RESEARCH ARTICLE
ISSN NO: 2410-8383

A Data-driven Model for Sustainable Deployment of Climate Smart Agriculture Practices Among Smallholder Farmers in Kakamega

Simon NDUNG'U*[1], Moses THIGA[2], Philip WANDAHWA[1] and Vitalis OGEMA[1]

1. Department of Agriculture and Land Use Management in the School of Agriculture, Veterinary Services and Technology, Masinde Muliro University of Science and Technology – Africa
2. Department of Computer Science and Information Technology, Kabarak University – Africa

Corresponding author: ndogo.ndungu@gmail.com

Submitted 5th October, 2023, Accepted 5th October, 2023 and Published 12th November 2023

ABSTRACT

Kenya's agriculture is dominated by millions of smallholder farmers who produce over 75 per cent of the national agricultural production. The smallholder farmers, however, are the most vulnerable to climate change because of various socioeconomics, demography, and policy trends limiting their capacity to adapt to change. To mitigate against the negative effects of climate change on smallholder farmers' numerous interventions, in the form of Climate Smart Agriculture Technologies have been developed and promoted by development partners and government departments. Not all the targeted smallholder farmers, however, participate in and adopt the technologies at the ideal rates and intensity leading to their dis-adoption and abandonment. This study, therefore, sought to develop a data-driven model for the sustainable deployment and adoption of CSA practices among smallholder farmers in Kakamega county. The study employed a mixed methods research design. Through a quantitative survey of 428 smallholder Climate Smart Agriculture Technology adopters and dis-adopters this study reviewed and investigated the major socio-economic and biophysical characteristics associated with the different smallholder farmer categories. Supervised Machine Learning using the Scikit-Learn library of Python Programming language was used to build, pilot, and review Decision Tree and Random Forest Classifier models for the sustainable deployment and adaptation of CSA practices among Kakamega county's smallholder farmers. 19 key variables were identified for the development of a predictive model for CSA Technology adoption. A predictive tool was developed and piloted among 15 smallholder CSA farmers. The classifier model produced a Mean Squared Error of 0.16. The proposed model predicted smallholder farmer adoption at an accuracy of 89.53 per cent and 90.0 per cent with test data and pilot data, respectively. This study, therefore, proposes a new model for the optimal selection of Climate Smart Agriculture intervention beneficiaries.

Keywords: Data-Driven Model, Climate Smart Agriculture, CSA Adoption, Sustainable Deployment

I. INTRODUCTION

Smallholder agriculture is a term used to describe rural producers predominantly in developing countries who mainly farm using family labour and for whom the farm provides the principal source of income (Cornish, 1998). Smallholder farmers are those who work on and own land ranging from 0.5 to 5 hectares, according to Kenya's Ministry of Agriculture. Kenya is estimated to be dominated by 4.5 million smallholder farmers who produce more than 75 per cent of the country's agricultural output (Kirimi et al., 2011). The contribution of smallholder farmers to agricultural development cannot be underestimated as they play a significant role in the food security of both the country and the continent of Africa. Available reports indicate that smallholder farmers produce over 80 per cent of the food produced in Africa (Hlophe-Ginindza & Mpandeli, 2021) . In addition, the smallholder farmers produce for their households thereby reducing the burden on the government to provide food for them.

Kenyan smallholder farmers face several challenges. First, because of their small landholdings, they produce only enough food to feed their families and have little to sell. As a result, their ability to generate income is reduced, and their poverty levels rise. Second, smallholder farmers cannot obtain agricultural credit to improve their farming practices because they lack adequate data to support their creditworthiness (Maru et al., 2018). Third, because the majority of these smallholder farmers live in remote and rural areas, they do not have access to the necessary infrastructure and other services that would enable them to access farm inputs and agricultural markets Aaron (Aaron, 2012). Fourth, smallholder farmers face pest and disease outbreaks, droughts, and a scarcity of arable land to both carry out their farming practices and live in (Hlophe-Ginindza & Mpandeli, 2021). Lastly, smallholder farmers are faced with the major challenge of climate change.

The Kenya Climate Change Act of 2016 defines climate change as the "change in climate systems which is caused by significant changes in the concentration of greenhouse gases as a consequence of human activity and which in addition to natural climate change that has been observed during a considerable period" ("Climate Change Act," 2016). This implies that human activity is primarily to blame for climate change. Thus, climate change is concerned with long-term changes in weather patterns around the world caused by the concentration of GHGs primarily from human activities. A report by Kenya Agricultural Research Institute [KARI] (2009) indicates that the zones that are considered semi-arid may become arid areas or too dry for any agricultural activity to take place. Climate change is, therefore, expected to result in losses in the production of basic staples like maize and beans, and livestock products which in effect may lower food accessibility and lower per capita calorie availability.

Climate change studies have identified rising temperatures, more variable rainfall, and changes in the onset and offset of rainfall as some of the major challenges facing agriculture today (Harvey & Pilgrim, 2011). In addition, high temperatures and drought conditions have been reported to harm maize and bean production, flowering, and yields in many tropical countries (Eitzinger et al., 2013). Furthermore, climate change has been reported to harm tropical agricultural production including high pest and disease incidences. ClimateChange.ie (2017), associates the invasion of fall armyworms and other pests in Africa with climate change.

The foregoing notwithstanding, climate change has impacted negatively on smallholder agriculture through unpredictable weather and intensified drought cycles making farming unpredictable and reducing agricultural productivity (ClimateChange.ie, 2017). As a result, smallholder farmers must develop coping strategies such as sustainable agriculture, climate-smart agriculture (CSA), precision agriculture, and other interventions.

To counter these challenges, Climate Smart Agriculture (CSA) interventions have been developed to increase smallholder farmers' resilience to climate change, reduce Greenhouse Gas (GHG) emissions, and increase agricultural productivity (FAO, 2020). CSA has been termed as the method of combining various sustainable methods to address a specific community's climate challenges (Rainforest-Alliance, 2020). While Sustainable Agriculture focuses on producing crops and livestock with minimal environmental impact, CSA is an approach that aims to assist those who manage agricultural systems in responding effectively to climate change. Thus, CSA practices can be defined as agricultural practices that consider both resilience and adaptation to climate change.

The implementation of CSA practices among smallholder farmers, however, has not achieve the intended goals because the current practices do not consider individual farm-level data and socio-economic characteristics during the design and implementation of the interventions. Individual smallholder farms are different ranging from management practices in each farm, soil characteristics, and other farm-based characteristics. For this reason, the lack of information, insights, and data-driven decisions leads to losses and reduced yields forcing some smallholder farmers to abandon CSA practices with the winding up of supporting projects. Data-driven agriculture informs smallholder farmers on the critical economic decisions of what to produce, how much to produce and when and how much to produce. This study, therefore, designed a data-driven model for the deployment and adaptation of CSA practices among smallholder farmers in Kakamega county.

Many studies have been conducted to model agricultural production. First, Johann et al. (2016) estimated the soil moisture content using an autoregressive error function. This model is suitable to estimate soil moisture in controlled systems applied no no-till machinery. A similar study by Chen et al. (2014) designed a Wireless Sensor Network (WSN) to monitor multi-layer soil temperature and moisture in a farmland field to improve water utilization and to collect basic data for research on soil water infiltration variations for intelligent precision irrigation. Muangprathub et al. (2019) developed a model for optimally irrigating crops based on a Wireless Sensor Network (WSN). In this model, a soil moisture sensor is used to monitor the field and connecting to the control box. A web-based application is designed to manipulate crop data and field information. This application applies data mining to analyze the data for predicting suitable temperature, humidity and soil moisture for optimal future management of crops growth. A mobile smart phone app is then developed to control crop watering.

Another notable model developed in the recent past is the Climate Smart Village Approach by Aggarwal et al. (2018). This model provides a means of performing agricultural research for development through testing technological and institutional options for dealing with climate variability and climate change using participatory methods.

According to Aggarwal et al. (2018), an ideal CSV approach gives guidance before and during the planting season on the most suitable CSA practices, technologies, services, processes, and institutional options considering market and resource availability such as capital, labor and markets.

The Climate Smart Decision Support system for analysing the water demand of a large-scale rice irrigation scheme is one of the models that have been developed to inform Climate Smart Agricultural decisions. This model by Rowshon et al. (2019), was applied to evaluate the impacts of climate change on irrigation water demand and other key hydro-climatic parameters in the Tanjung Karang Irrigation Scheme in Malaysia for the period 2010-2099. This model which has been used for analysing the water demand of a large-scale rice irrigation scheme helps promote adaptation and mitigation strategies that can lead to more sustainable water use at the farm level.

