IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 3, June 2022
Impact Factor: 6.252

Comparative Study Between KNN & SVM

Binitta Sunny[1], Leema George[2]
Student, Computer Science, Santhigiri College of Computer Sciences, Thodupuzha, India[1]
Assistant professor, Computer Science, Santhigiri college of computer sciences, Thodupuzha, India[2]

Abstract:*Two popular machine learning methods are Support Vector Machines (SVM) and k-Nearest Neighbour (KNN). KNN Algorithms simply store datasets throughout the training phase, and when new data is received, it is classified into a category that is quite similar to the new data. SVM is a supervised machine technique that may be used for both classification and regression. Though we also state regression problems, categorization is the best fit. The SVM algorithm's goal is to find a hyperplane in an N-dimensional space that distinguishes between data points. Used for categorising images, The KNN and SVM each have strengths and disadvantages. When classifying an image, the SVM creates a hyperplane, dividing the input space into classes and classifying the image based on that hyperplane.*

Keywords:K-Nearest Neighbour, Support Vector Machine, Remote Sensing, Small Unmanned Aircraft System, etc.

I. INTRODUCTION

Support Vector Machines (SVM) and k-Nearest Neighbour (KNN) are two common machine learning algorithms. KNN Algorithms in the training phase just store the datasets and when it gets new data, then it classifies that data into a category that is much similar to the new data. Support Vector Machine (SVM) is a supervised machine algorithm used for both classification and regression. Though we say regression problems as well it's best suited for classification. The objective of the SVM algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points. Used for classifying images, The KNN and SVM each have strengths and weaknesses. When classifying an image, the SVM creates a hyperplane, dividing the input space between classes, classifying based upon which side of the hyperplane an unclassified object lands when placed in the input space.

The KNN however, is used as the system of voting to determine which class an unclassified object belongs to, considering the class of the nearest neighbours in the input space. The SVM is extremely fast, classifying images in roughly ten seconds as opposed to the KNN which takes anywhere from forty to fifty seconds to classify the same image. The SVM is an eager learner, determining the decision boundary from training data before considering any pixels with an unknown class. Conversely, the KNN is a lazy learner. Just storing training pixels and waiting until it is given an unknown pixel before determining the decision criteria for the pixel. When classifying, the KNN will generally classify accurately; however, it generates several small misclassifications that interfere with the final classified image that is outputted. In comparison, the SVM will occasionally misclassify a large object that rarely interferes with the final classified image. While both algorithms yield positive results regarding the accuracy in which they classify the images, the SVM provides significantly better classification accuracy and classification speed than the KNN.

II. K-NEAREST NEIGHBOUR (KNN)

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on the Supervised Learning technique. It assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories. The K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a good suite category by using K- NN algorithm. KNN is a non-parametric algorithm, which means it does not make any assumptions on underlying data.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 3, June 2022
Impact Factor: 6.252

It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset. KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

The SVM is an eager learner, determining the decision boundary from the training data before considering any pixels with an unknown class. Conversely, the k-Nearest Neighbour (KNN) is a lazy learner, just storing training pixels and waiting until it is given an unknown pixel before determining the decision criteria for the pixel.

KNN learns by analogy, comparing an unknown pixel to its most closely neighboring pixels in the decision space. Each pixel is described by n attributes, the proximity between pixels in the decision space being determined by the similarity between their attributes. Attribute similarity is defined in terms of a distance metric such as Euclidean distance. The Euclidean distance between points x1, x2,…xn) and (y1, y2,…yn) is defined as:

Tuning K

There is no particular way to determine the best value for "K", so we need to try some values to find the best out of them. The selection of a value for k is very important when classifying with a KNN. As k is increased, the decision boundary is smoothed, resulting in increased generalization. Typically, k is selected through experimentation, iteratively U.S. Forest Service RMRS P-78. 2020. 101 running the classifier while increasing k, assessing classification accuracy at each iteration. The k value from the iteration resulting in the highest classification accuracy can be selected for use in that decision space.

A very low value for K such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model. large values for K are good, but it may find some difficulties.

A related approach uses an iterative n-fold cross validation using a single set of training pixels for both training and assessing accuracy for each value of k evaluated.

