2]. However, as found by Khatami, practitioners who rely heavily on maximum likelihood classifiers are bound to achieve lower accuracy levels. This information highlights some of the primary factors that may impact the accuracy of machine learning algorithms and models. By understanding these factors, data scientists can use the information in determining which method would be ideal for analyzing the complex data available [16]. Accordingly, there is a need to establish whether LightGBM meets these requirements, and the features provide the model with an advantage over its traditional counterparts in terms of land use and land cover classification.

Other studies have identified specific considerations that would help in determining the accuracy of a machine learning tool. In their study, Rodriguez-Galiano et al. [8] evaluated how Artificial Neural Networks (ANNs), Regression Trees (RTs), Random Forests (RF) and Support Vector Machines (SVMs) can be used in assessing the availability of minerals in an area [8]. The researchers found that while the Random Forests models performed the best, the other models studied did not necessarily perform poorly. However, when compared to simple vector machines, the Random Forests algorithm proved to more stable and robust. These findings reflect those of other authors, who argued that Random Forests could be more accurate as long as the model is designed to specify random feature subsets [17].

Some scholars have claimed that Random Forests are the most accurate type of classifier. For example, Fernández-Delgado et al. [18] performed a study in which they examined 179 classifiers derived from at least 17 different statistical/methodological families. These families included Random Forests, Support Vector Machines, and Neural Networks, among others. In doing so, the researchers used 121 data sets to determine which model was the most accurate. The study's findings indicated that Random Forests models were the most accurate, achieving 94.1% inaccuracy [18]. The scholars also found that Support Vector Machines were second in terms of accuracy to Random Forests, as one the versions recorded 92.3% in maximum accuracy. These results raise profound questions that data scientists need to consider when selecting the ideal machine learning tools. In our case, we are focusing on accuracy and time.

On the other hand, other researchers have refuted that Random Forests models produce the most accurate results. In their report, Wainberg et al. [19] asserted that the methods of the study that shown Random Forests to have the highest levels of accuracy were biased and, as such, incorrect. In line with the authors' assertions, Fernández-Delgado et al. [18] did not utilize a held-out test in evaluating the various parameters associated with Random Forests. This approach is incorrect because selectively setting the hyperparameters was bound to enhance the performance of the model over others. Wainberg gives great insight for developing a methodology with the aim of comparing algorithms, such as ours.

Furthermore, studies have evaluated some of the critical shortcomings of Random Forests. In their report, Lin et al. [20] claimed that the Random Forest algorithm requires relatively more computational power and other resources compared to other models. The fact that the Random Forest has to create multiple trees and consolidate their input causes substantial demands. Researchers have also emphasized the idea that the appropriate use of Random Forests requires much training on the part of the practitioner [5]. Users who lack adequate knowledge and skills may find it challenging to utilize the model efficiently. In some instances, attributes with more values tend to have a more significant effect on the algorithm, making the resulting attribute weights inaccurate and unreliable. Therefore, data scientists need information about how they can use other models such as LightGBM to avoid and resolve the drawbacks of traditional alternatives.

Similarly, researchers have discussed some of the main benefits and disadvantages of Support Vector Machines. Regarding the model's strengths, Wuest et al. argued that Support Vector Machines could be highly reliable in instances whereby the practitioner has little knowledge about the data. The algorithm has also proven to be effective when used on both structured and unstructured data and is very scalable. However, despite having profound advantages, Support Vector Machines have some notable shortcomings, which may affect the model's accuracy and reliability. Wuest et al. observed that efforts to use Support Vector Machines require an extensive and in-depth understanding of the hyper-parameters and tuning (25). Besides, SVM can be unreliable when used on large data sets [18]. Since minimal research has focused on the strengths and disadvantages of LightGBM, a study that examines this specific machine learning tool is warranted. Moreover, a comparison between already trusted traditional tools such as SVM and RF is useful to open to door to this new method. Indeed, additional research will offer valuable lessons about how practitioners can leverage the benefits of LightGBM in addressing the drawbacks of other models.

3. Study Area

The study area (Figure 1) is a 100 km × 105 km mixed-use landscape in the Rhine–Ruhr Functional Urban Region, spreading from Essen in the north, to the urban areas of the cities in and around

Düsseldorf down south to Cologne. This area was specifically chosen for its diverse geography with many farms and croplands, forests, rivers, cities and villages, and mining facilities. In Figure 2, you can see the split between the western and eastern side of the image between agriculture and forest with the Rhine river splitting the image down the middle with cities and villages lining it. As well, there is clear imagery of mining taking place in the western part of the study area, making it quite diverse in its land use and land cover.

To further demonstrate the wide LULC variety of the area, our study found that 23 of the 25 European Urban Atlas 2018 classes occurred in our study area and those were aggregated down to seven classes (Appendix A, Table A1). These are: (1) Urban: comprising areas that are considered "built-up"; (2) Infrastructure: comprising highways, roads, airports, ports, etc.; (3) Mines, dump and construction sites: comprising mineral extraction spots, construction sites, and sites without any classification; (4) Low density vegetation: comprising urban parks, herbaceous areas, and wetlands; (5) Crops: comprising arable land and pastures; (6) Dense vegetation: comprising forests; (7) Water: comprising lakes and rivers. This variety gives us a great data set to work with.

However, beyond the simple variety of the area, this region has also seen quite significant growth and urbanization over the past few years. Due to this growth, it is now considered a possible threat to the European Union's goal of hitting its target of a 60% reduction in greenhouse gasses by the year 2050 [21]. With the region being an economic hub of Germany, and Europe, it is an attractive place to move to due to its strong economic security. This being so, it is important to have up to date LULC maps for these regions to better aid in tracking changes and allow planners and stakeholders to better plan and make quicker decisions with more reliable and up-to-date data.

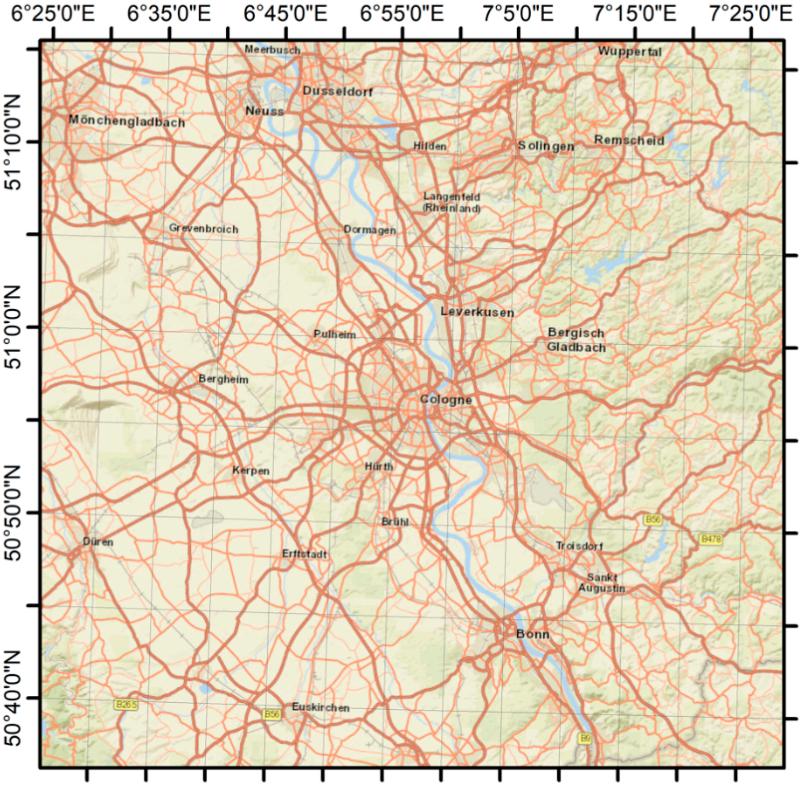


Figure 1. The Rhine–Ruhr metropolitan region. (Esri, HERE, Garmin, USGS, Intermap, INCREMENT P, NRCan, Esri Japan, METI, Esri China (Hong Kong), Esri Korea, Esri (Thailand), NGCC, (OpenStreetMap contributors, and the GIS User Community [22]).