Ascough Li et al. (2002), developed the Great Plains Framework for Agricultural Resource Management (GPFARM), to provide crop and livestock management support at the whole farm level in the Great Plains of the United States. This DSS provides producers, consultants, action agencies, and scientists with information for making management decisions that promote sustainable agriculture. GPFARM contains risk analyses that combine projected crop yield and animal production data with concurrent environmental impact data. Another DSS was developed by Bseiso et al. (2015) targeting greenhouse farmers in low-resource settings. The DSS provides farmers with slides of decision information which is only read through printed papers or in a PDF format. This means that this DSS tool can be made into an app instead of paperwork.

Fourati et al. (2014) present a climatic monitoring system for farmers. Using an integrated WSN weather station, farmers can display weather measures relative to temperature, humidity, wind and solar radiation. These measures allow the DSS to precisely calculate the water requirement in a daily calendar. Another DSS is by Panchard et al. (2007), known as Commonsense net. This DSS is a wireless sensor network for resource-poor agriculture in the semiarid areas of developing countries. This sensor network system aims at improving resource poor farmers' farming strategies in the wake of highly variable conditions. The risk management strategies include choice of crop varieties, planting and harvesting, pests and disease control and efficient use of irrigation water. This decision Support System uses WSN for the improvement of farming strategies in the face of highly variable conditions.

II. METHODOLOGY

Primary Data Collection

Primary data was collected from 428 smallholder farmers in Kakamega County (182 adopters and 246 dis-adopters). The purpose of the models, therefore, was to aid in decision-making through prediction on which smallholder farmers would be CSA adopters and who would be CSA dis-adopters using the different variables identified in the study.

Machine Learning Tools

The Google Collaboratory notebook was used for the model fitting and testing process. *Pandas, Numpy, Matplotlib, Scikit-learn* and *Seaborn* ML libraries were used in the modelling. These libraries were imported into the Collaboratory notebook as shown below:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn import metrics
import seaborn as sns
```

Importing the Data into The Notebook

This involved loading the dataset into a *pandas'* data frame using the read_csv function. The dataset was loaded as follows:

```
df = pd.read_csv("/content/Whole Data Set_610 Variables.csv")
```

Define the X and Y Variables

The dependent variable, V12, was defined as the smallholder CSA respondent categorization in terms of adopters and dis-adopters while the independent variables (X) were all the other variables that influenced the smallholder farmer to be either an adopter or a dis-adopter. The dependent variable (Y) and the independent variables (Y) were defined as follows:

```
X = df[["V17" , "V25" , "V44" , "V144" , "V48" , "V50" , "V133" , "V130" , "V135" ,
"V38" , "V143" , "V120" , "V107" , "V43" , "V169" , "V77" , "V40" , "V22" , "V47" ,
"V104" , "V76" , "V5" , "V112" , "V134" , "V8" , "V57" , "V75" , "V80" , "V119" ,
"V37" , "V165" , "V103" , "V68" , "V121" , "V18" , "V29" , "V58" , "V41" , "V28" ,
"V34" , "V164" , "V115" , "V146" , "V129" , "V141" , "V10" , "V168" , "V4" , "V49"
, "V6" , "V140" , "V145" , "V167" , "V163" , "V139" , "V162" , "V136" , "V161" ,
"V51" , "V138" , "V160"]]
Y = df['V12']
```

Splitting the Data into Training and Test Data Sets

The data were randomly split into two datasets; 70 per cent for training the model and 30 per cent for testing the model. The train and test datasets were set as follows:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=0)
```

RESEARCH ARTICLE
ISSN NO: 2410-8383

Fitting the Models

Model fitting was done to measure how well the ML models generalize to similar data to that on which they were trained. The models were defined and fit as shown on Table 1 below.

Table 1:

Fitting the Models

Classifier Model	Fitting the Model
Decision Tree.	*clf = DecisionTreeClassifier (max_depth = 5, random_state = 0)* *model = clf.fit(X_train, y_train)*
Random Forest	*clf=RandomForestClassifier (n_estimators=10)* *model = clf.fit(X_train, y_train)*

Making Predictions on the Test Data Set

The fitted models were used to fit the test data as follows:

y_pred = clf.predict(X_test)

Comparison of the Actual and Predicted Values

The actual values were compared with the predicted values as per the ML model. The actual values and the predicted values were compared as follows:

df=pd.DataFrame('Actual':y_test, 'Predicted':y_pred)

Model Evaluation

The models were evaluated using the following metrics:
Confusion Matrix; This metric was used in measuring recall, precision, specificity, accuracy, and AUC-ROC curves. The confusion matrix was developed for the models as follows:

metrics.confusion_matrix(y_test, y_pred, labels = [1, 2]).

The other metrics are described on Table 2, below.

Table 2:

Model Evaluation Using Various Metrics

Metric	Narrative
Training Accuracy.	Resultant model accuracy given when the model is applied to the training data implying that the model is tested on the examples it was constructed on
Prediction Accuracy	given by the ratio of the variables that are correctly predicted to the number of times the variables have been predicted in total.
Precision/Sensitivity	the proportion of observed positives that are predicted to be positives.
Recall.	frequency of the correct predictions that are positive values.
Specificity	the proportion of actual negatives that were correctly predicted to be negatives.
F1- Score	This is the harmonic mean of recall and precision. It is a statistical measure of the accuracy of a test or a model
ROC Curve	Presents the visual way of measuring the performance of a binary classifier. It is the ratio of the true positive rate (TPR) and the false positive rate (FPR).
AUC	This is the metric used to find the area under the ROC curve.

The Model AUC-ROC graphs were developed for the models as follows:

```
from sklearn.metrics import roc_curve, AUC
from sklearn.metrics import RocCurveDisplay
ax = plt.gca()
rfc_disp = RocCurveDisplay.from_estimator(model, X_test, y_test, ax=ax,
alpha=0.8)
```

```
plt.show().
```

Classification Report; A Classification report was used to measure the quality of predictions from a classification algorithm in terms of how many predictions were true and how many predictions were wrong. The Classification Report was developed for the models as follows:

```
from sklearn.metrics import classification_report
target_names = ['Adopt', 'Dis-Adopt']
print(classification_report(y_test, y_pred, target_names=target_names))
```

Computing Model Accuracy

Model accuracy was given by the number of classifications that a model predicted accurately divided by the number of predictions made. Mean Absolute Error (MEA), Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) were used to calculate the accuracy of the classification and regression model. The model accuracy, using the various approaches, was computed as follows;

Table 3:

Computing Model Accuracy

Model Accuracy Measure	Calculation
Mean Absolute Error	print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, y_pred))
Mean Squared Error	print('Mean Squared Error:', metrics.mean_squared_error(y_test, y_pred))
Root Mean Squared Error	print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))

Absolute Errors	# Calculate the absolute errors
	errors = abs(y_pred - y_test)
	# Print out the mean absolute error (MAE)
	print('Mean Absolute Error:', round(np.mean(errors), 2))

Mean Absolute Percentage Error	# Calculate mean absolute per centage error (MAPE)
	mape = 100 * (errors / y_test)

Accuracy	# Calculate and display accuracy
	accuracy = 100 - np.mean(mape)
	print('Accuracy:', round(accuracy, 2), '%.')

Plotting the Actual and Predicted Values and the Identification of Important Features

The actual and predicted values were plotted, and the important features identified as shown on Table 4 below.

Table 4:

Plotting the Actual and Predicted Values and the Identification of Important Features

Activity	Process

Plotting actual and predicted values	import seaborn as sns
	plt.figure(figsize=(5, 7))
	ax = sns.distplot(y, hist=False, color="r", label="Actual Value")
	sns.distplot(y_pred, hist=False, color="b", label="Fitted Values", ax=ax)
	plt.title('Actual vs Fitted Values for Adoption vs Dis-adoption')
	plt.show()
	plt.close()

Identification of key features	*pd.DataFrame(model.feature_importances_,index=features).sort_values(by=0, ascending=False)*
	*model.feature_importances_*
	sorted_idx = model.feature_importances_.argsort()
	features = X.columns
	plt.figure(figsize=(10, 15))
	plt.barh(features[sorted_idx], model.feature_importances_[sorted_idx])
	plt.xlabel("Random Forest Feature Importance")
	plt.ylabel("Variables")

Visualizing the Random Forest and the Decision Tree Classifier Models

Tree visualization was used to illustrate how underlying variables (data) predict a chosen target and highlights key insights about the Random Forest Classifier and the decision tree. The Gini index was used to measure the impurity or purity of the decision tree in the Classification and Regression Tree (CART) algorithm. The resulting trees were visualized as shown on Table 5 below.