Implementation

Python implementation of KNN Algorithm

Python implementation of the K-NN algorithm, we will use the same problem and dataset which we have used in Logistic Regression. But here we will improve the performance of the model.

Step 1: Set and prepare data

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import make_classification
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn import metrics
```

After loading important libraries, we create ourdata using sklearn. datasets with 200 samples, 8 features, and 2 classes. Then data is split into the train(80%) and test(20%) data and scaled using StandardScaler.

```
X,Y=make_classification(n_samples=
200,n_features=8,n_informative=8,n_redundant=0,n_repeated=0,n_classes=2,random_state=14)
X_train, X_test, y_train, y_test= train_test_split(X, Y, test_size= 0.2,random_state=32)
sc= StandardScaler()
```

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 3, June 2022
Impact Factor: 6.252

```
sc.fit(X_train)
X_train= sc.transform(X_train)
sc.fit(X_test)
X_test= sc.transform(X_test)
X.shape
(200, 8)
```

Step 2: Find the value for K

For choosing the K value, we use error curves and K value with optimal variance, and bias error is chosen as K value for prediction purposes. With the error curve plotted below, we choose K=7 for the prediction

```
error1= []
error2= []
for k in range(1,15):
knn= KNeighborsClassifier(n_neighbors=k)
knn.fit(X_train,y_train)
    y_pred1= knn.predict(X_train)
    error1.append(np.mean(y_train!= y_pred1))
    y_pred2= knn.predict(X_test)
    error2.append(np.mean(y_test!= y_pred2))
# plt.figure(figsize(10,5))
plt.plot(range(1,15),error1,label="train")
plt.plot(range(1,15),error2,label="test")
plt.xlabel('k Value')
plt.ylabel('Error')
plt.legend()
```

Step 3: Predict

In step 2, we have chosen the K value to be 7. Now we substitute that value and get the accuracy score = 0.9 for the test data.

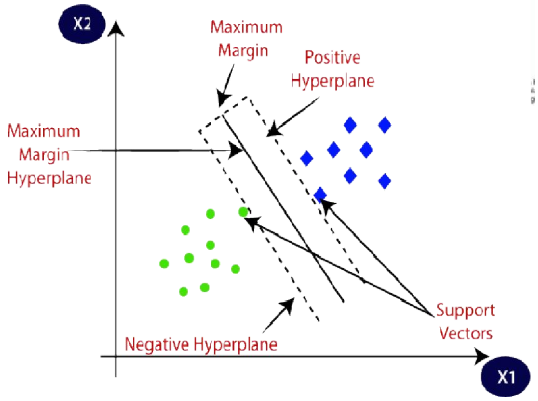
```
knn= KNeighborsClassifier(n_neighbors=7)
knn.fit(X_train,y_train)
y_pred= knn.predict(X_test)
metrics.accuracy_score(y_test,y_pred)
0.9
```

III. SUPPORT VECTOR MACHINE

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes .so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

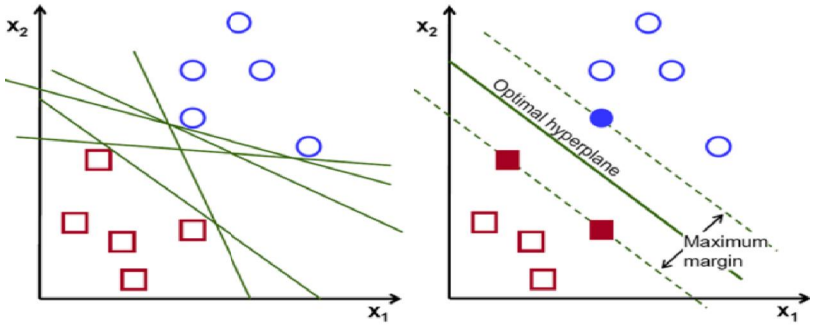
When classifying an image, the SVM creates a hyperplane, dividing the input space between classes, classifying based upon which side of the hyperplane an unclassified object lands when placed in the input space. The algorithm performs a pixel-based classification, labeling each pixel in the image with the postfire effects class determined by the classifier. SVM classifiers have been successfully used for image classification, including for mapping burn severity from medium resolution satellite imagery.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 3, June 2022
Impact Factor: 6.252



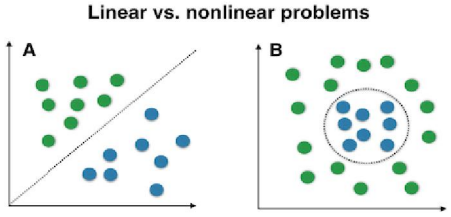
Hyperplane: There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space, but we need to find out the best decision boundary that helps to classify the data points. This best boundary is known as the hyperplane of SVM.The dimensions of the hyperplane depend on the features present in the dataset, which means if there are 2 features (as shown in image), then hyperplane will be a straight line. And if there are 3 features, then hyperplane will be a 2-dimension plane.

Support Vectors: The data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector. Since these vectors support the hyperplane, hence called a Support vector.



Types of SVM

- **Linear SVM**: Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.