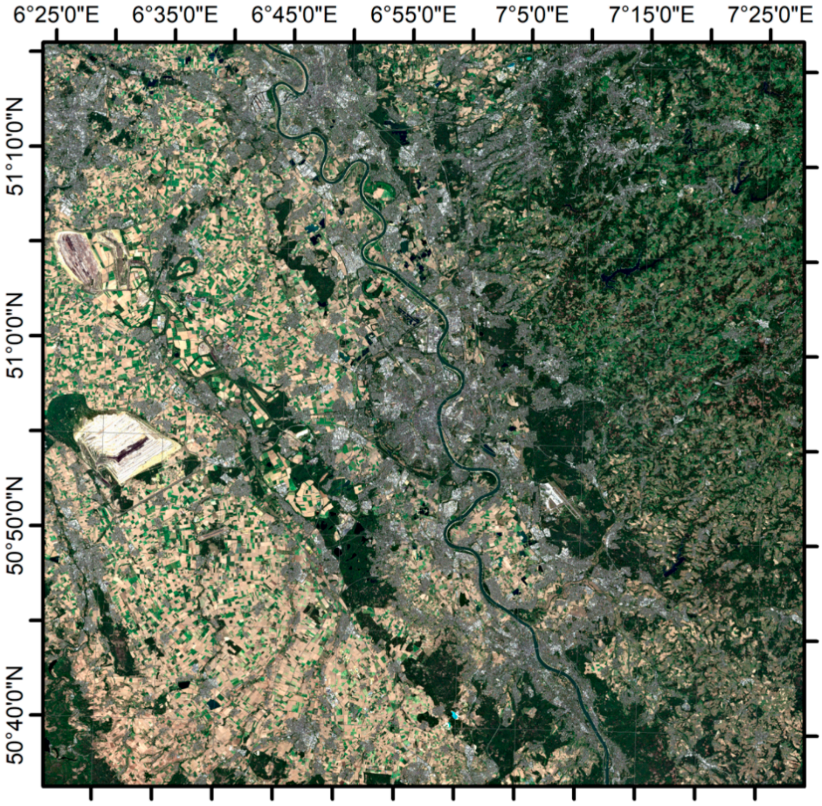


Figure 2. Satellite image of the study area in true color.

4. Data

Sentinel-2 Data

We chose to use the Sentinel-2 L2A data as it is already atmospherically corrected using the Sen2Cor processor and PlanetDEM Digital Elevation Model (DEM) with 10 m, 20 m, and 60 m spatial resolutions, depending on the spectral band [23]. Among all of the publicly available satellite images, Sentinel-2 has the highest spatial resolution at 10 m, making it a good choice for classification projects. For our scene selection, we used the Earth Observing System (EOS) LandViewer [24] tool to filter scenes that were of the L2A dataset with less than 1% cloud coverage for 5 August 2020 (Figure 2). Our date selection was also intentional; we wanted to show that the recent satellite images captured could be used for LULC with minimal data manipulation and cleaning. The satellite imagery was downloaded directly from the EOS LandViewers platform on 10 May 2020. Eleven bands at various wavelengths of the electromagnetic spectrum were selected for use in our classification procedure (Table 1). Each of the 20 m and 60 m resolution bands were resampled down to 10m using ESRI ArcMap and bilinear interpolation.

Table 1. Sentinel-2 Bands with wavelength and resolution [25].

Sentinel-2 Bands	Central Wavelength (µm)	Resolution (m)
B1—Coastal aerosol	0.443	60 [1]
B2—Blue	0.490	10
B3—Green	0.560	10
B4—Red	0.665	10

Table 1. *Cont.*

Sentinel-2 Bands	Central Wavelength (µm)	Resolution (m)
B5—Vegetation red edge	0.705	20 [1]
B6—Vegetation red edge	0.740	20 [1]
B7—Vegetation red edge	0.783	20 [1]
B8—NIR	0.842	10
B8A—Vegetation red edge	0.865	20 [1]
B9—Water vapor	0.945	60 [1]
B11—SWIR	1.610	20 [1]
B12—SWIR	2.190	20 [1]

[1] Resampled to 10 m.

5. Methods

5.1. Study Design and Sample Selection

Training the data is an immensely important part of any supervised machine learning algorithm and requites a large amount of good data. Moreover, satellite imagery has its own complexities and can be a challenging task to collect. It can be difficult to obtain a representative training dataset that is an acceptable resolution and covers a large enough area to adequately train the model [26]. Even more, to obtain adequate reference data that is up to date with your current data can be quite costly and restrictive. Due to that, it has become common to use older LULC maps and data to classify past satellite images and then apply that model to more modern images. For instance, by using thematic maps from the 1970s as a training dataset to classify satellite (Landsat-1) imagery from around the same time period in Vietnam, Tran [27] was able to build a classification model that worked well on more recent 2015 datasets. Another new approach that has been developing in more recent studies is that of using city/government-provided LULC maps as a reference for their training dataset [28–30]. One great benefit of this is that land cover maps with a high enough quality and accuracy have the ability to produce a very large amount of training samples for many different features [30].

The design of our methodology (Figure 3) first involved collecting the satellite images for our study areas; for that we used the Sentinel-2 satellite images collected from EOS LandViewer [24]. After the images were collected, we then processed them in ArcMap by first generating a fishnet over the raster files with a 30m resolution corresponding with the raster pixels themselves, and then used the feature extraction tool to give each point the value of each of the spectral bands, as well as utilizing the calculate geography tool to also add an x and y coordinate. After, we followed a similar approach as mentioned previously [30] and used the previously existing European Urban Atlas LULC maps for the 2018 year. These maps provide reliable, inter-comparable, high-resolution land use and land cover data for a large part of Europe [31]. This data was then also extracted to the previously created points to complete our dataset with x and y coordinates, all spectral bands, and now LULC classifications.

With the European Urban Atlas data, we were able to obtain 27 separate LULC classifications, which we then categorized into seven separate categories: urban; infrastructure; mines, dump and construction sites; low density vegetation; high density vegetation; crops; and water. We chose to use a sample size of 500 for each category, taking advantage of the large number of available data. Furthermore, to avoid any possible misclassification issues due to changed LULC types between 2018 and 2020, each sample from both years was analyzed and compared against RGB and false color composites of the Sentinel-2 image to pick out any samples that have undergone change between the years, meaning that the LULC in 2018 would not be the same as for today. If a change was detected, then a new sample was chosen.

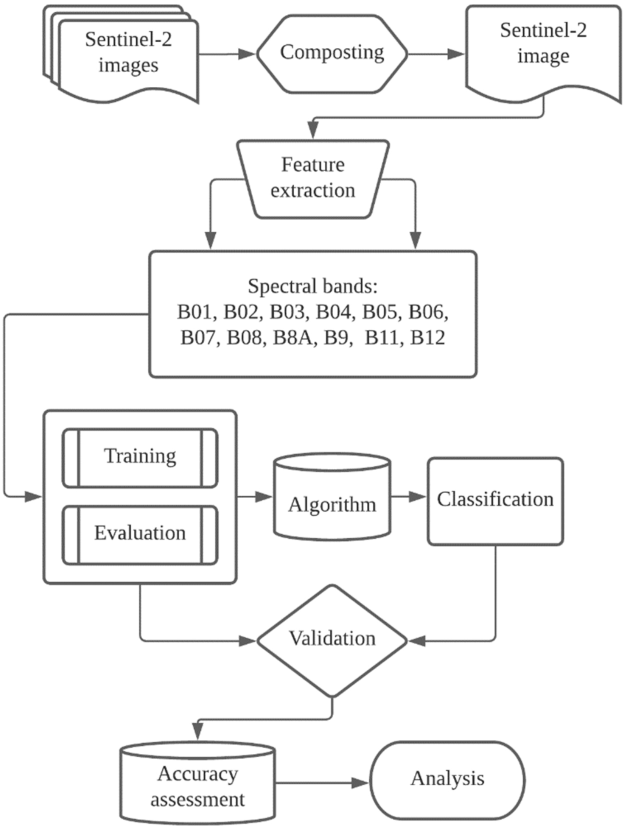


Figure 3. Flow chart of methodology.

Twenty-three of the 25 European Urban Atlas 2018 classes occurred in our study area and those were aggregated down to seven classes (Appendix A, Table A1). These are: (1) Urban: comprising areas that are considered "built-up"; (2) Infrastructure: comprising highways, roads, airports, ports, etc.; (3) Mines, dump and construction sites: comprising mineral extraction spots, construction sites, and sites without any classification; (4) Low density vegetation: comprising urban parks, herbaceous areas, and wetlands; (5) Crops: comprising arable land and pastures; (6) Dense vegetation: comprising forests; (7) Water: comprising lakes and rivers.