Table 5:

Visualizing the Random Forest and the Decision Tree Classifier Models

Classifier Model	Visualization
Decision Tree Visualization	*cn = ["Adopt","Disadopt"]* *fig = plt.figure(figsize=(30,10))* *_ = tree.plot_tree(model,* *feature_names=features,* *class_names=cn,* *filled=True, fontsize=12)*
Random Forest Classifier Visualization.	*from sklearn import tree* *cn = ["Adopt","Disadopt"]* *#plt.figure(figsize=(25,15))* *estimator = model.estimators_[5]* *from sklearn.tree import export_graphviz* *# Export as dot file* *export_graphviz(estimator, out_file='tree.dot',* *feature_names = features,* *class_names = cn,* *rounded = True, proportion = False,* *precision = 2, filled = True)* *# Convert to png using system command (requires Graphviz)* *from subprocess import call* *call(['dot', '-Tpng', 'tree.dot', '-o', 'tree.png', '-Gdpi=600'])* *# Display in jupyter notebook* *from IPython.display import Image* *Image(filename = 'tree.png')*

Rapid Prototyping of the Data-Driven Model

This step involved the development of a data-driven prototype that predicts whether a smallholder farmer will either adopt or dis-adopt CSA technologies. Prototyping is the first stage of product development, and it gives the potential users a complete idea of how the final product will look like. The prototype developed was used to simulate a real ground situation. The main aim of the prototype was to attract and inform potential users of a product that they could invest in before allocating resources to and implementation of CSA technologies in Kakamega County. The following steps were followed in this process.

Development of a data collection guide

An online data collection tool was developed for the top 18 variables as identified in Objective 2 as being the most important in influencing the adoption or dis-adoption of CSA technologies in Kakamega.

Primary data collection

A random sample of 15 smallholder farmers, 8 adopters, and 7 dis-adopters, was identified from Butere Subcounty. Their farm biophysical and socioeconomic data were collected based on the top 18 variables identified in objective 2.

Fitting the model

The Google Collaboratory notebook was used for the model fitting and testing process. The prediction capabilities of the model were tested as follows:
Importing the data into the notebook; The dataset was loaded into a *pandas'* data frame using the read_csv function as follows:

```
df_test = pd.read_csv("/content/drive/MyDrive/Model_Testing_15092022.csv")
```

Defining the X and Y variables; this step involved the use of all 18 variables and the resultant secondary independent variables. The independent variable, V12, was defined as the smallholder farmer categorization in terms of adopters and dis-adopters. The independent variables comprised the 18 important variables that were under investigation and the secondary independent variables resulting from the data collection exercise. The independent variables (X) and dependent variables (y) were then defined as follows.

```
X_test=
df_test[["V5","V6","V10","V37","V38","V39","V40","V41","V42","V43","V44","V45","V46","V47","V49","V50","V51","V58","V59","V103","V104","V107","V115","V119","V120","V112","V129","V136","V138","V139","V140","V141","V143","V144","V145","V146","V160","V161","V162","V163","V164","V165","V166","V167","V168","V169"]]
```

Splitting the data into training and test data sets; the data was split into two datasets, 70 per cent for training the model and 30 per cent for testing the model. The data was split as follows:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)
```

Making predictions on the test data set; the test data set was used to gauge the ability of the model to learn from the training data and make accurate predictions when input with new data. The fitted models were used to fit the test data as follows:

```
y_pred = model.predict(X_test)
```

RESEARCH ARTICLE
ISSN NO: 2410-8383

Comparing the actual and predicted values; 15 smallholder CSA farmers were sampled out of which eight were adopters while seven were dis-adopters. The actual values and the predicted values were compared as follows:

df=pd.DataFrame

Model evaluation; the model was evaluated using the metrics on Table 6, below.

Table 6:

Model Evaluation

Metric	Calculation
Absolute Errors	*errors = abs(y_pred - y_test)*
Mean Absolute Error	*print('Mean Absolute Error:', round(np.mean(errors), 2))*
Mean absolute percentage error	*mape = 100 * (errors / y_test)*
Model Accuracy	*accuracy = 100 - np.mean(mape)* *print('Accuracy:', round(accuracy, 2), '%')*
Precision/Sensitivity	*print('Precision:',precision_score(y_test,y_pred))*
Recall	*print('Recall:',recall_score(y_test,y_pred))*
Specificity	*tn, fp, fn, tp = confusion_matrix.ravel()* *specificity = tn / (tn+fp)* *print('Specificity:',specificity)*
F1 score	*print('f1 Score:',f1_score(y_test,y_pred))*

Confusion Matrix	*confusion_matrix = metrics.confusion_matrix(y_test, y_pred, labels = [1, 2])*
	print('Confusion_Matrix')
	print(confusion_matrix)

Classification report	*target_names = ['Adopt', 'Dis-Adopt']* *print(classification_report(y_test, y_pred, target_names=target_names))*

Data-Driven Prototype Evaluation and Piloting

This step involved conducting a focus group discussion with key stakeholders in the CSA ecosystem to get their input in the model development process. This step was important as it brought out the potential users' expectations about the model and the challenges it was meant to solve. In addition, this step was used to determine whether the model was useful to the potential users and to gauge its user-friendliness. The participants included the University academic staff and students, Research Organizations, County Government Agricultural Extension Staff, Smallholder CSA farmers and Organizations promoting CSA technologies among smallholder farmers in Kakamega County. A demonstration was conducted to show the workings of the data-driven model for the deployment and adaptation of CSA practices among Kakamega County's smallholder farmers. Dummy farmer biophysical and socio-economic data was used to predict the possibility of adoption of CSA technologies. The objective of this exercise was to elicit feedback on the applicability and suitability of the data-driven model for the deployment and adoption of CSA practices among Kakamega County's smallholder farmers.

III. RESULTS

Modelling Variables Selection

The study yielded 610 variables. The variables that were found to have a significant correlation at the 0.05 and 0.01 levels (2-tailed) were identified and used in ML Model, as shown in Table 7 below.

Table 7:

Selected Variables for the Classification Model

Variable Code	Variable	Correlation	Variable Code	Variable	Correlation
V17	Radio & TV	-.096*	V103	Group membership	.145**
V25	Computer	-.098*	V68	Solar Radio owned	-.148**
V44	ISLM/ISFM Trained	-.098*	V121	Ext. officer interaction	.152**
V48	CSA Organization	.099*	V18	Barazas	-.155**
V144	G/House abandoned	.099*	V29	Bicycle owned	-.159**
V50	Year Trained	.106*	V58	Land Size	-.161**
V133	Farming	-.106*	V41	Agroforestry Trained	-.162**
V130	Access to agric. credit?	.107*	V28	W/Barrow owned	-.163**
V38	SWC Trained	-.107*	V34	NGO Support?	.166**
V135	Other HH Activities	-.107*	V164	Agroforestry practised	-.166**
V143	ISLM/ISFM abandoned	.108*	V115	Farming	-.170**
V120	Agric credit	-.110*	V146	Vermiculture abandoned	.174**
V107	Left Group	.111*	V129	HH Monthly income	-.183**
V43	G/House Trained	-.112*	V141	PPT Abandoned	.193**
V169	Fallowing Practised	.115*	V10	Education	-.193**
V77	G/Nuts grown	-.115*	V168	Vermiculture Practised	-.197**
V40	PPT Trained	-.116*	V4	Sex	.216**
V22	TV Owned	-.119*	V49	Farming Experience	.216**
V47	Mulching Trained	-.119*	V6	Marital	.217**
V104	Reason not in a group	.120*	V140	SWC Abandoned	.235**
V76	Soybean grown	-.122*	V145	Composting Abandoned	.250**
V5	Age	-.124*	V167	W/Harvesting Practised	-.276**
V112	Position held	.125**	V163	Composting practised	-.304**
V134	Sch. Fees	-.125**	V139	W/Harvesting abandoned	.322**
V8	Decision Maker	.128**	V162	PPT Practised	-.327**
V57	Precision	-.128**	V136	Abandoned CSA Practices?	-.341**
V75	Cassava grown	-.129**	V161	SWC Practised	-.344**
V80	Fruit Trees Grown	-.137**	V51	Farmer Category	.370**
V119	Agric Trainings	-.139**	V138	CA Abandoned	.429**
V37	CA Trained	-.141**	V160	CA Practised	-.549**
V165	ISLM/ISFM Practised	-.143**			

***. Correlation is significant at the 0.01 level (2-tailed).*
**. Correlation is significant at the 0.05 level (2-tailed).*

Modelling for CSA Adoption

Decision tree Classifier and Random Forest Classifier Models for the Prediction of Adoption or Dis-adoption of CSA Practices were considered for prediction and behaviour analysis. The models were evaluated using the following metrics:

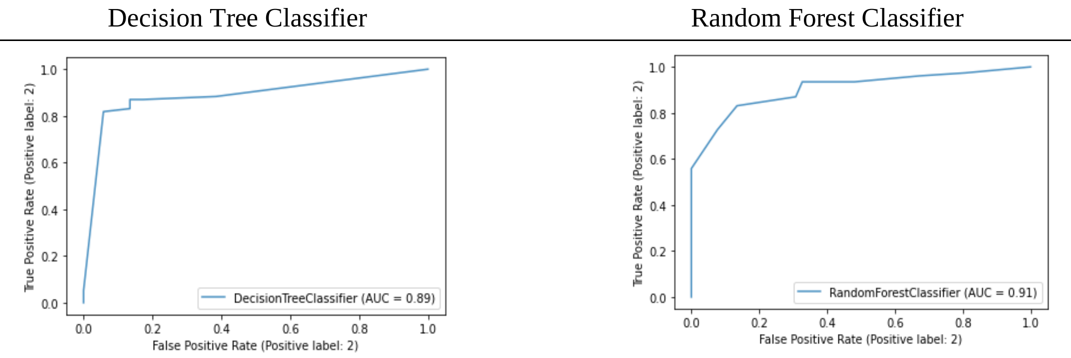
Confusion Matrix: table 8 below depicts the model confusion matrix.

Table 8:
Model Confusion Matrix

Decision Tree Classifier			*Random Forest Classifier*		
	Adopter	Dis-adopter		Adopter	Dis-adopter
Adopter	45	7	Adopter	45	7
Dis-adopter	11	66	Dis-adopter	13	64

Model AUC-ROC graphs; figure 1, below, depicts the model AUC-ROC graphs.

Figure 1:
AUC-ROC graphs



The area under Curve (AUC); as depicted in Figure 1 (above) and Table 9 (below), the models under review produced AUCs of 0.89 and 0.91 under the Decision Tree Classifier and Random Forest Classifier, respectively.

Table 9:
Model Metrics

Metric	Decision Tree Classifier	Random Forest Classifier
Training Accuracy	0.9431438127090301	0.9966555183946488
Prediction Accuracy	0.8604651162790697	0.8449612403100775
Precision / Sensitivity	0.8035714285714286	0.7758620689655172
Recall	0.8653846153846154	0.8653846153846154
Specificity	0.8653846153846154	0.8653846153846154
F1- Score	0.8333333333333334	0.8181818181818181
AUC – ROC	0.89	0.91