- **Non-linear SVM:** Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.



How to Find the Optimal Hyperplane?

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 3, June 2022
Impact Factor: 6.252

The best hyperplane will be whose margin is the maximum. Generally, the margin can be taken as 2* p, where p is the distance b/w separating hyperplane and nearest support vector.How can we find the biggest margin?It is rather simple:

1. You have a dataset
2. Select two hyperplanes which separate the data with no points between them
3. Maximize their distance (the margin)

The region bounded by the two hyperplanes will be the biggest possible margin.If it is so simple why does everybody have so much pain understanding SVM?It is because as always, the simplicity requires some abstraction and mathematical terminology to be well understood.
So, we will now go through this recipe step by step:

Step 1: You have a dataset DD and you want to classify it. Most of the time your data will be composed of nn vectors xixi.Each xixi will also be associated with a value yiyi indicating if the element belongs to the class (+1) or not (-1).Note that yiyi can only have two possible values -1 or +1.
Moreover, most of the time, for instance when you do text classification, your vector xixi ends up having a lot of dimensions. We can say that xixi is a pp-dimensional vector if it has pp dimensions.So your dataset DD is the set of nn couples of element (xi,yi)(xi,yi)The more formal definition of an initial dataset in set theory is:
D={(xi,yi)|xi∈Rp,yi∈{−1,1}}ni=1D={(xi,yi)|xi∈Rp,yi∈{−1,1}}i=1n

Step 2: You need to select two hyperplanes separating the data with no points between them. Finding two hyperplanes separating some data is easy when you have a pencil and a paper. But with some pp-dimensional data it becomes more difficult because you can't draw it.Moreover, even if your data is only 2-dimensional it might not be possible to find a separating hyperplane!You can only do that if your data ***is*** linearly separable.

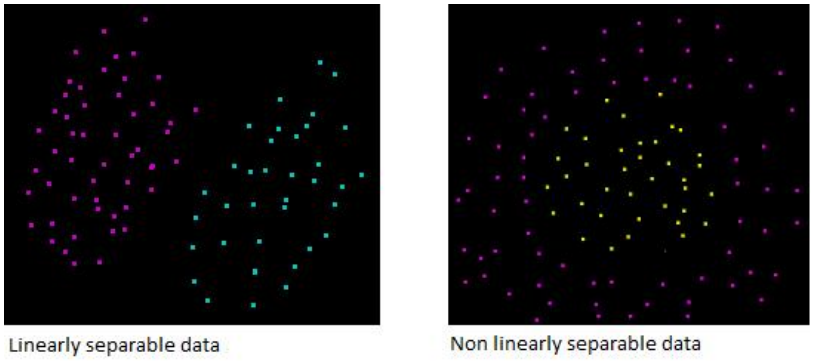


Figure 1: Data on the left can be separated by a hyperplane, while data on the right can'tSo let's assume that our dataset DD IS linearly separable. We now want to find two hyperplanes with no points between them, but we don't have a way to visualize them.

SVM Implementation

For this study, we used the SVM implementation that is available in OpenCV (OpenCV 2017), which used the LibSVM implementation which is very widely used by machine learning practitioners (Chang and Lin 2011). When users selected training pixels, it was common for there to be very small spatial spectral clusters of pixels in the training regions that did not match the class label assigned by the user. To compensate for this noise, the SVM used a soft margin when searching for the optimal separating hyperplane. Consideration was given to what value of Cshould be used while searching for the optimal separating hyperplane using a soft margin. Assessment of C allowed the determination of the most effective weight to be assessed for the error resulting from noisy training pixels.
Additionally, we investigated the degree of linear separability of the training data in order to determine whether the original decision space was adequate for classification or if it was necessary to map the decision space to a

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 3, June 2022
Impact Factor: 6.252

higher dimensionality. In the situations where the original space was not adequately linearly separable, the RBF, Chi Squared and Histogram Intersection Kernels were evaluated.

