ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/336373109

Land Cover Classification from Multispectral Remote Sensing Image Using K-Nearest Neighbor, Deep Neural Network, Decision Tree and SVM Algorithms: A machine learning-based comparis...

Conference Paper · October 2019

CITATIONS
0

READS
243

3 authors:

Torit Chakraborty
Khulna University
8 PUBLICATIONS **3** CITATIONS

SEE PROFILE

Md. Shaharier Alam
Asian Disaster Preparedness Center
12 PUBLICATIONS **28** CITATIONS

SEE PROFILE

Md Didarul Islam
Central Michigan University
5 PUBLICATIONS **3** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project Revision of Functional Master Plan and Detailed Area Plan to make Rajshahi Metropolitan Development Plan (RMDP) Disaster Risk Sensitive View project

Project Health GIS: Incorporating Geospatial technologies in the Public Health Realm View project

All content following this page was uploaded by Torit Chakraborty on 10 October 2019.

The user has requested enhancement of the downloaded file.

Research Paper

Land Cover Classification from Multispectral Remote Sensing Image Using Deep Neural Network, K-Nearest Neighbor, Decision Tree and SVM Algorithms: A machine learning based comparison approach

1 Torit CHAKRABORTY, Graduated, Urban and Rural Planning Discipline, Khulna University; Bangladesh

2 Md. Shaharier ALAM, Graduated, Urban and Rural Planning Discipline, Khulna University; Bangladesh

3 Md. Didarul ISLAM, Graduated, Urban and Rural Planning Discipline, Khulna University; Bangladesh

Abstract

Land cover classification has great contribution in many stages of city planning, urban management and understanding human-environment interaction. It is widely used to analyse spatial pattern of land cover change of an area and further used in simulating land cover change and its impact on environment. The aim of this paper is to utilize machine learning algorithms such as support vector machine (SVM), k- nearest neighbor, deep neural network, decision tree in terms of land cover classification from multispectral satellite data. For this study, Landsat 8 OLI/TIRS sensor data was used for land cover classification. To evaluate the performance of these algorithms in land cover classification, Cohen kappa and overall accuracy score are used using 155 random known points from google earth. The result shows that DNN scores 88% cohen kappa and 92% overall accuracy whereas SVM gains 84% cohen kappa ,89% overall accuracy; decision tree gains 80% cohen kappa, 86% overall accuracy score and k-nearest neighbor gains 84% cohen kappa, 89% overall accuracy. This result clearly indicates that DNN has better ability in classification of low-resolution satellite data than other machine learning algorithms which has the scope to use in different stages of urban planning and management.

Keywords

Deep Neural Network, Support Vector Machine, Random forest, K-nearest neighbor, Land cover Classification

1. Introduction

Land use/cover classification of remotely sensed data has a significant influence on planning and decision making. Urban land use/cover mapping helps planner and decision-maker to investigate urban expansion and its consequences on the environment which aid them to make sustainable development strategy in land management and control. Currently, a lot of algorithms is practiced in land use/cover classification. But accuracy is the main concern in such classification as input data used in most classification is coarse 30 m resolution Landsat

TM or OLI/TIRS data. So, only a few of these algorithms can address better accuracy in terms of this coarse resolution data. There are some popular algorithms for land cover classification such as support vector machine (SVM) (Thai, L., et al., 2012; Ustuner, M., et al., 2015), decision tree (Punia, M., et al., 2011; Torma, M., 2013), k-nearest neighbor (Samaniego, L., & Schulz, K., 2009) artificial neural network (ANN) (Yuan, H., et al., 2009) etc. SVM actually a non-parametric classifier which was first proposed by Vapnik and Chervonenkis (Vapnik, N., and Chervonenkis, A., 1971). SVM has been successfully used for remote sensing data classification. An SVM aims to fit an optimal separating hyperplane or set of hyperplanes in a high or infinite-dimensional space, to locate the optimal boundaries between classes. Linearly separable classes can be used to train SVM. The main limitation of the SVM approach in the application of classification lies in the choice of the kernel and another is speed and size, both in training and testing (Burges, C., J., C., 1998). K-nearest neighbor is also a non-parametric method mostly used in classification and regression (Altman, N., S., 1992). In classification, K closest training samples used as input in feature class where output is also class that classified by a majority vote of its neighbors., the most common class among its k nearest neighbor attached as assigned object.The background principle is to find the closest training(predefined) sample based on distance and predict the level of that. The main limitation of this algorithm is to determine the value of parameter K (number of nearest neighbors). Besides, which type of distance to use and which attribute to use to produce the best results are not clear in this algorithm. Decision tree also a non-parametric supervised learning method used for classification and regression which performance is massively investigated in machine learning and data mining (Quinlan, 1986). In decision analysis, a decision tree can be used to visually and explicitly represent decision and decision making. Actually, decision tree is a schematic, tree-shaped diagram used to determine a course of action or show a statistical probability (Staff, 2017). The tree is structured to show how and why one choice may lead to the next, with the use of the branches indicating each option is mutually exclusive (Staff, 2017). The limitation of the decision tree is extremely overfitting in training data and a slight change in data can result in a drastically different tree.