Training Accuracy; as shown in Table 9 the models had a training accuracy of 0.943 and 0.996 for the decision tree and random forest classifiers respectively.
Prediction Accuracy; the model prediction accuracy was tested on 30 per cent of the data, and as Table 9 depicts, the models' prediction accuracy was 0.860 and 0.8445 for the decision tree and random forest classifier, respectively.
Precision; as shown in Table 9 the models' evaluation gave precisions of 0.80 and 0.78 for the decision tree and random forest classifier, respectively.
Recall; the model had a Recall of 0.86 for both decision tree classifier and random forest classifier as indicated in Table 9.
Specificity; the Model evaluation gave a specificity of 0.865 for both the decision tree and random forest classifiers as shown in Table 9.
F1 Score; this models had F1 scores of 0.833 and 0.818 for the decision tree classifier and random forest classifier, respectively as indicated in Table 9.
Classification Report; a Classification report was used to measure the quality of predictions from a classification algorithm in terms of how many predictions were true and how many predictions were wrong. Table 10, below, depicts the model classification report.

Table 10:
Model Classification Report

	Decision Tree Classifier				Random Forest Classifier			
	Precision	Recall	F1-Score	Support	Precision	Recall	F1-Score	Support
Adopt	0.80	0.87	0.83	52	0.78	0.87	0.82	52
Dis-Adopt	0.90	0.86	0.88	77	0.90	0.83	0.86	77
Accuracy			0.86	129			0.84	129
macro avg	0.85	0.86	0.86	129	0.84	0.85	0.84	129
weighted avg	0.86	0.86	0.86	129	0.85	0.84	0.85	129

Model Accuracy

Several approaches were used to calculate the accuracy of the classification and regression model. These approaches were the following:
Mean Absolute Error (MEA) Approach; As indicated in Table 11 below, this model had MEAs of 0.13953488372093023 and 0.15503875968992248 for the Decision Tree Classifier and Random Forest Classifier, respectively.
Mean Squared Error (MSE) approach; As indicated in Table 11 below, this model had MSEs of 0.13953488372093023 and 0.15503875968992248 for the Decision Tree Classifier and Random Forest Classifier, respectively.
Root Mean Squared Error (RMSE); As depicted in Table 11 below, the RMSEs for the Decision Tree Classifier and Random Forest Classifier were 0.3735436838188142 and 0.3937496154790789, respectively.
Accuracy; as depicted in Table 11 below, the Accuracy Values for the Decision Tree Classifier and Random Forest Classifier were 90.31 per cent and 89.53 per cent, respectively.

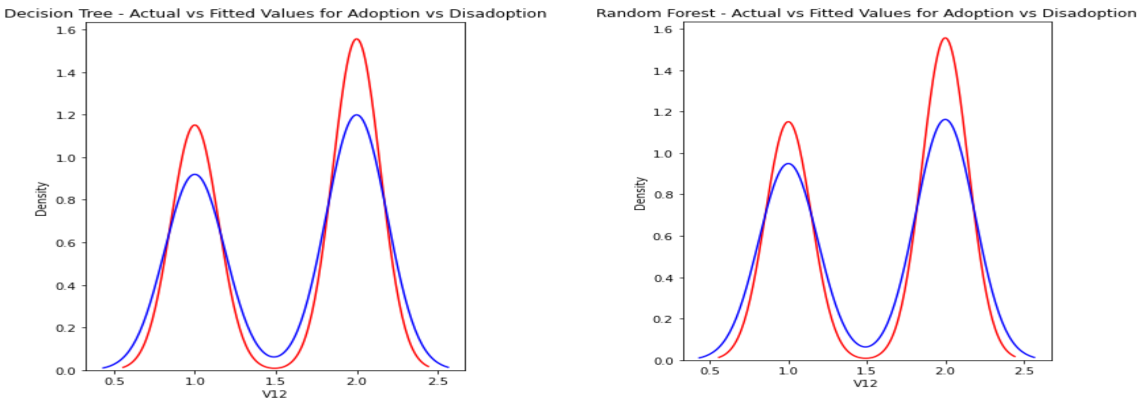
Table 11:
Model Accuracy Using Different Approaches

Model Accuracy	Decision Tree Classifier	Random Forest Classifier

Mean Absolute Error	0.13953488372093023	0.15503875968992248
Mean Squared Error	0.13953488372093023	0.15503875968992248
Root Mean Squared Error	0.3735436838188142	0.3937496154790789
Accuracy	90.31%	89.53%

Plot the Actual Vs Predicted Values; the actual and predicted values were plotted together for visualizing and analysing how the actual data correlate with those predicted by the model. As depicted on Figure 2 below, the plots displayed identical distributions both for the decision tree classifier and the random forest classifier.

Figure 2:
Actual vs Fitted Values for Adoption Vs Dis-adoption



Decision Tree Classifier

Random Forest Classifier

Identification of important Features; the model identified and ranked the following selected variables (features) from the most important to the least important in Predicting the adoption and dis-adoption of CSA technologies among smallholder farmers in Kakamega County. Tables 12 and 13, below, depict the ranking of the Important features in the Decision Tree and Random Forest Classifier, respectively.

Table 12:
Decision Tree Feature (Variable) Importance

Variable	Contribution (%)	Variable	Contribution (%)	Variable	Contribution (%)
V160	0.345985	V4	0	V120	0
V161	0.185694	V168	0	V107	0
V162	0.13754	V18	0	V43	0
V163	0.113938	V6	0	V169	0
V165	0.05611	V140	0	V77	0
V167	0.029799	V145	0	V40	0
V50	0.026095	V48	0	V22	0
V51	0.024686	V139	0	V47	0
V164	0.023641	V144	0	V104	0
V28	0.022019	V136	0	V76	0
V129	0.013212	V44	0	V5	0
V57	0.008783	V138	0	V112	0
V49	0.007431	V29	0	V134	0

V8	0.005067	V121	0	V75	0
V10	0	V133	0	V80	0
V58	0	V68	0	V119	0
V41	0	V130	0	V37	0
V34	0	V135	0	V25	0
V115	0	V38	0	V103	0
V146	0	V143	0	V17	0
V141	0				

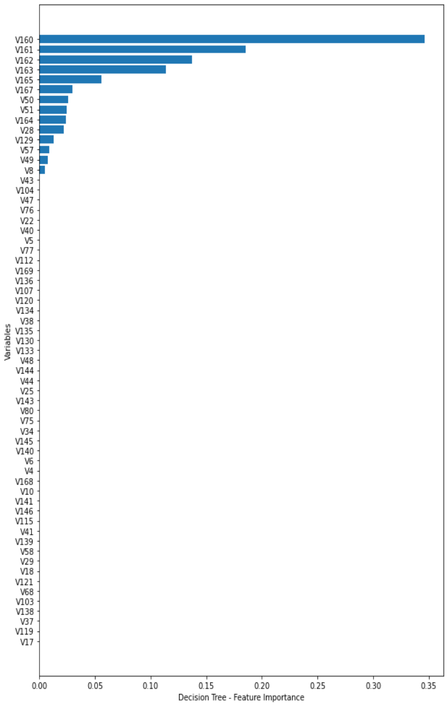
Table 13:
Random Forest Classifier Feature (Variable) Importance

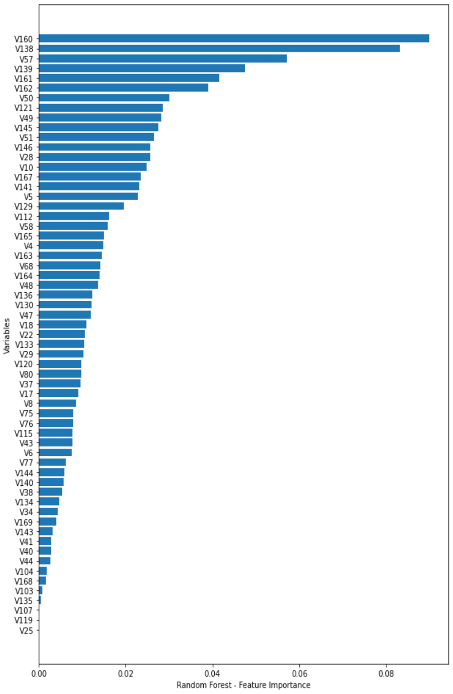
Variable	Contribution (%)	Variable	Contribution (%)	Variable	Contribution (%)
V160	0.15765086	V136	0.01898321	V38	0.00384722
V161	0.10778454	V140	0.01819589	V43	0.00328322
V138	0.08937059	V6	0.01659587	V164	0.00292064
V10	0.03544279	V145	0.01502669	V119	0.00233905
V5	0.03502229	V51	0.01404221	V40	0.00221336
V163	0.03405033	V59	0.01121503	V103	0.00215789
V162	0.03392181	V44	0.01049681	V115	0.00212019
V139	0.03362288	V46	0.0102526	V39	0.00184252
V58	0.0317778	V42	0.01008639	V166	0.00178529
V49	0.030106	V146	0.0095861	V168	0.00166549
V112	0.02888547	V117	0.0095354	V107	0.00158925
V4	0.02571036	V116	0.00940684	V105	0
V120	0.02330297	V41	0.00869023	V109	0
V165	0.02226637	V114	0.00831827	V106	0
V167	0.02223764	V144	0.00711218	V108	0
V50	0.02166705	V45	0.00619122		
V129	0.02114275	V169	0.00610769		
V141	0.02027508	V143	0.00499815		

Figures 3 below, depict the graphical representation of the key features from the most important to the least important.

Figure 3:
Important Features

Decision Tree Classifier	Random Forest Classifier


Variables
V160
V161
V162
V163
V165
V167
V50
V51
V164
V28
V129
V57
V49
V8
V43
V104
V47
V76
V22
V40
V5
V77
V112
V169
V136
V107
V120
V134
V38
V135
V130
V133
V48
V144
V44
V25
V143
V80
V75
V34
V145
V140
V6
V4
V168
V10
V141
V146
V115
V41
V139
V58
V29
V18
V121
V68
V103
V138
V37
V119
V17
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
Decision Tree - Feature Importance


Variables
V160
V138
V57
V139
V161
V162
V50
V121
V49
V145
V51
V146
V28
V10
V167
V141
V5