IV. ALGORITHM COMPARISON HYPOTHESIS

In testing whether an SVM or kNN classifier can map wildland post-fire effects more accurately, a null hypothesis (H0) was specified along with an associated alternate hypothesis (H1). If H0 was rejected, then H1 would be accepted in its place. For this experiment, the independent variable was the algorithm selection between SVM and kNN. The dependent variable was burn severity mapping accuracy. The null and alternate hypotheses were: H0: Support Vector Machines and kNN have equal accuracy when mapping burn severity classes using hyperspatial color imagery. H1: Support Vector Machines have different accuracy than kNN when mapping burn severity classes using hyperspatial color imagery.

H0 was tested by comparing algorithm accuracy, which was calculated from confusion matrices generated by validating an image classification by each of the algorithms against user labeled pixels. A two-tailed Student's t-test established the statistical significance of the accuracy results on the difference in accuracy between the SVM and kNN classifications. A p-value below a significance level of 0.05 rejected H0 in favor of H1 which established that the mean accuracy of the SVM is different from the kNN. While H0 is rejected in favour of H1, the more accurate algorithm is identified by selecting the algorithm that has the highest accuracy results. The statistical analysis was again conducted with a single-tailed Student's t-test establishing the statistical significance of the accuracy results from the classifier with the higher accuracy against the classifier with the lower accuracy.

Algorithm Comparison of Experimental Methodology

To evaluate the accuracy of both the SVM and the as well as KNN for determining burn extent and biomass consumption, a collection of orthomosaics obtained with sUAS flying over fires in the Boise National Forest (BNF) as well as the Bureau of Land Management - Boise the district (BLM-B) was classified hierarchically. SVM and KNN classifiers were used. The initial stage based on burn extent, segmenting the image burned and unburned areas The following stage black was used to identify the burned areas of the image ash (representing partial combustion) versus white ash (This indicates a more complete combustion).

Creation of Training Data

Creation of training post -fire imagery was used to select training pixels. It obtained with a sUAS over burned areas throughout the BNF as well as BLM-B. The user chose training pixels by identifying regions in the image that contained a set of homogeneous pixels, which the user specified the class label.the training Data Selector (TDS), a graphic tool developed as part of this research effort, aided the user in identifying, selecting, and labelling homogeneous regions of the image.

It was advantageous to have approximately the same number of training pixels in each of the classes when desig ning training regions in the image to ensure that the training data contained balanced amounts of both classes, re sulting in more accurate classifications.

This criterion was more applicable to the kNN than the SVM, but because we were using the same training regio ns for both classifiers, the training data had to accommodate both algorithms' implementations. It was also disco vered that keeping the training sets under 1,000 pixels in size helped the implementations of both algorithms avo id overfitting the decision boundary to the data, resulting in greater generalisation. To ensure that the training re gions cover an adequate range.Only a user-specified percentage of the pixels were extracted from the regions when the training pixels were extracted. The regions were assigned to the training pixel set and labelled with the associated user-specified class region of training Typically, when pixel extraction from Only one to two percent of the pixels in a training region were retained in training within the training regions TDS determines. This training subsample is regions provided sufficient representation of the while reducing the variation within the training region the amount of training data.