KLASIFIKASI SPESIES RIANG-RIANG MENGGUNAKAN TEKNIK HUTAN RAWAK

Chin Khai Hoong
Dr. Afzan Adam
Dr. Johari Jalinas

Fakulti Teknologi & Sains Maklumat, Universiti Kebangsaan Malaysia

ABSTRAK

Menurut Dr. Johari Jalinas dari Fakulti Sains dan Teknologi (FST) Universiti Kebangsaan Malaysia (UKM), perbandingan ciri-ciri suara panggilan riang-riang secara manual masih lagi digunakan bagi melakukan pengelasan spesies. Cara manual ini merupakan satu kaedah yang perlahan dan mudah terdedah kepada kesilapan manusia. Projek ini merupakan satu kajian untuk membangun sistem pengelasan spesies riang-riang bagi menggantikan kaedah manual tersebut dengan mengaplikasikan teknik pembelajaran mesin, \textit{Random Forest} bagi mengira kebarangkalian kepunyaan spesies diberikan ciri-ciri suara panggilan riang-riang. Modal pengelasan akan dibangun menggunakan informasi mengenai ciri suara panggilan riang-riang yang diperolehi daripada pakar domain, Dr. Johari Jalinas. Sistem yang telah dibangun pada akhir projek ini akan diuji oleh beliau untuk menilai prestasinya.

1 PENGENALAN

Cicada ataupun disebut riang-riang dalam Bahasa Melayu adalah sejenis serangga unik yang mengeluarkan suara panggilan yang bising. Walaupun biasanya dikenal daripada suara panggilan mereka, hanya riang-riang jantan sahaja yang mengeluarkan suara tersebut (Mogzai 2019). Putaran hidup riang-riang bermula dari sebiji telur, kemudian menetas sebagai ulat yang masuk ke dalam tanah dan keluar semula sebagai riang-riang. Riang-riang dewasa kemudian akan mengawan dan mati. Di dunia ini, terdapat kira-kira 3000 jenis spesies riang-riang yang telah ditemui oleh saintis dan banyak lagi spesies yang masih belum dikenalpasti.

Pengelasan spesies merupakan satu proses yang penting bagi membolehkan para saintis mengenal pasti dan memberi nama kepada sesuatu organisma melalui satu sistem yang selaras. Proses ini biasanya melibatkan analisis terhadap sesuatu ciri penting organisma tersebut bagi mengenal pasti kepunyaan spesies. Menurut Dr. Johari Jalinas, seorang pensyarah dari fakulti Sains dan Teknologi (FST) Universiti Kebangsaan Malaysia, kaedah manual untuk menganalisis ciri-ciri suara panggilan bagi mengelas spesies riang-riang masih

lagi digunakan dan proses tersebut amat perlahan dan membebankan terutamanya apablia bilangan sampel yang perlu dikelas bertambah banyak.

Projek ini adalah hasil daripada kerjasama dengan Dr. Johari Jalinas untuk menggabungkan penggunaan IT bagi mengautomatikkan proses menganalisis ciri suara panggilan riang-riang untuk tujuan pengelasan spesies.

2 PENYATAAN MASALAH

Dalam proses pengelasan spesies riang-riang, ciri-ciri suara panggilan perlu diekstrak melalui analisis terhadap suara panggilan mereka. Selepas itu, ciri-ciri suara panggilan ini perlu dianalisis bagi mengenal pasti spesies kepunyaan suara panggilan tersebut. Menurut Dr. Johari Jalinas, kaedah manual yang masih lagi digunakan untuk menganalisis ciri-ciri suara panggilan riang-riang bagi mengelas spesies amat perlahan dan membebankan dan isu ini bertambah serius apabila bilangan sampel yang perlu dikelas bertambah banyak. Selain itu, penglibatan manusia dalam apa-apa jua proses juga bererti ada kemungkinan berlakunya kesilapan manusia yang tidak dapat dielak. Oleh itu, beliau mengharapkan bahawa kedah manual tersebut dapat diganti dengan kaedah automatik melalui penggunaan sistem cerdas.