Next, we increased our sample size further to keep up with the data-driven framework of our machine learning models [32]. We used a stratified random sampling method producing and increased 4600 samples per class. This number was selected after running multiple models and stopping at the point the model accuracy rate became flat, meaning that more samples did not really increase the accuracy. We did this in consideration of computing power and time. The 32,200 samples were then split into two portions: a training dataset making up 70% (22,540) of the samples and an evaluation dataset comprising the remaining 30% (9660). Each LULC class was assigned the same number of training (3220) and evaluation (1380) samples (Appendix A, Table A1).

Figure 4 gives a few examples of our training area comparing the European Urban Atlas LULC to the real color satellite images of the same area after we performed the first training and evaluation of the area. For these, 1a/2a shows our methods of classifying infrastructure, with multiple highways, train tracks, and train stations in the image. Image 1b/2b is used to show the classification of urban areas and how the classification is able to distinguish between urban areas and infrastructure to an okay extent. Image 1c/2c shows the classification of a large mine that is present in the study area. Image 1d/2d demonstrates the classification of low-density vegetation in the area, especially those near urban areas that are many times park; in this specific image (2d), you can see that this low-density urban area has infrastructure going through it and is bordered by crops. Image 1e/2e shows the classification of dense vegetation, in this case a forest. Image 1f/2f shows the classification of water bodies, the highest performing classifier. Lastly, 1g/2g shows the classification of crops.

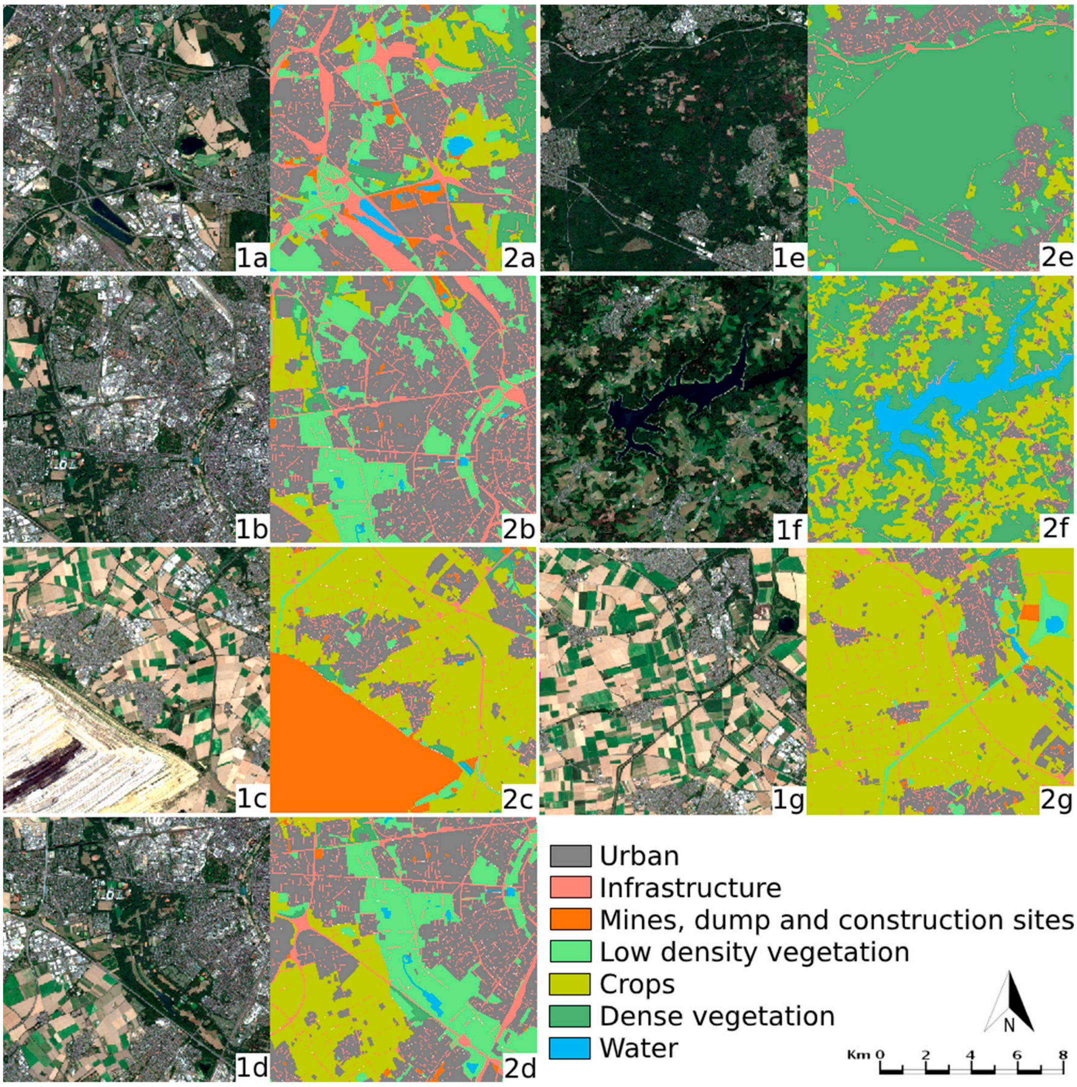


Figure 4. Comparison between the real color satellite image and the LULC classification for the first training and evaluation.

5.2. *Machine Learning Algorithms*

Continuing with our model in Figure 3, after collecting our satellite data and processing the raster files in ArcMap to convert the images into a more processable dataset, we then began the algorithm and classification parts of our study. This following section provides more detailed descriptions of the various algorithms that we used for our study.

5.2.1. Random Forests (RF)

RF is an ensemble learning algorithm working off the theory that a combination of bootstrap aggregated classifiers performs better than a single classifier [4]. In this case, bootstrap means that each individual tree is parameterized using a randomly sampled set of observations from the training dataset [33]. By doing this, there is a reduced risk of multicollinearity since the trees are naturally de-correlated via this process. RF works by creating several decision tree models based on different groupings from the input variables from the training data, resulting in an output that is an unweighted majority vote of each class that is averaged across all of the produced trees. Breiman (2001) defines Random Forest as:

"a classifier consisting of a collection of tree-structured classifiers {h(x, k), k = 1,...} where the {k } are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input x" [4],

He provides a more in-depth mathematical explanation and proofs of RF in his 2001 paper [4]. In our paper, we utilized Python Scikit-Learn to perform our Random Forest analysis. With the final parameters:

```
params: {
max_features: 0.5,
min_samples_split: 4,
min_samples_leaf: 5
}
```

5.2.2. Support Vector Machines (SVM)

SVM is a supervised learning technique that is frequently used with research related to remote sensing. The SVM algorithm works by finding the "decision boundary"; this boundary is better referred to as the hyperplane and it is used to differentiate the classification problem into a predefined set of separate classes that are in line with the provided training data. Through numerous iterations, the algorithm attempts to find the optimum hyperplane boundary so that it can better distinguish patterns in the training data. When these patterns are found, the algorithm then applies them to the evaluation dataset to check its accuracy [10]. There are different kernels through which the hyperplane boundary can be defined. In our research, we use a radial basis function kernel due to some of the Sentinel-2 spectral bands not being linearly separable. For a more detailed mathematical explanation of this algorithm, please refer to Cortes [34]. In our paper, we utilized Python Scikit-Learn to perform our Support vector machine analysis, with the final parameters:

```
params: {
C: [10, 100, 1000],
gamma: [0.1, 1, 10],
kernel: ['rbf', 'poly', linear']
}
```

5.2.3. Light Gradient Boosting

The typical gradient boosting machine (GBM) constructs an additive model of simple decision trees that are not very well optimized and then generalizes them by optimizing an arbitrarily defined loss function to make stronger predictions [33]. At the moment, a widely used GBM, especially in remote sensing, is Extreme Gradient Boosting (Xgboost). The utility behind Xgboost that does not exist with other algorithms is that it builds an objective function, which combines the loss function and a regularization term that controls model complexity. This enables parallel calculations and the maintenance of optimal computational speed. While Xgboost has been successful [35,36], we chose to take a different approach and analyze Light Gradient Boosted Machine (LightGBM) for a few key reasons: (1) it has a faster training speed and higher efficiency than many other algorithms. LightGBM uses a histogram-based algorithm i.e., it buckets continuous feature values into discrete bins which fastens the training procedure. (2) It has lower memory usage by replacing continuous values to discrete bins. (3) It has better accuracy than any other boosting algorithm. It produces much more complex trees by following a leaf wise split approach rather than a level-wise approach, which is the main factor in achieving higher accuracy. (4) It has better compatibility with large datasets. It is capable of performing equally well with large datasets, with a significant reduction in training time, as compared to Xgboost. A more detailed mathematical explanation of LightGBM is provided in Ke [37]. This study utilized Python Scikit-Learn to perform our Light Gradient Boosting machine analysis, with the final parameters:

```
params: {
num_leaves: 1024,
bagging_freq: 3,
objective: regression,
bagging_fraction: 0.3,
learning_rate: 0.005,
feature_fraction: 1
},
```