In the 1940s, Warren McCulloch and Walter Pitts showed that networks of artificial neurons could, in principle, compute any arithmetic or logical function (Hagan et al. 2014). After that, the practical application of artificial neural networks along with the concept of perception network and associated learning came. (Hagan et al. 2014). It is also known as single-layer perceptron network. This network could solve only linearly separable problems. In 1980s, the discovery of the backpropagation algorithm for training multilayer perceptron networks overcome these limitations. Artificial neural network (ANN) is mainly a parallel operating system of many functions which is determined or controlled by its network structure, connection strength, and nodes(Cheng, 2003). These networks are intended to mimic the way humans solve problems through a series of repeated observations between neurons and synapses within the brain (Thakkudan, 2008). Each time data is feedforward and then backpropagated iteratively through the network (known as a cycle) error are reduced (Pijanowski et al, 2005). Application of Neural Network is simple form mathematical point of view. Because the input data sets are randomly weighted for input nods and readjust the recalculated weight according to error and thus minimize error automatically.

Feed-forward network:

$$a=f\left(\sum_{1}^{n} W_n X_n + b_n\right) \quad \text{.....(1)}$$

Activation Function:

$$f(x)=\frac{1}{1+e^{-x}} \quad \text{.....(2)}$$

Adjust weight through backpropagation algorithms:

$$W_{(n+1)} = W_{(n)} + \eta[d_{(n)} - y_{(n)}]x_{(n)} \quad \text{.....(3)}$$

The most popular neural network is multilayer perceptron neural network consisting input layer, hidden layer, and output layer. The input (X) variables are feed-forwarded through the node where it is multiplied by the random weight (W) and then added to bias (b). Then all are summed in summing node that is known as net input and it is forward through activation function which turns the net input into binary value 0 and 1 or multiple categories. The output (y_n) is subtracted from desired or target output (d_n) which is known as error correction. According to error, the weight is readjusted through backpropagation algorithms. This process is repeated until error minimized.

Actually, there are two types of multilayer perceptron neural network. These are a) Non-deep feedforward neural network and b) Deep neural network. Currently, deep neural network (DNN) is a popular term in image classification, facial recognition, and others. Simple non-deep neural network uses a single hidden layer but a deep neural network uses multiple hidden layers. The deep neural network also uses the same feedforward backpropagation algorithm like non-deep feedforward neural network. The advantage of using deep neural network over non-deep feedforward neural network is that it has the ability to address complex non-linearity. The main limitation of the deep neural network is overfitting and underfitting on training data. Srivastava, et al., (2014) proposed a method dropout of nodes in the hidden layer which shows significant improvement of the performance of DNN in training and validation.

2. Materials and Methods

2.1. Study area

The study area selected for this research is Khulna city and its surrounding area, located in Khulna district of Bangladesh which is the third-largest city of Bangladesh and located in the southwestern part of the country. The city is important for the country's economy due to its strategic location. Because it is located close to the second largest seaport of Bangladesh namely Mongla seaport and gateway of the world's largest tract of mangrove forest, the Sundarbans. Besides, it is one of the key hubs of shrimp processing and export of Bangladesh and has a strong industrial and commercial base.