V129
V112
V58
V165
V4
V163
V68
V164
V48
V136
V130
V47
V18
V22
V133
V29
V120
V80
V37
V17
V8
V75
V76
V115
V43
V6
V77
V144
V140
V38
V134
V34
V169
V143
V41
V40
V44
V104
V168
V103
V135
V107
V119
V25
0.00
0.02
0.04
0.06
0.08
Random Forest - Feature Importance

Visualizing the Random Forest Classifier and the Decision Tree Classifier; the classifiers were visualized illustrate how underlying variables (data) predict a chosen target. Figures 4 and 5, below, depict the decision tree visualization Random Forest Classifier visualization, respectively, in the form of flowchart diagrams.

Figure 4:

Decision Tree Visualization

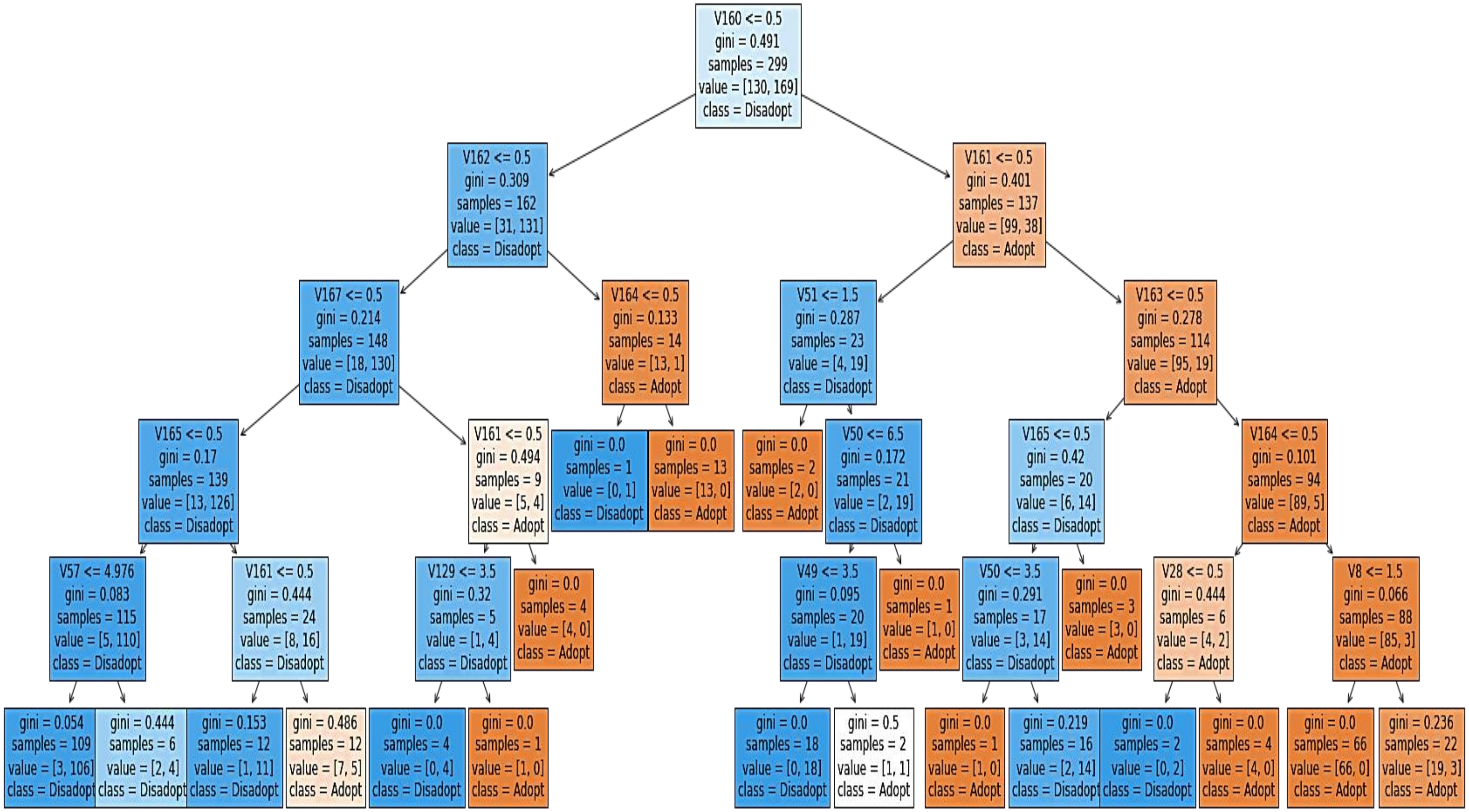
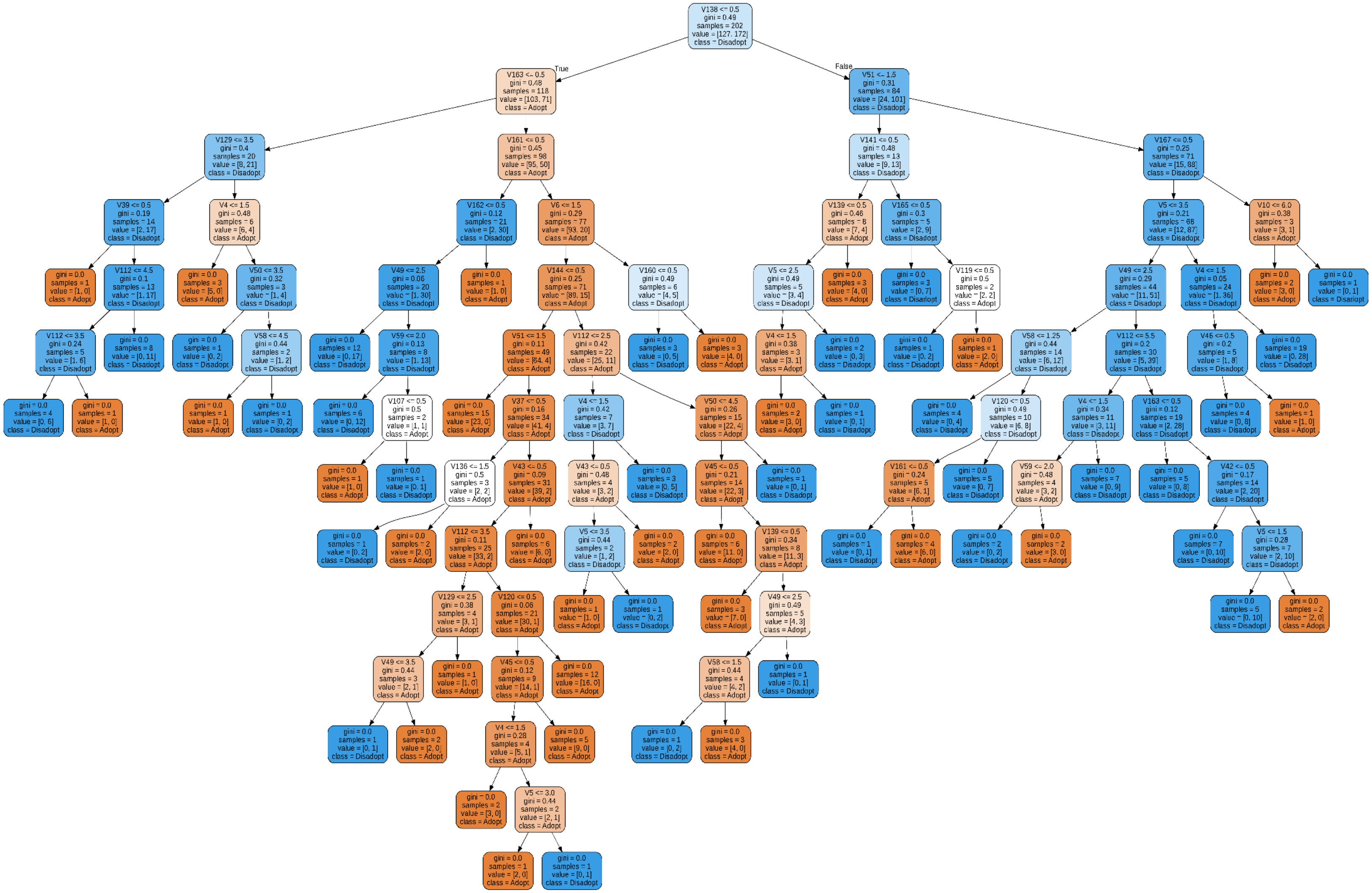


Figure 5:

Random Forest Classifier Visualization


V138 <= 0.5 gini = 0.49 samples = 202 value = [127, 172] class = Disadopt
True
V163 <= 0.5 gini = 0.48 samples = 118 value = [103, 71] class = Adopt
False
V51 <= 1.5 gini = 0.31 samples = 84 value = [24, 101] class = Disadopt
V129 <= 3.5 gini = 0.4 samples = 20 value = [8, 21] class = Disadopt
V161 <= 0.5 gini = 0.45 samples = 98 value = [95, 50] class = Adopt
V141 <= 0.5 gini = 0.48 samples = 13 value = [9, 13] class = Disadopt
V167 <= 0.5 gini = 0.25 samples = 71 value = [15, 88] class = Disadopt
V39 <= 0.5 gini = 0.19 samples = 14 value = [2, 17] class = Disadopt
V4 <= 1.5 gini = 0.48 samples = 6 value = [6, 4] class = Adopt
V162 <= 0.5 gini = 0.12 samples = 21 value = [2, 30] class = Disadopt
V6 <= 1.5 gini = 0.29 samples = 77 value = [93, 20] class = Adopt
V139 <= 0.5 gini = 0.46 samples = 8 value = [7, 4] class = Adopt
V165 <= 0.5 gini = 0.3 samples = 5 value = [2, 9] class = Disadopt
V5 <= 3.5 gini = 0.21 samples = 68 value = [12, 87] class = Disadopt
V10 <= 6.0 gini = 0.38 samples = 3 value = [3, 1] class = Adopt
gini = 0.0 samples = 1 value = [1, 0] class = Adopt
V112 <= 4.5 gini = 0.1 samples = 13 value = [1, 17] class = Disadopt
gini = 0.0 samples = 3 value = [5, 0] class = Adopt
V50 <= 3.5 gini = 0.32 samples = 3 value = [1, 4] class = Disadopt
V49 <= 2.5 gini = 0.06 samples = 20 value = [1, 30] class = Disadopt
gini = 0.0 samples = 1 value = [1, 0] class = Adopt
V144 <= 0.5 gini = 0.25 samples = 71 value = [89, 15] class = Adopt
V160 <= 0.5 gini = 0.49 samples = 6 value = [4, 5] class = Disadopt
V5 <= 2.5 gini = 0.49 samples = 5 value = [3, 4] class = Disadopt
gini = 0.0 samples = 3 value = [4, 0] class = Adopt
gini = 0.0 samples = 3 value = [0, 7] class = Disadopt
V119 <= 0.5 gini = 0.5 samples = 2 value = [2, 2] class = Adopt
V49 <= 2.5 gini = 0.29 samples = 44 value = [11, 51] class = Disadopt
V4 <= 1.5 gini = 0.05 samples = 24 value = [1, 36] class = Disadopt
gini = 0.0 samples = 2 value = [3, 0] class = Adopt
gini = 0.0 samples = 1 value = [0, 1] class = Disadopt
V112 <= 3.5 gini = 0.24 samples = 5 value = [1, 6] class = Disadopt
gini = 0.0 samples = 8 value = [0, 11] class = Disadopt
gini = 0.0 samples = 1 value = [0, 2] class = Disadopt
V58 <= 4.5 gini = 0.44 samples = 2 value = [1, 2] class = Disadopt
gini = 0.0 samples = 12 value = [0, 17] class = Disadopt
V59 <= 2.0 gini = 0.13 samples = 8 value = [1, 13] class = Disadopt
V51 <= 1.5 gini = 0.11 samples = 49 value = [64, 4] class = Adopt
V112 <= 2.5 gini = 0.42 samples = 22 value = [25, 11] class = Adopt
gini = 0.0 samples = 3 value = [0, 5] class = Disadopt
gini = 0.0 samples = 3 value = [4, 0] class = Adopt
V4 <= 1.5 gini = 0.38 samples = 3 value = [3, 1] class = Adopt
gini = 0.0 samples = 2 value = [0, 3] class = Disadopt
gini = 0.0 samples = 1 value = [0, 2] class = Disadopt
gini = 0.0 samples = 1 value = [2, 0] class = Adopt
V58 <= 1.25 gini = 0.44 samples = 14 value = [6, 12] class = Disadopt
V112 <= 5.5 gini = 0.2 samples = 30 value = [5, 39] class = Disadopt
V46 <= 0.5 gini = 0.2 samples = 5 value = [1, 8] class = Disadopt
gini = 0.0 samples = 19 value = [0, 28] class = Disadopt
gini = 0.0 samples = 4 value = [0, 6] class = Disadopt
gini = 0.0 samples = 1 value = [1, 0] class = Adopt
gini = 0.0 samples = 1 value = [1, 0] class = Adopt
gini = 0.0 samples = 1 value = [0, 2] class = Disadopt
gini = 0.0 samples = 6 value = [0, 12] class = Disadopt
V107 <= 0.5 gini = 0.5 samples = 2 value = [1, 1] class = Adopt
gini = 0.0 samples = 15 value = [23, 0] class = Adopt
V37 <= 0.5 gini = 0.16 samples = 34 value = [41, 4] class = Adopt
V4 <= 1.5 gini = 0.42 samples = 7 value = [3, 7] class = Disadopt
V50 <= 4.5 gini = 0.26 samples = 15 value = [22, 4] class = Adopt
gini = 0.0 samples = 2 value = [3, 0] class = Adopt
gini = 0.0 samples = 1 value = [0, 1] class = Disadopt
gini = 0.0 samples = 4 value = [0, 4] class = Disadopt
V120 <= 0.5 gini = 0.49 samples = 10 value = [6, 8] class = Disadopt
V4 <= 1.5 gini = 0.34 samples = 11 value = [3, 11] class = Disadopt
V163 <= 0.5 gini = 0.12 samples = 19 value = [2, 28] class = Disadopt
gini = 0.0 samples = 4 value = [0, 8] class = Disadopt
gini = 0.0 samples = 1 value = [1, 0] class = Adopt
gini = 0.0 samples = 1 value = [1, 0] class = Adopt
gini = 0.0 samples = 1 value = [0, 1] class = Disadopt
V136 <= 1.5 gini = 0.5 samples = 3 value = [2, 2] class = Adopt
V43 <= 0.5 gini = 0.09 samples = 31 value = [39, 2] class = Adopt
V43 <= 0.5 gini = 0.48 samples = 4 value = [3, 2] class = Adopt
gini = 0.0 samples = 3 value = [0, 5] class = Disadopt
V45 <= 0.5 gini = 0.21 samples = 14 value = [22, 3] class = Adopt
gini = 0.0 samples = 1 value = [0, 1] class = Disadopt
V161 <= 0.5 gini = 0.24 samples = 5 value = [6, 1] class = Adopt
gini = 0.0 samples = 5 value = [0, 7] class = Disadopt
V59 <= 2.0 gini = 0.48 samples = 4 value = [3, 2] class = Adopt
gini = 0.0 samples = 7 value = [0, 9] class = Disadopt
gini = 0.0 samples = 5 value = [0, 8] class = Disadopt
V42 <= 0.5 gini = 0.17 samples = 14 value = [2, 20] class = Disadopt
gini = 0.0 samples = 1 value = [0, 2] class = Disadopt
gini = 0.0 samples = 2 value = [2, 0] class = Adopt
V112 <= 3.5 gini = 0.11 samples = 25 value = [33, 2] class = Adopt
gini = 0.0 samples = 6 value = [6, 0] class = Adopt
V5 <= 3.5 gini = 0.44 samples = 2 value = [1, 2] class = Disadopt