5.3. Model Training and Tuning

Optimization, i.e., tuning, of the parameters was an important part of the training process in order to get the best possible outcome. For this, we utilized mljar, an open-sourced program [12,38] developed for training and testing machine learning models. With mljar's AutoML() function, we were able to train and test hundreds of models while fine-tuning the parameters along the way. In the case of our chosen algorithms, we chose to optimize our parameters by using randomized sampling of all hyper-parameter combinations up to a certain number of iterations [39]. In order to utilize AutoML(), the best format for the data was a csv file that could then be used by Python Pandas to create a data frame. As mentioned in Section 5.1, we were able to easily accomplish changing the images into a more usable dataset by processing the satellite image raster files in ArcMap through a process of first generating datapoints via the fishnet function to create a net of points corresponding with the images 30m resolution, and then doing a simple feature extraction to get the spectral band values as well as the LULC data, before calculating the x and y coordinates. In our dataset, our independent variables were the spectral bands and our dependent variable was the LULC class value (Appendix A).

For the purpose of AutoML(), our training dataset was split (via random sampling) into multiple sets of equal sized blocks (k); of those, a single one was kept for later validation, leaving a total sample size equal to $k - 1$. This procedure was repeated multiple times to reduce the variance in the model [39,40]. We chose to use a 10-fold cross-validation with 5 repeats to help us reach a suitable balance between ensuring a robust model and a lower computational time, to help us get the maximum number of accurate and highly tuned models in the shortest time. However, after utilizing AutoML() to find the best parameters, we went with a 70/30, training and testing, split for our final models, as mentioned in Section 5.1, as it produced a more accurate results.

5.4. Accuracy and Misclass Assessment

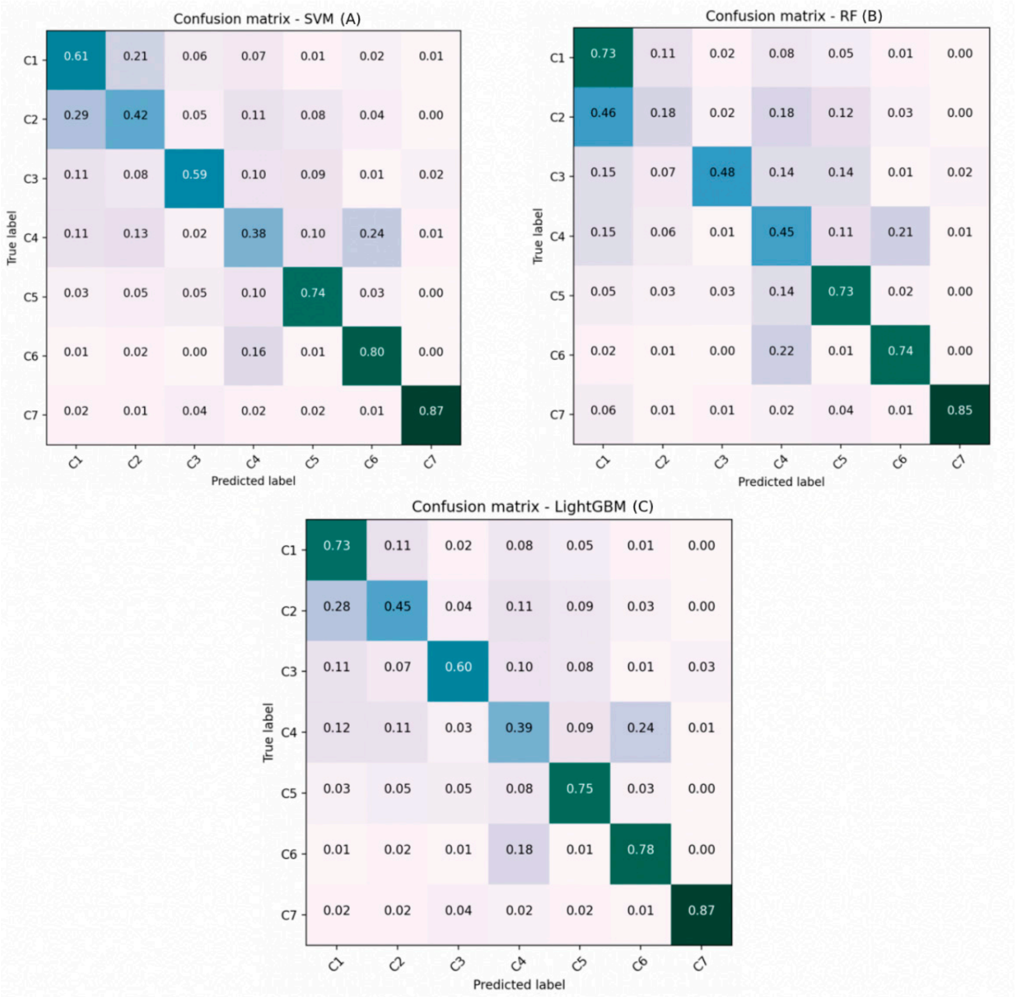
In Table 2, you can see a breakdown of the Overall Accuracy (OA), Kappa, Producer's Accuracy (PA), and User's Accuracy (UA), with UA corresponding to error of commission (inclusion) and PA corresponding to error of omission (exclusion). From looking at the OA and Kappa coefficient, it can be seen that the LightGBM has the highest accuracy among the three at a 65.3% OA and 0.596 Kappa. This compared to SVM with 64.2% and 0.583, followed by RF with 59.4% and 0.527. The accuracy and misclass of each algorithm are represented in Figure 5 with confusion matrices. In that matrix, it seems that the hardest classes to classify for the algorithms as a whole were C2, C3, and C4 (Infrastructure; Mines, dump, and construction sites; and Low-density vegetation).

Table 2. Algorithm OA, Kappa score, PA, and UA.

	Class	**Producer's Accuracy**	**User's Accuracy**
SVM **OA = 0.642, Kappa = 0.583**	Urban	0.61	0.56
	Infrastructure	0.42	0.45
	Mines, dump and construction sites	0.59	0.72
	Low density vegetation	0.42	0.40
	Crops	0.74	0.70
	Dense vegetation	0.80	0.69
	Water	0.87	0.95

Table 2. *Cont.*

	Class	Producer's Accuracy	User's Accuracy
RF **OA = 0.594, Kappa = 0.527**	Urban	0.73	0.45
	Infrastructure	0.18	0.38
	Mines, dump and construction sites	0.47	0.84
	Low density vegetation	0.45	0.36
	Crops	0.73	0.60
	Dense vegetation	0.74	0.71
	Water	0.85	0.96
LightGBM **OA = 0.653, Kappa = 0.596**	Urban	0.73	0.56
	Infrastructure	0.45	0.54
	Mines, dump and construction sites	0.60	0.75
	Low density vegetation	0.39	0.40
	Crops	0.75	0.68
	Dense vegetation	0.77	0.70
	Water	0.87	0.95



Confusion matrix - SVM (A)

	C1	C2	C3	C4	C5	C6	C7
C1	0.61	0.21	0.06	0.07	0.01	0.02	0.01
C2	0.29	0.42	0.05	0.11	0.08	0.04	0.00
C3	0.11	0.08	0.59	0.10	0.09	0.01	0.02
C4	0.11	0.13	0.02	0.38	0.10	0.24	0.01
C5	0.03	0.05	0.05	0.10	0.74	0.03	0.00
C6	0.01	0.02	0.00	0.16	0.01	0.80	0.00
C7	0.02	0.01	0.04	0.02	0.02	0.01	0.87

Confusion matrix - RF (B)