3 OBJEKTIF KAJIAN

Tujuan utama projek ini adalah untuk membangun sistem cerdas yang mampu menggantikan kaedah manual untuk mengelas spesies riang-riang yang senang diguna dan mudah difahami oleh pengguna akhir termasuk pengguna yang tidak celik IT. Bagi mencapai tujuan tersebut, tiga objektif perlu dilaksanakan iaitu:

1. Mengkaji antara beberapa kaedah pengelasan untuk mengenal pasti kaedah yang sesuai untuk mengelas spesies riang-riang

2. Membangun modal pengelasan spesies yang mengaplikasikan teknik pembelajaran mesin

3. Membangun antaramuka pengguna (UI) yang senang difahami dan mudah untuk diguna

4 METOD KAJIAN

Bagi memastikan kejayaan projek ini, langkah-langkah yang perlu dijalankan bagi setiap fasa perlu dirancang terlebih dahulu. Metod kajian yang efektif akan memastikan kemajuan projek yang lancar. Fasa yang perlu dijalankan bagi projek ini boleh dilihat seperti yang ditunjukkan di bawah.

4.1 Fasa Perancangan

Fasa ini bertujuan untuk menyenaraikan setiap turutan langkah yang perlu dijalankan bagi projek ini. Ia melibatkan proses pengenalpastian masalah, objektif kajian, cadangan penyelesaian, skop kajian dan sebagainya. Kesemua ini perlu ditentukan bagi menghasilkan cadangan projek dan memberi gambar keseluruhan untuk projek ini. Sebab dan objektif bagi menjalakan projek ini juga diterangkan dengan jelas.

4.2 Fasa Analisis

Fasa ini melibatkan kajian sastera bagi membanding kajian/sistem yang sedia ada dengan cadangan penyelesaian bagi projek ini untuk membuat penambahbaikan jika perlu. Kajian lepas yang diuji telah memberikan sumbangan besar terhadap projek ini. Ciri-ciri suara panggilan yang diperlukan untuk mengelas spesies riang-riang bagi projek ini telah berjaya dikenal pasti melalui kajian lepas yang bertajuk 'Kepelbagaian Bunyi Panggilan Riang-Riang di Empat Hutan Terpilih di Semenanjung Malaysia'. Kajian tersebut telah berjaya mencirikan suara panggilan riang-riang pelbagai spesies menggunakan lima ciri iaitu *Echeme Period*, *Echeme Duration*, *Inter-Echeme Duration*, *No. of Echeme per Second* dan *Dominant Frequency*. Projek ini menggunakan tiga ciri untuk mengelas spesies iaitu *Echeme Period*, *Echeme Duration* dan *Inter-Echeme Duration*. Dua ciri tidak digunakan kerana wujudnya kekurangan data dan data yang tidak seragam bagi kedua-dua ciri tersebut.