gini = 0.0 samples = 2 value = [2, 0] class = Adopt
gini = 0.0 samples = 6 value = [11, 0] class = Adopt
V139 <= 0.5 gini = 0.34 samples = 8 value = [11, 3] class = Adopt
gini = 0.0 samples = 1 value = [0, 1] class = Disadopt
gini = 0.0 samples = 4 value = [6, 0] class = Adopt
gini = 0.0 samples = 2 value = [0, 2] class = Disadopt
gini = 0.0 samples = 2 value = [3, 0] class = Adopt
gini = 0.0 samples = 7 value = [0, 10] class = Disadopt
V5 <= 1.5 gini = 0.28 samples = 7 value = [2, 10] class = Disadopt
V129 <= 2.5 gini = 0.38 samples = 4 value = [3, 1] class = Adopt
V120 <= 0.5 gini = 0.06 samples = 21 value = [30, 1] class = Adopt
gini = 0.0 samples = 1 value = [1, 0] class = Adopt
gini = 0.0 samples = 1 value = [0, 2] class = Disadopt
gini = 0.0 samples = 3 value = [7, 0] class = Adopt
V49 <= 2.5 gini = 0.49 samples = 5 value = [4, 3] class = Adopt
gini = 0.0 samples = 5 value = [0, 10] class = Disadopt
gini = 0.0 samples = 2 value = [2, 0] class = Adopt
V49 <= 3.5 gini = 0.44 samples = 3 value = [2, 1] class = Adopt
gini = 0.0 samples = 1 value = [1, 0] class = Adopt
V45 <= 0.5 gini = 0.12 samples = 9 value = [14, 1] class = Adopt
gini = 0.0 samples = 12 value = [16, 0] class = Adopt
V58 <= 1.5 gini = 0.44 samples = 4 value = [4, 2] class = Adopt
gini = 0.0 samples = 1 value = [0, 1] class = Disadopt
gini = 0.0 samples = 1 value = [0, 1] class = Disadopt
gini = 0.0 samples = 2 value = [2, 0] class = Adopt
V4 <= 1.5 gini = 0.28 samples = 4 value = [5, 1] class = Adopt
gini = 0.0 samples = 5 value = [9, 0] class = Adopt
gini = 0.0 samples = 1 value = [0, 2] class = Disadopt
gini = 0.0 samples = 3 value = [4, 0] class = Adopt
gini = 0.0 samples = 2 value = [3, 0] class = Adopt
V5 <= 3.0 gini = 0.44 samples = 2 value = [2, 1] class = Adopt
gini = 0.0 samples = 1 value = [2, 0] class = Adopt
gini = 0.0 samples = 1 value = [0, 1] class = Disadopt

RESEARCH ARTICLE
ISSN NO: 2410-8383

The ML model was piloted with 15 randomly selected smallholder CSA farmers from Butere Sub County. As depicted in Table 14 below, the model accurately predicted 12 out of the 15 farmers.

Table 14:
Comparison between Actual and Predicted Values

Index	Actual	Predicted	Type of Prediction
0	2	2	Accurate
1	1	1	Accurate
2	1	1	Accurate
3	1	1	Accurate
4	1	1	Accurate
5	1	1	Accurate
6	1	1	Accurate
7	2	1	Non-accurate
8	2	2	Accurate
9	1	1	Accurate
10	2	1	Non-accurate
11	2	2	Accurate
12	1	1	Accurate
13	2	1	Non-accurate
14	2	2	Accurate

Model Evaluation

Table 15, below, presents the Pilot Model Metrics. The piloting of the model gave a precision of 73 per cent implying that it had a high level of prediction precision. Further, the model had a Recall of 1 implying that the model had no false negatives. The Model evaluation gave a specificity of 1 implying that the model could accurately predict all the smallholder CSA adopting farmers. The model had an F1 score of 0.8421 implying that it is a good model in predicting smallholder CSA farmer ability to adopt or dis-adopt CSA technologies.

Table 15:
Pilot Model Metrics

Metric	Random Forest Classifier
Precision / Sensitivity	0.7272727272727273
Recall	1.0
Specificity	1.0
F1- Score	0.8421052631578948
Accuracy	90.0%

Confusion Matrix

The confusion matrix was used to visualize the performance of the ML Algorithm. As shown in Table 16 below, the Model gave 8 True positives, 3 False Positives, 0 False negatives and 4 True Negatives. These values were used to calculate model metrics such as model accuracy, F1 score, specificity, recall and precision. The pilot model prediction accuracy in this case was 80 per cent.

Table 16:
Confusion Matrix

	Adopter	Dis-adopter
Adopter	8	0
Dis-adopter	3	4

Classification report

Table 17, below, depicts the model classification report.

Table 17:
Model Classification Report

	Precision	Recall	F1-Score	Support
Adopt	0.73	1.00	0.84	8
Dis-Adopt	1.00	0.57	0.73	7
Accuracy			0.80	15
macro avg	0.86	0.79	0.78	15
weighted avg	0.85	0.80	0.79	15

IV. DISCUSSION

The ML Model predictions were compared with the actual values to determine their predictive accuracy. The confusion matrix was used to visualize the performance of the ML Algorithms. The Decision Tree Classifier had a prediction accuracy of 86.05 per cent with 45 True positives, 11 False Positives, 7 False negatives and 66 True Negatives. The Random Forest Classifier, on the other hand had a prediction accuracy of 84.50 per cent with 45 True positives, 13 False Positives, 7 False negatives and 64 True Negatives. The Area Under Curve imply that both models were excellent and had a good measure of separability. The training accuracy of the two models imply that they (models) could predict accurately a high number of smallholder CSA farmers. The prediction accuracy of the models indicate that the models have high prediction abilities given that the testing data was completely new to them. The precision given by the models imply that that they were pretty good in their prediction ability. The Recall of both models imply that they (models) had no false negatives. The specificity of the models imply that they (models) could accurately predict 86.5 per cent of the smallholder CSA adopting farmers.

The Mean Absolute Errors, Mean Squared Errors, Root Mean Squared Errors and the accuracy indicate that the models only had a few errors had high accuracy and, therefore, were good models to predict the adoption of CSA practices among smallholder farmers in Kakamega County. The visualization of the classifiers gives the various levels of importance of the different variables in predicting the farmer categorization. For the decision tree the most important variables is V160. Variables V161 and V162 are on the second most important level followed by variables V167, V164, V51 and V163. For the Random Forest, on the other hand, the most important variable on the first level is V138. Variables V163 and V51 are on the second most important level as variables V129, V161, V141 and V167 are on the third level of importance. The ML model was piloted with 15 randomly selected smallholder CSA farmers from a new area and correctly predicted 12 of them.