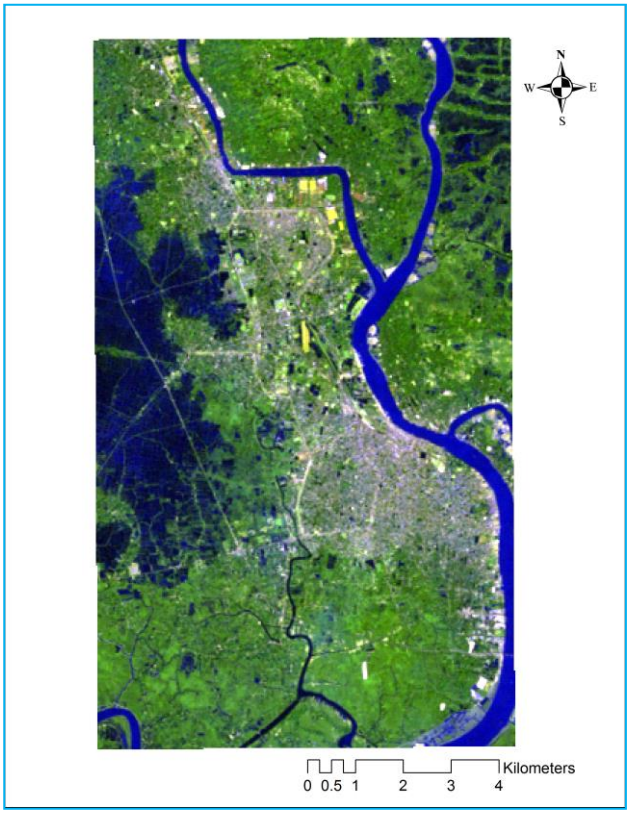

Figure 1: Study area

2.2. Data acquisition

In this research, Landsat TM sensor data was used for land cover classification. The study area lies in a single path and row of Landsat image (path 137, row 44). The cloud-free Landsat TM data of 2011 was downloaded from USGS earth explorer. The data was acquired for the dry season in between November to January to avoid unwanted seasonal variation which may affect the classification.

2.3. Image Processing

After downloading from the USGS Earth Explorer, the raw file was processed to further use. To get the multispectral image of 2011, all the layers were combined by using layer stacking function in ERDAS Imagine software. Then the area of interest was collected from multispectral image data and subset. Data of all seven bands were used in training samples as input in DNN, SVM, KNN, Decision tree. A total of 5652 pixels were collected for four class namely 1) Agriculture & vacant 2) Built-up 3) Wetland and 4) River & Waterbody. So, for 7 bands, total 5652×7 pixels matrix generated which was input matrix for training and total 139769 pixels (139769×7 matrix for 7 bands) would be used for classification.

2.4. Data Preparation for Classifiers

In this research, the land cover classification was conducted using multiple software in different stages. According to the machine learning approach, for image classification, the classifiers need training data to classify the whole data based on the extensive learning from the training datasets. So, training samples were extracted from the image using ArcGIS 10.5 software. Image dataset is 8 bits, so all input pixel's value lies between range 0-255. These pixels values are normalized to 0-1 using equation-

$$X= (x - x.min) / (x.max – x.min) \quad \ldots\ldots (4)$$

The target pixels values were 4 class, so it was converted to binary category class using One Hot Encoding. This binary category class was used in the network as a target variable. The network architecture was used in training was 1 input layer, 2 hidden layers with 60 nodes and 120 nodes and 2 dropout layers where 20% nodes were dropped and 1 output layer with 4 nodes. Rectified linear unit (RELU) was used as the activation function in the hidden layer which has an advantage in vanishing gradient problem in neural network and SoftMax activation function used in output layer for the multiclass problem in output. All the machine learning algorithms (DNN, SVM, Decision-Tree, KNN) were performed to classify using Anaconda Python environment.

2.5. Validation Techniques

To validate model performance three validation techniques Cohen kappa (Shao et al.2012), overall accuracy (Roy et al. 2015) and precision score were used. Cohen kappa measures the inter-rater agreement of category items. The equation of Cohen kappa is-

$$K=(P_o-P_e)/(1-P_e) \quad \ldots\ldots (5)$$

Where, P_o = relative observed agreement among raters and P_e = the hypothetical probability of chance agreement. The kappa values 1 represent the complete agreement between two raters Overall accuracy is the sum of correctly predicted samples divided by a total number of test samples. Equation of overall accuracy is -

$$\text{Overall accuracy}= o / n \quad \ldots\ldots (6)$$

Where, o = number of correctly predicted samples observed and n = total number of test samples.