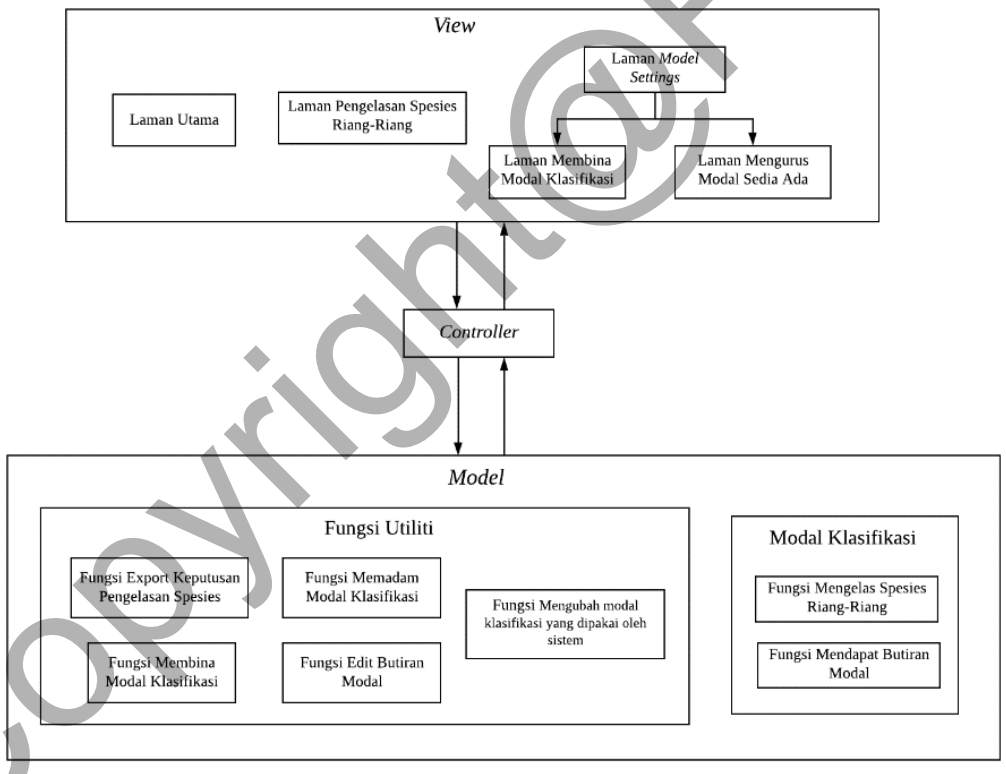
Species	Echeme period (s)	Echeme duration (s)	Inter-echeme duration (s)	No. of echeme / s	Dominant frequency (kHz)
Platylomia spinosa	0.164 ± 0.01	0.100 ± 0.03	0.064 ± 0.08	8.571	3.202
Pomponia near *picta*	0.118	0.086	0.032	6.526	3.180
Orientopsaltria sp. 01	0.381 ± 0.07	0.233 ± 0.00	0.149 ± 0.07	2.545	3.5 -3.7
Chremistica sp. 01	0.531 ± 0.99	0.254 ± 0.05	0.277 ± 0.96	1.913	2.652
Purana sp. 01	0.841 ± 0.82	0.548 ± 0.12	0.292 ± 0.84	1.697	8.587
Mogannia sp. 01	0.188 ± 0.05	0.098 ± 0.02	0.090 ± 0.05	2.977	10.912
Dundubia vaginata	1.937 ± 0.76	1.874 ± 0.81	0.063 ± 0.08	0.621	5.303
Purana sp. 02	2.588 ± 0.26	1.595 ± 0.23	0.994 ± 0.21	0.381	3.091
Orientopsaltria sp. 02	3.416 ± 2.28	2.065 ± 1.07	1.351 ± 1.78	0.335	3.361
Purana sp. 03	2.503 ± 0.51	1.420 ± 0.49	1.083 ± 0.34	0.332	3.080
Sp. 01	0.321 ± 0.05	0.163 ± 0.06	0.158 ± 0.08	3.587	4.7-5.2
Sp. 02	0.250 ± 0.07	0.163 ± 0.05	0.087 ± 0.05	4.061	4.4 – 5.1
Sp. 03	0.140 ± 0.05	0.127 ± 0.05	0.013 ± 0.00	7.379	4.018