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 3, June 2022
Impact Factor: 6.252

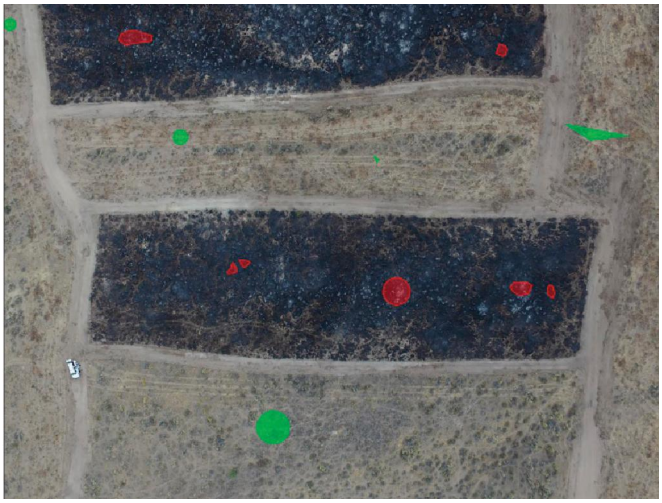


Figure1: Training pixel regions for training burn extent as denoted by the user using the Training Data Selector. Green polygons are unburned pixels, red polygons denote black ash pixels.

V. IMAGE CLASSIFICATION WITH SVM AND KNN

The goal of classifying a post-fire photograph is to identify the size of the burn and the amount of biomass consumed in the burned region. To do this, the image was first categorised into burned and non-burned areas The categorization The spectral analysis of burn extent made it easier. the ability to distinguish between burned and unburned vegetation as previously proved in this research effort (Hamilton and colleagues, 2017). In order to accomplish this, the user within the selected burned and unburned regions Using the TDS, create a training image.The classifiers were trained using pixels from these locations. SVM or KNN were used to classify the image into burned and unburned training pixels. Iterative 5-fold cross-validation (Han et al. 2012) was used to identify the optimal classifier parameter values for both classifiers, as indicated in table 1.

After the image was classified into burned and unburned pixels, image processing morphological functions such as dilation, erosion, and the detection of extremely small connected components were used to apply a size filter to the classed output (Gonzalez and Woods 2008). Dilation enlarges the borders of foreground pixels over time, whereas erosion erodes the boundaries of foreground pixels.These functions were used to filter out clusters of pixels that were smaller than the objects being identified, removing spatial clusters of homogeneous pixels that were sub-object in size.

Unburned vegetation smaller than sagebrush, for example, was absorbed into the surrounding black ash class when it was surrounded by burned vegetation. Small patches of herbaceous plants that were too scarce to carry enough fire or create enough charred vegetation to be detected were the most common cause of these misclassified clusters.Misclassifications were frequently induced by shadows in unburned areas of the image.(It was discovered that completing image capture flights as close to solar noon as feasible, or flying on cloudy days and reducing the shutter speed, minimised misclassifications due to shadows).

In the categorised image, the size filter was also discovered to smooth the boundary between the burned and unburned classes.The image was hierarchically classed after the burn extent classification was cleaned, with the burned region classified by biomass consumption using a binary classification between white ash (high consumption) and black ash (low consumption) using training pixels prepared as mentioned above. The ideal classifier parameter values were determined using an iterative five-fold cross validation method, as indicated in table 1.Training pixel regions classifying biomass consumption as denoted by the user using the Training Data Selector. Blue polygons are white ash, red polygons denote black ash.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 3, June 2022