The precision score is the ratio of $t_p/(t_p + f_p)$ …….(7)

where t_P is the number of true positives and f_p is the number of false positive. Precision score 1 indicates completely precise prediction and 0 indicate there is no precision.

3. Analysis and Results

Input data was divided into 300 small batches and run through the network with 50 epochs. Each epoch means one forward pass and one backward pass of all batches through the network. After completing the training stage, model accuracy goes up to 98.37% and model error goes down to 6.4%.

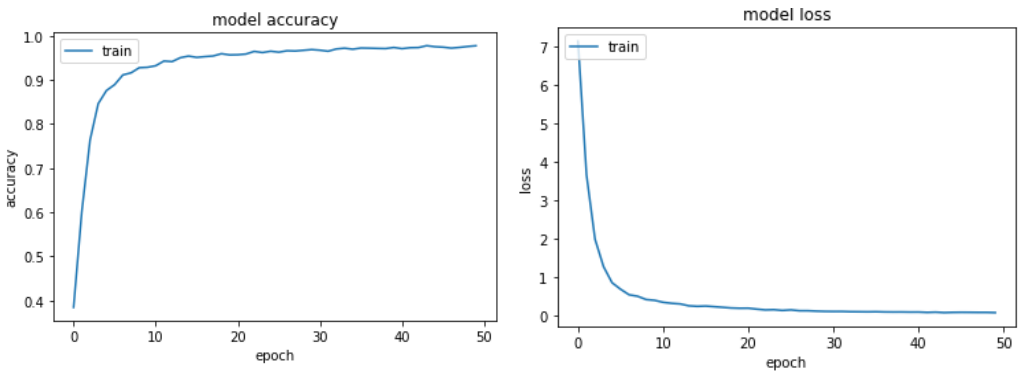

Figure 2. a) model accuracy b) model error (loss) in total 50 epochs

Then the original dataset 139769×7 pixels matrix run through this trained network and the model output was the classified image which was reshaped to the shape of the actual area of interest. Later this array was converted to tiff file format and exported. To compare this model performance with support vector machine (SVM), decision tree and k-nearest neighbor same sample data run though these algorithms for training and same original dataset run through these trained models and similarly exported the classified output to tiff file format. To validate these model's accuracy, 155 random points generated on the classified images which contain classified information. These points were then observed with the actual ground scenario of google earth historical imagery 2011. Then observed truth value of these points were collected and stored in a shapefile format. Later these 155 points observed truth values and class values were analyses using cohen kappa statistics, overall accuracy, and precision score. The below tables show error matrix, cohen kappa score, overall accuracy and the precision score of classified image for DNN, SVM, decision tree and k-nearest neighbor algorithm.

Table 1. Error matrix for DNN:

Error matrix		Observed				
Classified	**DNN**	Wetland	River	Built up	Agriculture & Vacant	Total
	Wetland	30	0	1	2	33
	River	0	47	9	0	56
	Built up	1	0	57	0	58
	Agriculture & Vacant	0	0	0	8	8
Total		31	47	67	10	155

Table 2. Error matrix for SVM:

Error matrix		Observed				
Classified	**SVM**	Wetland	River	Built up	Agriculture & Vacant	Total
	Wetland	30	2	1	3	36
	River	0	42	7	0	49
	Built up	1	3	59	0	63
	Agriculture & Vacant	0	0	0	7	7
Total		31	47	67	10	155

Table 3. Error matrix for Decision Tree:

Error matrix		Observed				
Classified	**Decision Tree**	Wetland	River	Built up	Agriculture & Vacant	Total
	Wetland	30	3	4	2	39
	River	0	35	2	0	37
	Built up	1	9	61	0	71
	Agriculture & Vacant	0	0	0	8	8
Total		31	47	67	10	155

Table 4. Error matrix for K-nearest neighbor:

Error matrix		Observed				
Classified	**K-nearest neighbor**	Wetland	River	Built up	Agriculture & Vacant	Total
	Wetland	27	0	1	3	31
	River	1	44	6	0	51
	Built up	3	3	60	0	66
	Agriculture & Vacant	0	0	0	7	7
Total		31	47	67	10	155

Table 5. Validation results of classification algorithms

Validation Method	*DNN*	*SVM*	*K-nearest neighbor*	*Decision tree*
Cohen Kappa (%)	88	84	84	80
Overall Accuracy (%)	92	89	89	86
Precision Score (%)	93	90	90	88

It seems that DNN has strong ability in pixel-based classification than SVM, k-nearest neighbor and decision tree whereas SVM and k-nearest neighbor equally perform in classification but decision tree perform less than other algorithms. DNN scores 88% cohen kappa and 92% overall accuracy and SVM and k-nearest neighbor equally scores 84% cohen kappa and 89% overall accuracy. All algorithms perform very well according to precision score but kappa and the overall score are the most used validation method. Based on visual analysis also, it seems that DNN has excellent ability in single pixel-wise classification whereas SVM and k-nearest neighbor seems more compact in classification.