	C1	C2	C3	C4	C5	C6	C7
C1	0.73	0.11	0.02	0.08	0.05	0.01	0.00
C2	0.46	0.18	0.02	0.18	0.12	0.03	0.00
C3	0.15	0.07	0.48	0.14	0.14	0.01	0.02
C4	0.15	0.06	0.01	0.45	0.11	0.21	0.01
C5	0.05	0.03	0.03	0.14	0.73	0.02	0.00
C6	0.02	0.01	0.00	0.22	0.01	0.74	0.00
C7	0.06	0.01	0.01	0.02	0.04	0.01	0.85

Confusion matrix - LightGBM (C)

	C1	C2	C3	C4	C5	C6	C7
C1	0.73	0.11	0.02	0.08	0.05	0.01	0.00
C2	0.28	0.45	0.04	0.11	0.09	0.03	0.00
C3	0.11	0.07	0.60	0.10	0.08	0.01	0.03
C4	0.12	0.11	0.03	0.39	0.09	0.24	0.01
C5	0.03	0.05	0.05	0.08	0.75	0.03	0.00
C6	0.01	0.02	0.01	0.18	0.01	0.78	0.00
C7	0.02	0.02	0.04	0.02	0.02	0.01	0.87

Figure 5. Error matrices showing correct and incorrect prediction percentages for each machine learning algorithm. C1–C7 match with the LULC category, as shown in Appendix A, Table A1. Support Vector Machines—(**A**), Random Forests—(**B**), Light Gradient Boosting Machine—(**C**). With the color gradient going from white (no prediction) to dark green (high number of predictions).

By using a simple, but powerful, Chi-squared test [41] we can compare (statistically) the error matrices by investigating the equality in the overall distributions of variables predicted by one algorithm compared to another. As well, a two proportion *Z*-test [42] was used to compare the proportions of correctly classified pixels for two algorithms at a time. With a null hypothesis that there will be no difference between the algorithms, this will be another method for us to check if there are any statistically significant similarities between the algorithms. For both tests, we will be using a 0.05 confidenceinterval.

6. Results and Discussion

6.1. Comparing the Alogrithms

The highest OA was produced by LightGBM (0.653), closely followed by SVM (0.642) and then RF (0.594) (Table 2). All three algorithms produced roughly similar maps that did a decent job of representing the area (Figure 6). In order to limit class biases by OA, all seven LULC classes had the same number of training and evaluation samples [43]. It is possible to obtain higher OA values if the classes are not standardized with respect to samples. When using an imbalanced class distribution, the accuracy of more well-represented classes increases the OA. However, we believe that this focus on OA negates the individual performance of the classes and does not fully account for the lesser represented classes in the study [44]. With that as a consideration, sampling design decisions should focus on whether the objective of the study is to find the highest OA possible regardless of distribution of class or to equally represent all classes being researched. In our case, with comparing multiple algorithms, we chose the latter.

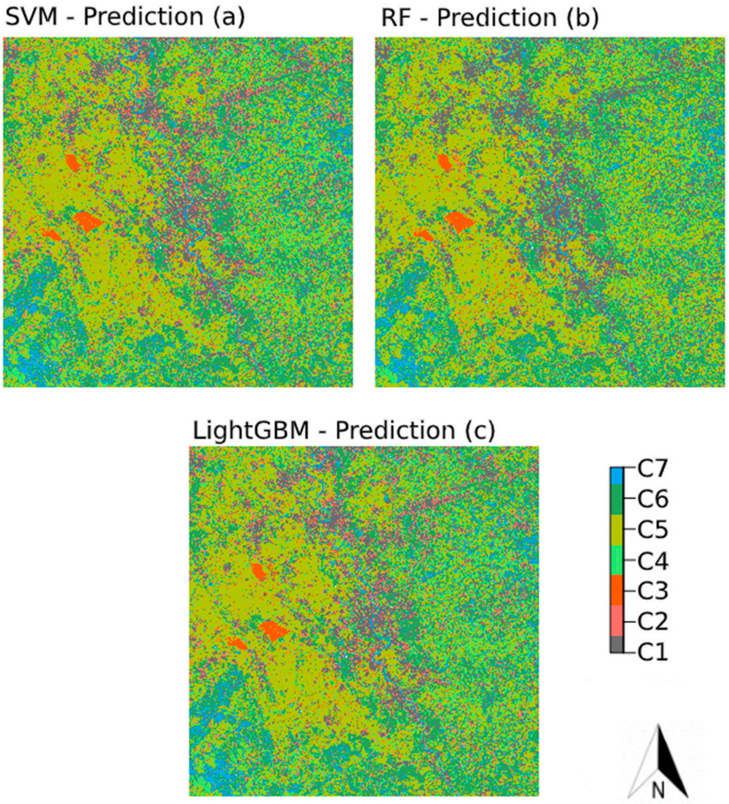


Figure 6. Produced prediction maps for the three algorithms (LightGBM, SVM, and RF), with Figure 2 being the reference satellite image.

The lowest PA across all three algorithms was for Infrastructure (Table 2), which highlights the difficulty in distinguishing these classes from the rest. This class comprised many subclasses including roads, highways, train tracks, airports, shipping ports, etc. From looking at the confusion matrix (Figure 5), it can be seen that the biggest confusion with Infrastructure (C2) was with Urban classes (C1). This is likely due to similarities in materials between infrastructure and urban areas having similar spectral reflectance. The Random Forest algorithm demonstrates this best; it incorrectly predicted Urban areas as Infrastructure 46% of the time, it's largest misclass percentage, and correctly predicted Infrastructure 18% of the time. This error could have been minimized with the inclusion of land surface temperatures [45]; however, there is currently no thermal imagery on the Sentinel-2 satellite. This should be a consideration in future study, to develop more of a hybrid approach utilizing multiple data sources.

Looking at Figure 5, each of the three algorithms appeared to have an easier time classifying natural and green spaces such as water, dense vegetation, low density vegetation, and crops, with an average PA of 0.86 for Water (C7), 0.77 for Dense vegetation (C6), and 0.74 for Crops (C5). This is likely due in large part to the availability of the four separate red edge bands provided by Sentinel-2, making it able to detect vegetation at an increased level of accuracy. In the case of low density vegetation (mean PA = 0.42), where the data was harder to separate as the class borders and connects with other classes frequently (i.e., bordering the Dense vegetation, being contained in the Urban class, etc.), the algorithms had a harder time deciding which was the true class [34].

McNemar's Chi-square test results (Table 3) showed that 42.9% of the classification predictions between paired algorithms were significant at p = 0.01; 14.3% at p = 0.05; and 4.8% at p = 0.10. Additionally, 38.1% of pairings were not statistically significant. All of the pairings for C5 (Crops) were statistically non-significant as well as the classes C2, C3, C4, C5, and C7 for the LightGBM vs. SVM pairing. These non-significant values show that there is no significant difference between the accuracy of the paired models. In our case, all models performed equally in regard to the accuracy of classifying Crops (C5). As well, LightGBM and SVM have only slight differences (at p = 0.1) in performance when classifying Urban (C1) and Dense vegetation (C6). Furthermore, LightGBM and SVM both have statistically high differences in accuracy when compared to RF. This information, paired with the confusion matrix (Figure 5), leads us to believe that, in this case, RF is the less adequate model for this specific project.

Table 3. McNemar's chi-squared test (χ^2) with associated probability (p). *** = $p \leq 0.01$, ** = $p \leq 0.05$, * = $p \leq 0.1$, NS = Not Significant.

	Result	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**
LightGBM vs. RF	χ^2	46.745	100.947	55.125	8.256	1.76	6.282	5.263
	p	***	***	***	***	NS	***	**
LightGBM vs. SVM	χ^2	2.481	1.215	1.306	0.125	1.161	3.361	0.1
	p	*	NS	NS	NS	NS	*	NS
RF vs. SVM	χ^2	30.862	88.506	43.5224	10.803	0.093	20.023	5.882
	p	***	***	***	***	NS	***	**

The Z-test results (Table 4) showed that the proportions of correctly classified pixels of two of the algorithm pairs (LightGBM vs. SVM (χ^2 = 3.19) and RF vs. SVM (χ^2 = 3.06)) were not statistically significant, indicating that there is not a difference in the proportion of correctly classified pixels between the pairs. Based on these results, we can infer that the proportions of correctly classified pixels produced by SVM are different from those correctly classified by RF and LightGBM. For the LightGBM vs. RF pairing, the null hypothesis was not rejected, meaning their accuracies are similar.

Table 4. A two-proportion z-test at p < 0.05.