This implies that, given a new data set, the ML model could accurately predict smallholder CSA farmer's ability to adopt CSA technologies. The pilot model Precision, Recall, Specificity and F1 scores showed that the models had a high level of prediction precision and were good models in predicting smallholder CSA farmer ability to adopt or dis-adopt CSA technologies.

Conclusion

Using the random forest classifier and decision tree, it was found that it was possible to predict which smallholder farmers would be CSA technology adopters and which ones would be dis-adopters. These findings will go a long way to solve the farmer's problem of dis-adoption of CSA technologies. These models are able to guide extension officers and policy makers on the right interventions for smallholder farmers in Kakamega County. Ultimately, data-driven agriculture will inform the smallholder farmers on the critical economic decisions of what to produce, how much to produce and when and how much to produce.

Recommendations

This study considered the adoption of bundled CSA technologies among smallholder farmers in Kakamega County. A study that targets commercial and large-scale farmers in Kakamega and other areas encouraged as it would enhance the findings of this study and support the United Nations Sustainable Development Cooperation framework principle of Leaving No One Behind. Future research should also seek to model the adoption of CSA technologies through larger samples that would cover bigger regions such as the former Western Province or the Western Region including the former Nyanza Province. The adoption of individual CSA technologies may be influenced by the different biophysical and socio-economic characteristics that are specific to the technology. For this reason, future studies, and the development of models for the sustainable deployment of specific CSA technologies should be considered. Future studies may also focus on seasonal and crop-specific CSA technologies.

REFERENCES

Aaron, J. (2012). A framework for the development of smallholder farmers through cooperative development. *Department of Agriculture, Forestry and Fisheries. Republic of South Africa*, 1-8.

Aggarwal, P. K., Jarvis, A., Campbell, B. M., Zougmoré, R. B., Khatri-Chhetri, A., Vermeulen, S. J., . . . Bonilla-Findji, O. (2018). The climate-smart village approach: framework of an integrative strategy for scaling up adaptation options in agriculture.

Ascough Ii, J., Shaffer, M. J., Hoag, D. L., McMaster, G. S., Ahuja, L. R., & Weltz, M. (2002). GPFARM: An integrated decision support system for sustainable Great Plains agriculture.

Bseiso, A., Abele, B., Ferguson, S., Lusch, P., & Mehta, K. (2015, 8-11 Oct. 2015). A decision support tool for greenhouse farmers in low-resource settings. 2015 IEEE Global Humanitarian Technology Conference (GHTC),

Chen, K.-T., Zhang, H.-H., Wu, T.-T., Hu, J., Zhai, C.-Y., & Wang, D. (2014). Design of monitoring system for multilayer soil temperature and moisture based on WSN. 2014 International Conference on Wireless Communication and Sensor Network,

Climate Change Act, (2016). Republic of Kenya. http://kenyalaw.org/kl/fileadmin/pdfdownloads/Acts/ClimateChangeActNo11of2016.pdf

Cornish, G. (1998). *Modern Irrigation Technologies for Smallholders in Developing Countries*. Immediate Technology.

Eitzinger, A., Läderach, P., Sonder, K., A, S., G, S., Beebe, S. E., . . . Nowak, A. (2013). Tortillas on the roaster: Central America's maize–bean systems and the changing climate [Brief]. Retrieved 2013-02, from https://cgspace.cgiar.org/handle/10568/34958

FAO. (2020). *Climate-Smart Agriculture*. http://www.fao.org/climate-smart-agriculture/en/

Fourati, M. A., Chebbi, W., & Kamoun, A. (2014). Development of a web-based weather station for irrigation scheduling. 2014 Third IEEE International Colloquium in Information Science and Technology (CIST),

Harvey, M., & Pilgrim, S. (2011). The new competition for land: Food, energy, and climate change. *Food policy*, *36*, S40-S51.

Hlophe-Ginindza, S. N., & Mpandeli, N. S. (2021). The Role of Small-Scale Farmers in Ensuring Food Security in Africa. *Food Secur. Afr*.

Johann, A. L., de Araújo, A. G., Delalibera, H. C., & Hirakawa, A. R. (2016). Soil moisture modeling based on stochastic behavior of forces on a no-till chisel opener. *Computers and Electronics in Agriculture*, *121*, 420-428. https://doi.org/https://doi.org/10.1016/j.compag.2015.12.020

KARI. (2009). *The Major Challenges Of The Agricultural Sector In Kenya*. https://www.kari.org/the-major-challenges/

Kirimi, L., Sitko, N., Ts, J., Karin, F., Muyanga, M., Sheahan, M., . . . Bor, G. (2011). A farm gate-to-consumer value chain analysis of Kenya's maize marketing system.

Maru, A., Berne, D., Beer, J. d., Ballantyne, P. G., Pesce, V., Kalyesubula, S., . . . Chavez, J. (2018). Digital and data-driven agriculture: Harnessing the power of data for smallholders.

Muangprathub, J., Boonnam, N., Kajornkasirat, S., Lekbangpong, N., Wanichsombat, A., & Nillaor, P. (2019). IoT and agriculture data analysis for smart farm. *Computers and electronics in agriculture*, *156*, 467-474.

Panchard, J., Prabhakar, T., Hubaux, J.-P., & Jamadagni, H. (2007). Commonsense net: A wireless sensor network for resource-poor agriculture in the semiarid areas of developing countries. *Information Technologies & International Development*, *4*(1), pp. 51-67.

Rainforest-Alliance. (2020). *What Is Climate-Smart Agriculture?* @RainforestAlliance. https://www.rainforest-alliance.org/articles/what-is-climate-smart-agriculture

Rowshon, M. K., Dlamini, N. S., Mojid, M. A., Adib, M. N. M., Amin, M. S. M., & Lai, S. H. (2019). Modeling climate-smart decision support system (CSDSS) for analyzing water demand of a large-scale rice irrigation scheme. *Agricultural Water Management, 216*, 138-152. https://doi.org/10.1016/j.agwat.2019.01.002