Impact Factor: 6.252

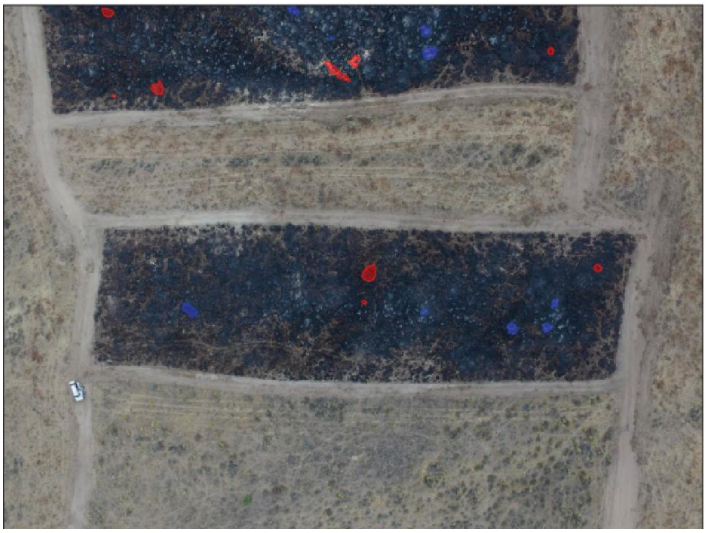


Table 1—Optimal classifier parameter values.

Algorithm	Parameter	Classification type	Optimal value
SVM	Kernel	Burn extent	None (linear)
		Ash type	None (linear)
	C	Burn extent	0.1
		Ash type	0.1
*k*NN	*k*	Burn extent	3
		Ash type	3

Table 1

Validation of Classified Output

Both the SVM and the KNN were tested for accuracy on each of the categorization kinds (burn extent, ash type, and vegetation type). Validation regions were constructed in the same manner as training data for each categorization type. The validation pixel sets were then fed into a tool that created a confusion matrix by comparing the user-specified class to the class predicted by the classifier (Han et al. 2012).

Establishment of Statistical Significance

The student t-test was used to determine statistical significance between the SVM and kNN classification accuracy findings, as reported in Section 2.8. According to H0, the accuracy with which the fire effects land cover classes can be classified using either SVM or kNN is the same. We evaluated the difference in accuracy outcomes for the two algorithms with each of the classification categories because we are utilising a hierarchical classification. This allowed us to analyse accuracy on a finer scale, allowing us to compare the accuracy of both algorithms with each classification category. We discarded H0 (no difference in classification accuracy) in favour of H1 (no difference in classification accuracy) because the two-tailed t-test p-value for a classification type dropped below the previously established significance limit of 0.05.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 3, June 2022
Impact Factor: 6.252

VI. ALGORITHM COMPARISON

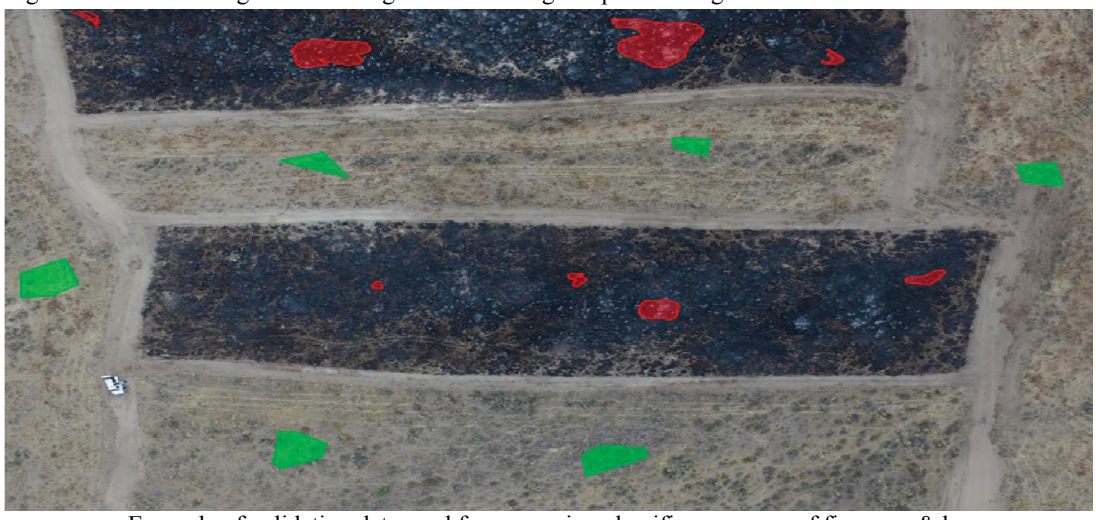
Assessment of the advantages of wildfire mapping Support Vector Machine effects after a fire (SVM) k-Nearest Neighbor (kNN) classifiers was achieved by training an SVM and a kNN with the same data. then classifying on the same training regions orthomosaic after the burn, allowing for comparison comparing both methods' classification results from a standard training set the evaluation of Over a set of burns, both strategies were investigated. 16 fire orthomosaics, the findings of which were RMRS P-78, US Forest Service, 2020. 105 A two-tailed t-test was used.

The p-values that result fell below the threshold of significance, rejecting the hypothesis The null hypothesis was that the mean accuracy was the same for both groups. Both algorithms support the alternative hypothesis. that the algorithms classify the size of the burn and the biomass consumption. Assessment of the advantages of wildfire mapping Support Vector Machine effects after a fire (SVM) To determine statistical significance, a one-tailed, paired t-test was used.accurate results to demonstrate the extent of the burn and Higher biomass consumption can be categorised.

Algorithm Comparison Accuracy Results

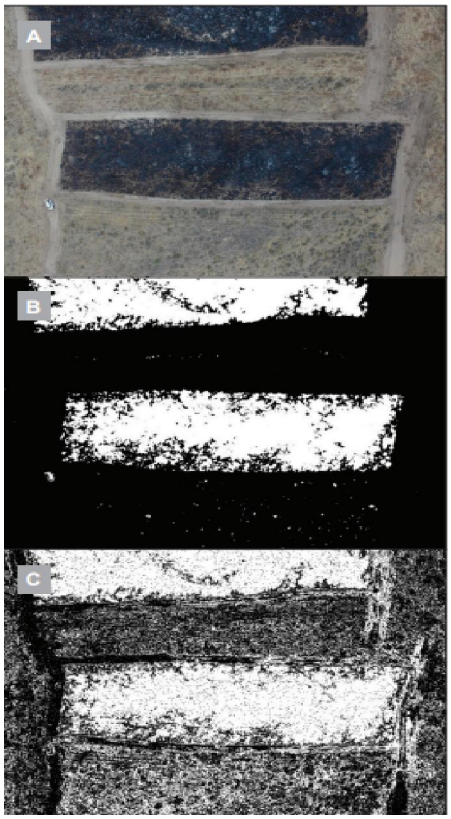