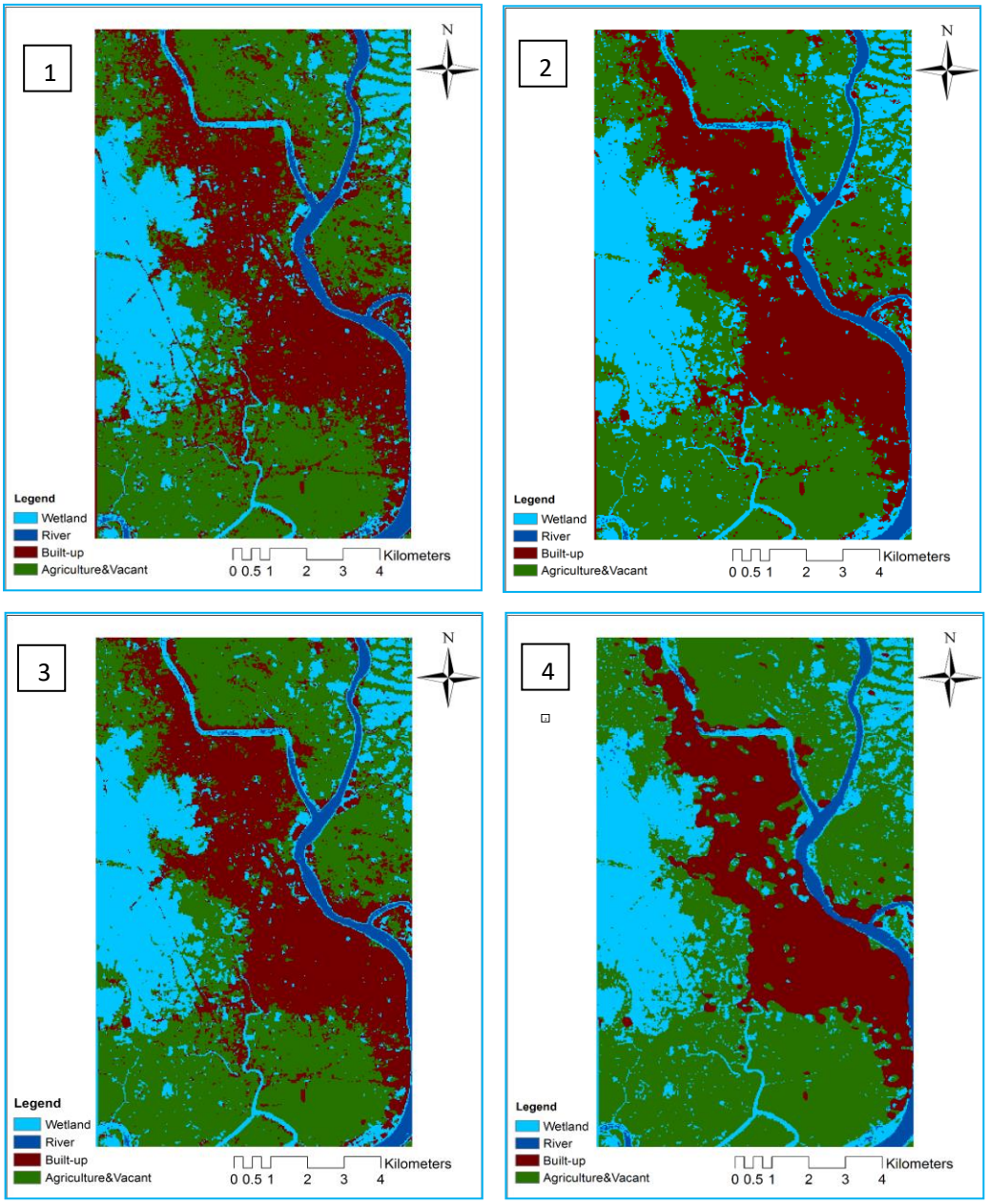

Figure 2. Classified image using 1. DNN 2. SVM 3. KNN 4. DT

4. Conclusion

In this paper, DNN applied in low resolution 30 m satellite image classification and the result is compared with other image classification algorithms such as SVM, k-nearest neighbor, and decision tree. Based on a statistical analysis of results, it can be concluded that DNN can be a better alternative method for image classification especially for low-resolution satellite image classification where it performs better than other methods. DNN has some limitation including overfitting on training data and parameterization. To combat overfitting 20% nodes in hidden layers dropped which make the network not to read 20% nodes weights in weight readjustment in the time of forward pass and backward pass. This way network will fight for overfitting. For the parameterization, trial and error method was used to set parameters. If these parameters properly set and other minor limitation can overcome, DNN can be a powerful algorithm for machine learning and land use/cover classification. So, this algorithm can be used to classify land cover from satellite image by which it can be a tremendous support for working further on land cover change, land-use modeling, land cover prediction, urban planning and management, etc.

5. References

Thai, L., Hai, T. and Thuy, N. (2012). Image Classification using Support Vector Machine and Artificial Neural Network. *International Journal of Information Technology and Computer Science*, 4(5), pp.32-38.

Ustuner, M., Sanli, F.B. and Dixon, B., 2015. Application of support vector machines for landuse classification using high-resolution RapidEye images: a sensitivity analysis. *European Journal of Remote Sensing*, *48*(1), pp.403-422.

Punia, M., Joshi, P.K. and Porwal, M.C., 2011. Decision tree classification of land use land cover for Delhi, India using IRS-P6 AWiFS data. *Expert systems with Applications*, *38*(5), pp.5577-5583.

Törmä, M., 2013. Land cover classification of FINNISH LAPLAND using decision tree classification algorithm. *The photogrammetric Journal of Finland*, *23*(2).

Demuth, H.B., Beale, M.H., De Jess, O. and Hagan, M.T., 2014. *Neural network design*. Martin Hagan.