	Z-Test Results	
LightGBM vs. RF	χ^2 p	0.123 0.45
LightGBM vs. SVM	χ^2 p	3.19 0.00
RF vs. SVM	χ^2 p	3.06 0.00

6.2. Variable Importance Metrics

As part of the output, each of the three algorithms provides variable importance charts (Appendix B, Figures A1–A3.). In Appendix B, Figures A1 and A2 the Shapley Additive Explanation (SHAP) score is used to explain the importance of the SVM (Figure A1) and LightGBM (Figure A2). SHAP measures the impact of variables, taking into account the interaction with other variables. On each chart, the y-axis represents each band of the Sentinel-2 satellite image used in the analysis, in order of importance from top to bottom. On the x-axis is the SHAP value, which indicates the extent of the change in log-odds. From this information, we can extract the probability of success. For Figure A1, SVM, it is shown that B12 (SWIR2), B03 (Green), B11 (SWIR2), B8A (NIR narrow), and B02 (Blue) are the top five bands, with the other bands not falling too far behind in importance. For Figure A2, LightGBM, the top five bands were B09 (Water vapor), B01 (Coastal aerosol), B12 (SWIR2), B11 (SWIR1), and B04 (Red). Both models had quite different importance applied to each band despite having similar results in the end. We were surprised to not see more utilization of the NIR (near infrared) bands; however, the SWIR (short-wave infrared) bands both seemed important for both models. In Figure A3, RF used a different method to categorize the importance of each of the bands, utilizing a more traditional approach of percentages. The top bands for this model were B01 (Coastal aerosol), B11 (SWIR1), B12 (SWIR2), B09 (Water vapor), and B04 (Red). These bands aligned very closely with those in the LightGBM model; however, as mentioned before, the RF model performed less precisely than the LightGBM model. However, in all three models, SWIR bands seemed to hold importance in the predictions.

The high importance of the SWIR bands could be due to the large amounts of forests and foliage in the study area [46]. The Sentinel-2's SWIR bands are centered at 1610 nm (B11) and 2190 nm (B12), allowing them to more easily and accurately detect subtle differences in water content between different types of foliage [47]. With that ability, it is easier for the models to use that band when working in areas with larger tree coverage.

6.3. Calculation Times

With OA's. of 0.653, 0.642, and 0.594 for the LightGBM, SVM, and RF models, respectively, there was not much variation in the accuracy of each model. However, when it came to calculation times, we noticed more difference (Table 5). The quickest model was LightGBM at 287 s vs. SVM at 367 s and RF at 410 s. These processing times are based on using an 8 CPU, 15gb RAM machine to process the data. With these results, we see that LightGBM performed 1.33 min faster than SVM (21.8% decrease) and 2.05 min faster than RF (30% decrease).

Table 5. Process time for each model in seconds.

Model	Process Time (s)	Machine Size
LightGBM	287 s	8 CPU and 15GB RAM
SVM	367 s	8 CPU and 15GB RAM
RF	410 s	8 CPU and 15GB RAM

7. Conclusions

Compared to other available satellite imagery devices, Sentinel-2 is very useful with its four separate red edge bands (B05, B06, B07, B8A) that are very useful and intelligent for observing vegetation at a comparably high spatial resolution (10 m). To the researcher's knowledge, this paper is one of the first to utilize the Light Gradient Boosting (LightGBM) model to classify land-cover and land-use using Sentinel-2 data over a large and varied landscape. Our study compares three separate models, two of which, Support Vector Machines and Random Forests, are widely used today in remote sensing research. The other, Light Gradient Boosting, has been growing in popularity in other fields but has not yet been widely utilized in urban sciences or remote sensing.

The three algorithms yielded near similar overall accuracies with LightGBM having the highest OA of 0.653, closely followed by SVM at an OA of 0.642 and then RF at 0.594. When comparing the classifier accuracies with a Z-test, we found that two-thirds of the algorithm pairings were statistically distinct from one another. Likewise, with the McNemar's test, we found that 67% of the class-wise predictions were statistically significant. Finally, the variable importance metrics show that there is a clear importance with the SWIR bands, which can likely be described by the study area's makeup. However, despite the similarities in accuracies, LightGBM was able to more quickly process the model in each instance, performing 25.9% quicker on average.

When compared to previous studies, our models did not produce accuracy scores that were as high. Those such as Rodriguez-Gailano [8] attained an accuracy of 92% using an RF model. Neither were they as high as other researchers who were able to reach over 95% accuracy with their SVM model [9,10], or compared to previous researchers who were able to get accuracies of 92.07% [11] when categorizing crop types using LighGBM. However, these previous studies were not the same scale as our study in both size and scope. The aim of our study was to generate large scale LULC maps quickly and cheaply (cost wise and with minimal variables) and compare multiple methods to determine if LightGBM is a valid contender, while, on the other hand, these other studies were more focused on specific case examples that have more specific and direct classification needs. However, this does present one limitation to our study, the inability to focus more specifically on a single case or class, as our models are built to be more generalized and focused on classifying multiple, vastly different, classes.

In studies, like ours, that are comparing multiple machine learning algorithms, it is important to avoid the introduction of accidental bias in the analysis. To lessen that issue, each algorithm should undergo equally robust hyper-parameter selection. If one model is calibrated with prior knowledge of the best parameters for that model whilst the others are calibrated more randomly or by using just default parameters, there is a large chance that the comparison will not be accurate, and bias may be present. For that, the use of random iterations over a defined number of parameter combinations, as we have done in this research, can be used to eliminate potential bias; however, coming at the extends the algorithms not reaching optimum accuracies. To be better able to reach higher accuracies within the models, we suggest increasing the number of iterations (however, this can significantly increase the computing requirement) or assessing each model individually to find the most suitable thresholds for the parameters to get the best result possible.

While we did our best to control for any issues or limitations, there are a few worth mentioning. The first being that of computational power, a typical limitation in ML research. While we were able to maintain a relatively low computational load, there is always room for improvement in this area. As the study area increases in size, this limitation becomes increasingly prominent. As well, another obvious limitation is our satellite image resolution. However, as mentioned multiple times in the paper, we intentionally chose our satellite images due to access and cost aspects of obtaining more high-resolution images. In future studies, it would be valuable to use higher resolution satellite images; however, we still believe that the cost and access should remain variables when selecting the images to ensure the widest available usability of the research.

Furthermore, we did have a few other limitations that could be solved more readily. Our models all had a difficult time in distinguishing between water and areas of dense, healthy forests. This could have been overcome by incorporating more variables into the study such as topography or DEM data points. As well, future research could benefit by incorporating indices such as normalized difference vegetation index (NDVI), normalized difference water index (NDWI), and normalized difference built-up index (NDBI) into the models to help further differentiate between the water and vegetation and built-up areas.

Author Contributions: Conceptualization, D.A.M. and H.W.K.; Data curation, D.A.M.; Formal analysis, D.A.M.; Investigation, D.A.M.; Methodology, D.A.M.; Project administration, H.W.K.; Writing—original draft, D.A.M.; Writing—review & editing, H.W.K. and H.K.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Research Foundation (NRF) of Korea (Grant #: NRF-2019R1G1A1010318).

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. LULC classification with total pixel count for the sample area.