The accuracy of the output classifications of both classifiers was used to measure the competence with which the SVM and KNN were able to classify burn extent and biomass consumption. The number of samples accurately predicted by a classifier divided by the total number of samples equals accuracy (Han et al. 2012). Both the SVM and the KNN were trained on a set of burn extent (burned and unburned) and biomass consumption (black ash and white ash) training regions inside an image to determine accuracy (fig. 6). The classifiers initially classified the burn extent of the pixels in the image, identifying them as burned or unburned. The burned image regions were then categorised by biomass consumption, with burned pixels labelled as white ash or black ash, as illustrated in figure.Each image's validation data sets were created by selecting user-labeled regions of pixels inside the image, then running the pixels from each validation dataset through the SVM and subsequently the KNN. For each algorithm, accuracy was determined by calculating the proportion of validation pixels from a classifier that were classified the same as those identified by the user.

The user labelling of validation data was based on the user's visual observation of the image, complemented with ground observations gathered during the sUAS image acquisition flights.



Example of validation data used for measuring classifier accuracy of figures a & b

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 3, June 2022
Impact Factor: 6.252



a. Unclassified image of Reynolds Creek Prescribed Burn;b. SVM Classification Result;c. NN classification result.

The number of pixels properly predicted by the SVM was divided by the total number of pixels in the validation data set multiplied by 100 to assess accuracy. To achieve a more comprehensive assessment of the accuracy of a set of classifier inputs, accuracy was assessed first on burn extent (ash vs unburned pixels), then on biomass consumption accuracy (black ash vs white ash). Table 2 shows the accuracy results for both classifiers for each fire for burn extent and biomass consumption classifications for each of the validation sets.

The SVM and KNN classification accuracy for both burn extent and biomass consumption were averaged, then multiplied by 100. Table 1 shows the resultant Mean Classification Accuracy.

Table 2—SVM vs. *k*NN burn extent and biomass consumption classification accuracy.

	Burn extent		Biomass Consumption	
Fire	**SVM**	***k*NN**	**SVM**	***k*NN**
Jack	99.8	96.7	99.1	96.9
Northside	95.7	86.1	98	92.5
Reynolds Creek	99.59	79.51	98.7	96.85
Kane Fire	98.96	91.52	96.66	48.83
Deer Flat	99.99	72.62	99.86	78.06
Hoodoo 1	99.29	71.68	95.22	94.07
Hoodoo 2	89.23	50.36	99.25	92.88
Lucky Peak	97.61	74.62	99.85	96.17
mm107 (clip)	97.72	88.76	99.03	89.8
Camp	99.48	94.98	99.74	97.53
Oyhee plot	88.75	79.9	90.57	89.43
Immigrant (clip)	99.48	90.19	95.66	83.96
Elephant	95.08	91.44	99.15	96.68

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 3, June 2022
Impact Factor: 6.252

Table 3—SVM vs. kNN mean classification accuracy.

Algorithm	Burn extent	Biomass consumption
SVM	96.97	97.75
kNN	82.18	88.74

The SVM had superior accuracy for both the burn extent and biomass consumption classifications when the Mean Classification Accuracy was compared to the KNN.

VII. ALGORITHM COMPARISON ACCURACY STATISTICAL SIGNIFICANCE

Using two-tailed paired t-tests, the statistical significance of increasing accuracy between the SVM and the KNN throughout the validation sets for the burn photos was determined. The null hypothesis is that the SVM and KNN both accurately categorize burn extent and biomass consumption. The alternative hypothesis is that the SVM and KNN are not equally accurate in classification. The accuracy results for both the SVM and the KNN were t-tested, with the burn extent being tested first, followed by biomass consumption. the t-test passed the significance level of 0.05, indicating that the null hypothesis is 95 percent certain to be rejected in favor of the alternate hypothesis with a p-value of 0.0005, the burn extent accuracy tests rejected the null hypothesis.