Thekkudan, T.F., 2008. *Calibration of an Artificial Neural Network for Predicting Development in Montgomery County, Virginia: 1992-2001* (Doctoral dissertation, Virginia Tech).

Cheng, J., 2003. *Modelling spatial and temporal urban growth*. Utrecht University.

Pijanowski, B.C., Pithadia, S., Shellito, B.A. and Alexandridis, K., 2005. Calibrating a neural network-based urban change model for two metropolitan areas of the Upper Midwest of the United States. *International Journal of Geographical Information Science*, *19*(2), pp.197-215.

Yuan, H., Van Der Wiele, C. and Khorram, S., 2009. An automated artificial neural network system for land use/land cover classification from Landsat TM imagery. *Remote Sensing*, *1*(3), pp.243-265.

Vapnik, V.N. and Chervonenkis, A.Y., 2015. On the uniform convergence of relative frequencies of events to their probabilities. In *Measures of complexity* (pp. 11-30). Springer, Cham.

Altman, N.S., 1992. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, *46*(3), pp.175-185.

Quinlan, J.R., 1986. Induction of decision trees. *Machine learning*, *1*(1), pp.81-106.

Staff, I.,2017. *Decision Tree*. *Investopedia*. Retrieved 7 November 2017, from http://www.investopedia.com/terms/d/decision-tree.asp

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, *15*(1), pp.1929-1958.

Samaniego, L. and Schulz, K., 2009. Supervised classification of agricultural land cover using a modified k-NN technique (MNN) and Landsat remote sensing imagery. *Remote Sensing*, *1*(4), pp.875-895.

Burges, C.J., 1998. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, *2*(2), pp.121-167.

Shao, Y. and Lunetta, R.S., 2012. Comparison of support vector machine, neural network, and CART algorithms for the land-cover classification using limited training data points. *ISPRS Journal of Photogrammetry and Remote Sensing*, *70*, pp.78-87.

Roy, S., Farzana, K., Papia, M. and Hasan, M., 2015. Monitoring and prediction of land use/land cover change using the integration of Markov chain model and cellular automation in the Southeastern Tertiary Hilly Area of Bangladesh. *Int. J. Sci. Basic Appl. Res*, *24*, pp.125-148.

View publication stats