LULC Type	Class ID	Urban Atlas Class	Total Pixels	Pixels (Training/Evaluation)
Urban	1	11100: Continuous Urban fabric 11210: Discontinuous Dense Urban Fabric 11220: Discontinuous Medium Density Urban Fabric 11230: Discontinuous Low-Density Urban Fabric 11240: Discontinuous very low-density urban fabric 11300: Isolated Structures 12100: Industrial, commercial, public, military and private units	216,422	3220/1380
Infrastructure	2	12210: Fast transit roads and associated land 12220: Other roads and associated land 12230: Railways and associated land 12300: Port areas	54,568	3220/1380

Table A1. *Cont.*

LULC Type	Class ID	Urban Atlas Class	Total Pixels	Pixels (Training/Evaluation)
Mines, dump and construction sites	3	13100: Mineral extraction and dump sites 13300: Construction sites 13400: Land without current use	49,441	3220/1380
Low density vegetation	4	14100: Green urban areas 14200: Sports and leisure facilities 32000: Herbaceous vegetation associations 40000: Wetlands	53,722	3220/1380
Crops	5	21000: Arable land 23000: Pastures	474,464	3220/1380
Dense vegetation	6	31000: Forests	149,775	3220/1380
Water	7	50000: Water	28,908	3220/1380

Appendix B

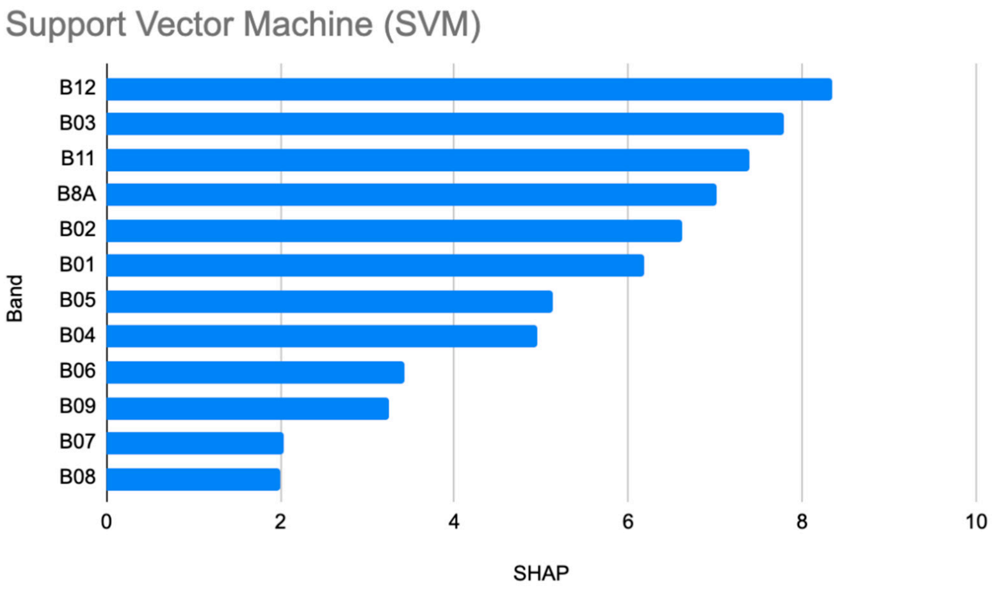


Figure A1. Support Vector Machine (SVM) on SHAP score.

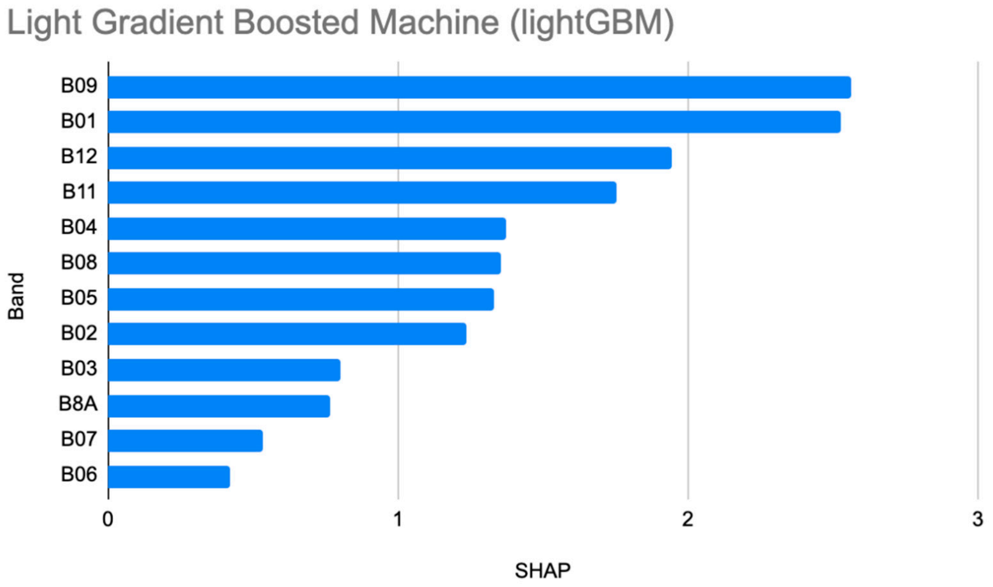


Figure A2. Light Gradient Boosted Machine (LightGBM) on SHAP score.

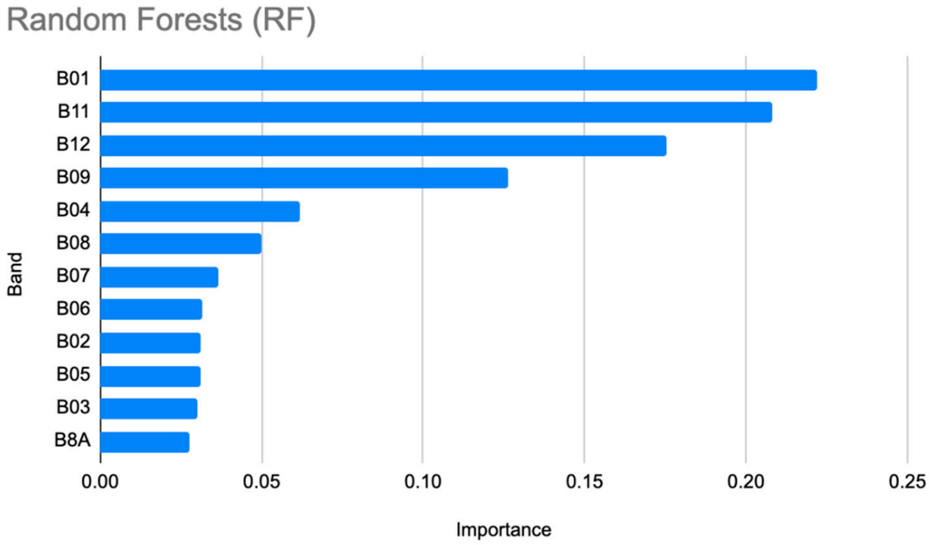


Figure A3. Random Forests (RF) on importance score.

References

1. Remote Sensing Imagery. Wiley. Available online: https://www.wiley.com/en-us/Remote+Sensing+Imagery-p-9781848215085 (accessed on 17 August 2020).
2. Khatami, R.; Mountrakis, G.; Stehman, S.V. A meta-analysis of remote sensing research on supervised pixel-based land-cover image classification processes: General guidelines for practitioners and future research. *Remote Sens. Environ.* **2016**, *177*, 89–100. [CrossRef]
3. Ng, E.; Chen, L.; Wang, Y.; Yuan, C. A study on the cooling effects of greening in a high-density city: An experience from Hong Kong. *Build. Environ.* **2012**, *47*, 256–271. [CrossRef]
4. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
5. Woznicki, S.A.; Baynes, J.; Panlasigui, S.; Mehaffey, M.; Neale, A. Development of a spatially complete floodplain map of the conterminous United States using random forest. *Sci. Total Environ.* **2019**, *647*, 942–953. [CrossRef] [PubMed]
6. Betts, M.G.; Wolf, C.; Ripple, W.J.; Phalan, B.; Millers, K.A.; Duarte, A.; Butchart, S.H.M.; Levi, T. Global forest loss disproportionately erodes biodiversity in intact landscapes. *Nature* **2017**, *547*, 441–444. [CrossRef]