Similarly, the biomass consumption accuracy tests with a p-value of 0.031 rejected the null hypothesis. The hypothesis is rejected in both cases, confirming the alternate hypothesis that the SVM and kNN do not categorise burn extent or biomass consumption with same accuracy. The SVM had a higher Mean Classification Accuracy for both burn extent and biomass consumption, as shown in table 3. Table 2 further demonstrates that the SVM was more accurate for each of the fires. The accuracy values of both the SVM and the kNN were ran via a one-tailed t-test to demonstrate that the SVM has a measurable increase in accuracy over the kNN for both burn extent and biomass use to establish that the SVM has a measurable increase in accuracy over the kNN.

VIII. CONCLUSION

The Support Vector Machine (SVM) and k-Nearest Neighbour (kNN) methods were compared and found to accurately estimate burn extent and biomass consumption from hyperspatial sUAS data. On the fires examined, the SVM beat the KNN in terms of identifying burn extent and biomass consumption, mapping burn extent with an average accuracy of 96.81 percent and biomass consumption with an average accuracy of 97.74 percent. Although the KNN did not map fire effects as accurately as the SVM, it still had an average accuracy of 81.37 percent for burn extent and 88.74 percent for biomass consumption. The addition of Second Order Entropy to the three colour bands as a classifier input was found to boost burn extent.

FUTURE WORK

Improve the efficiency and accuracy of the SVM algorithm. Identify prostate cancer using a research method. Identify and map archaeological items of interest using the research method. More data is needed for testing and validating outcomes. For white ash, more data is needed to determine statistical significance.

ACKNOWLEDGMENT

I have taken efforts in this paper. However, it would not have been possible without the kind support and help of many individuals. I would like to extend my sincere thanks to all of them. I am highly indebted to Ms.Leema George for their guidance and constant supervision as well as for providing necessary information regarding the paper & also for their support in completing the seminar. I would like to express my gratitude towards my parents & friends for their kind cooperation and encouragement which help me in completion of this paper.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 3, June 2022
Impact Factor: 6.252

REFERENCES

[1] Calkins, A.T. 2017. Unmanned aircraft systems (UAS) and photogram metrics as a tool for archaeological investigation in 19th Century historic archaeology. Thesis. Reno, NV: University of Nevada. 122 p
[2] Eidenshink, J.C.; Schwind, B.; Brewer, K.; Zhu, Z.-L.; Quayle, B.; Howard, S.M. 2007. A project for monitoring trends in burn severity. Fire Ecology.
[3] Aplet, G.H.; B. Wilmer, B. 2010. Potential for restoring fire-adapted ecosystems: Exploring opportunities to expand the use of wildfire as a natural change agent. Fire Management Today. 70(1):
[4] Zhou, G.; Li, C.; Cheng, P. 2005. Unmanned aerial vehicle (UAV) real-time video registration for forest fire monitoring. Geosci. Remote Sens. Symp. 2005 IGARSS05 Proc. 2005 IEEE Int., vol. 3, 2005

BIOGRAPHY

Binitta Sunny is studying Master of Computer Applications in Santhigiri College of Computer Sciences, Vazhithala, Idukki, Kerala. She has completed her Bachelor of Computer Applications from Mahatma Gandhi University, Kerala. She has published a paper in IJSR.

Leema George received the M. Tech degree. She is currently working as an assistant professor in Santhigiri College of Computer Sciences, Vazhithala.