7. Kavzoglu, T. Object-Oriented Random Forest for High Resolution Land Cover Mapping Using Quickbird-2 Imagery. In *Handbook of Neural Computation*; Elsevier Inc.: Amsterdam, The Netherlands, 2017; pp. 607–619. ISBN 9780128113196.
8. Rodriguez-Galiano, V.F.; Ghimire, B.; Rogan, J.; Chica-Olmo, M.; Rigol-Sanchez, J.P. An assessment of the effectiveness of a random forest classifier for land-cover classification. *ISPRS J. Photogramm. Remote Sens.* **2012**, *67*, 93–104. [CrossRef]
9. Shao, Y.; Lunetta, R.S. Comparison of support vector machine, neural network, and CART algorithms for the land-cover classification using limited training data points. *ISPRS J. Photogramm. Remote Sens.* **2012**, *70*, 78–87. [CrossRef]
10. Mountrakis, G.; Im, J.; Ogole, C. Support vector machines in remote sensing: A review. *ISPRS J. Photogramm. Remote Sens.* **2011**, *66*, 247–259. [CrossRef]
11. Ustuner, M.; Sanli, F.B. Polarimetric target decompositions and light gradient boosting machine for crop classification: A comparative evaluation. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 97. [CrossRef]
12. mljar/mljar-supervised: Automates Machine Learning Pipeline with Feature Engineering and Hyper-Parameters Tuning. Available online: https://github.com/mljar/mljar-supervised (accessed on 17 August 2020).
13. O'Neill, R.V.; Hunsaker, C.T.; Jones, K.B.; Riitters, K.H.; Wickham, J.D.; Schwartz, P.M.; Goodman, I.A.; Jackson, B.L.; Baillargeon, W.S. Monitoring environmental quality at the landscape scale. *Bioscience* **1997**, *47*, 513–520. [CrossRef]
14. Belmaker, J.; Zarnetske, P.; Tuanmu, M.N.; Zonneveld, S.; Record, S.; Strecker, A.; Beaudrot, L. Empirical evidence for the scale dependence of biotic interactions. *Glob. Ecol. Biogeogr.* **2015**, *24*, 750–761. [CrossRef]
15. Hastings, A.; Byers, J.E.; Crooks, J.A.; Cuddington, K.; Jones, C.G.; Lambrinos, J.G.; Talley, T.S.; Wilson, W.G. Ecosystem engineering in space and time. *Ecol. Lett.* **2007**, *10*, 153–164. [CrossRef] [PubMed]
16. Dudek, G. Short-Term Load Forecasting Using Random Forests. In *Advances in Intelligent Systems and Computing*; Springer: Berlin/Heidelberg, Germany, 2015; Volume 323, pp. 821–828.
17. Dimopoulos, T.; Tyralis, H.; Bakas, N.P.; Hadjimitsis, D. Accuracy measurement of Random Forests and Linear Regression for mass appraisal models that estimate the prices of residential apartments in Nicosia, Cyprus. *Adv. Geosci.* **2018**, *45*, 377–382. [CrossRef]
18. Fernández-Delgado, M.; Cernadas, E.; Barro, S.; Amorim, D.; Fernández-Delgado, A. Do we Need Hundreds of Classifiers to Solve Real World Classification Problems? *J. Mach. Learn. Resear.* **2014**, *15*, 3133–3181.
19. Wainberg, M.; Alipanahi, B.; Frey, B.J. Are Random Forests Truly the Best Classifiers? *J. Mach. Learn. Resear.* **2016**, *17*, 1–5.
20. Lin, L.; Wang, F.; Xie, X.; Zhong, S. Random forests-based extreme learning machine ensemble for multi-regime time series prediction. *Expert Syst. Appl.* **2017**, *83*, 164–176. [CrossRef]
21. Melkonyan, A.; Koch, J.; Lohmar, F.; Kamath, V.; Munteanu, V.; Alexander Schmidt, J.; Bleischwitz, R. Integrated urban mobility policies in metropolitan areas: A system dynamics approach for the Rhine-Ruhr metropolitan region in Germany. *Sustain. Cities Soc.* **2020**, *61*, 102358. [CrossRef]
22. Esri, HERE, Garmin, USGS, Intermap, INCREMENT P, NRCan, Esri Japan, METI, Esri China (Hong Kong), Esri Korea, Esri (Thailand), NGCC, (c) OpenStreetMap Contributors, and the GIS User Community. Available online: https://www.aacounty.org/departments/public-works/ourwaater/images/ProposedEligibleAreas_Basemap.pdf (accessed on 16 August 2020).
23. Drusch, M.; Del Bello, U.; Carlier, S.; Colin, O.; Fernandez, V.; Gascon, F.; Hoersch, B.; Isola, C.; Laberinti, P.; Martimort, P.; et al. Sentinel-2: ESA's Optical High-Resolution Mission for GMES Operational Services. *Remote Sens. Environ.* **2012**, *120*, 25–36. [CrossRef]
24. LandViewer. EARTH OBSERVING SYSTEM. Available online: https://eos.com/lv/ (accessed on 16 August 2020).
25. Spatial-Resolutions-Sentinel-2 MSI-User Guidez-Sentinel Online. Available online: https://sentinel.esa.int/web/sentinel/user-guides/sentinel-2-msi/resolutions/spatial (accessed on 16 August 2020).
26. Inglada, J.; Vincent, A.; Arias, M.; Tardy, B.; Morin, D.; Rodes, I. Operational High Resolution Land Cover Map Production at the Country Scale Using Satellite Image Time Series. *Remote Sens.* **2017**, *9*, 95. [CrossRef]
27. Tran, H.; Tran, T.; Kervyn, M. Dynamics of land cover/land use changes in the Mekong Delta, 1973–2011: A Remote sensing analysis of the Tran Van Thoi District, Ca Mau Province, Vietnam. *Remote Sens.* **2015**, *7*, 2899–2925. [CrossRef]

28. Wessels, K.; van den Bergh, F.; Roy, D.; Salmon, B.; Steenkamp, K.; MacAlister, B.; Swanepoel, D.; Jewitt, D. Rapid Land Cover Map Updates Using Change Detection and Robust Random Forest Classifiers. *Remote Sens.* **2016**, *8*, 888. [CrossRef]
29. Zhang, H.K.; Roy, D.P. Using the 500 m MODIS land cover product to derive a consistent continental scale 30 m Landsat land cover classification. *Remote Sens. Environ.* **2017**, *197*, 15–34. [CrossRef]
30. Hermosilla, T.; Wulder, M.A.; White, J.C.; Coops, N.C.; Hobart, G.W. Disturbance-Informed Annual Land Cover Classification Maps of Canada's Forested Ecosystems for a 29-Year Landsat Time Series. *Can. J. Remote Sens.* **2018**, *44*, 67–87. [CrossRef]
31. Urban Atlas 2018—Copernicus Land Monitoring Service. Available online: https://land.copernicus.eu/local/urban-atlas/urban-atlas-2018?tab=metadata (accessed on 16 August 2020).
32. Brink, H.; Richards, J.; Fetherolf, M.; Cronin, B. *Real-World Machine Learning*; Manning: New York, NY, USA, 2017.
33. Hastie, T.; Tibshirani, R.; Friedman, J. *Random Forests*; Springer: New York, NY, USA, 2009; pp. 587–604.
34. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
35. Georganos, S.; Grippa, T.; Vanhuysse, S.; Lennert, M.; Shimoni, M.; Wolff, E. Very High Resolution Object-Based Land Use-Land Cover Urban Classification Using Extreme Gradient Boosting. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 607–611. [CrossRef]
36. Chen, T.; Guestrin, C. XGBoost: A scalable tree boosting system. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; Association for Computing Machinery: New York, NY, USA, 2016; Volume 13–17, pp. 785–794.
37. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.-Y. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In Proceedings of the Annual Conference on Neural Information Processing Systems 2017, Long Beach, CA, USA, 4–9 December 2017; pp. 3146–3154.
38. MLJAR: Platform for Building Machine Learning Models. Available online: https://cloud.mljar.com/app/#/p/PVd39X0qkODn/datasources (accessed on 16 August 2020).
39. Kuhn, M. Building predictive models in R using the caret package. *J. Stat. Softw.* **2008**, *28*, 1–26. [CrossRef]
40. Kim, J.H. Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap. *Comput. Stat. Data Anal.* **2009**, *53*, 3735–3745. [CrossRef]
41. McNemar, Q. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika* **1947**, *12*, 153–157. [CrossRef]
42. Lachin, J.M. Introduction to sample size determination and power analysis for clinical trials. *Control. Clin. Trials* **1981**, *2*, 93–113. [CrossRef]
43. He, H.; Garcia, E.A. Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.* **2009**, *21*, 1263–1284. [CrossRef]
44. Maxwell, A.E.; Warner, T.A.; Fang, F. Implementation of machine-learning classification in remote sensing: An applied review. *Int. J. Remote Sens.* **2018**, *39*, 2784–2817. [CrossRef]
45. Abdi, A. *Decadal Land-use/land-cover and Land Surface Temperature Change in Dubai and Implications on the Urban Heat Island Effect: A Preliminary Assessment*; Center for Open Science: Charlottesville, VA, USA, 2019. [CrossRef]
46. Eklundh, L.; Harrie, L.; Kuusk, A. Investigating relationships between landsat ETM+ sensor data and leaf area index in a boreal conifer forest. *Remote Sens. Environ.* **2001**, *78*, 239–251. [CrossRef]
47. Lukeš, P.; Stenberg, P.; Rautiainen, M.; Mõttus, M.; Vanhatalo, K.M. Optical properties of leaves and needles for boreal tree species in Europe. *Remote Sens. Lett.* **2013**, *4*, 667–676. [CrossRef]