Rajah 1 Cici-ciri suara panggilan riang-riang

4.3 Fasa Reka Bentuk

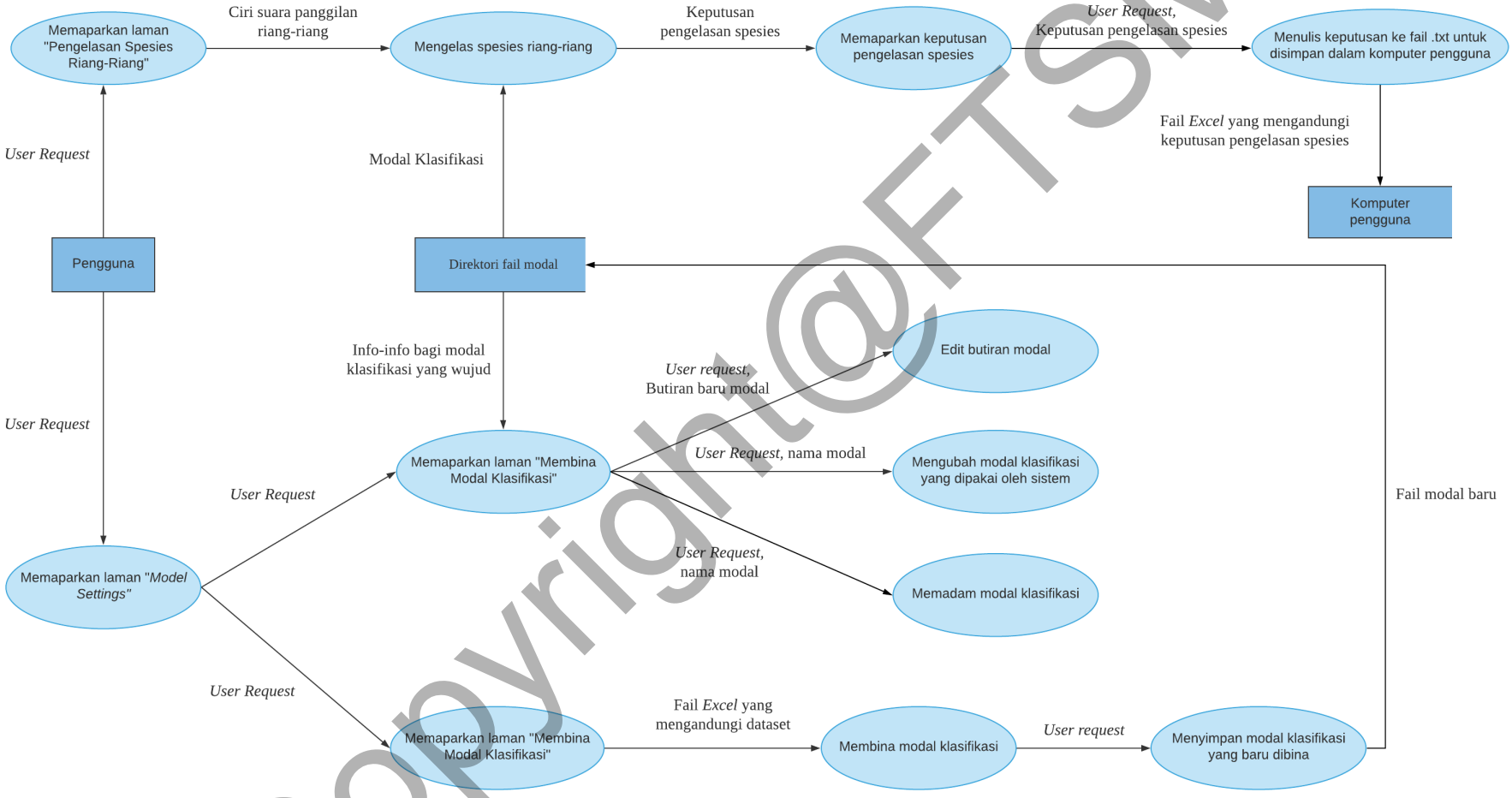
4.3.1 Reka Bentuk Sistem

Fasa ini menghasilkan pelan reka bentuk bagi pelbagai aspek projek ini termasuklah seni bina sistem, pangkalan data, antara muka dan algoritm. Seni bina sistem projek ini mengamalkan corak reka bentuk MVC (*Model-View-Controller*). Dalam corak MVC, keseluruhan sistem akan dibahagikan kepada tiga komponen, iaitu *Model*, *View* dan *Controller*. Setiap komponen ini mempunyai peranan masing-masing yang tidak bertindih dengan komponen yang lain. *Model* merupakan komponen yang mengandungi dan berperanan memproses segala data yang akan digunakan oleh sistem. *Model* bagi sistem ini bertanggunajawab untuk mengendali operasi mengenai modal klasifikasi spesies yang akan digunakan oleh sistem. Jika sesuatu modal tidak digunakan lagi, *Model* juga mampu memadam modal tersebut.



Rajah 2 Seni bina sistem yang bercorak MVC

Rajah aliran data dapat menggambarkan aliran data antara proses-proses dalam sesebuah sistem. Melalui rajah tersebut, sumber dan destinasi bagi setiap data dapat dilihat dan setiap langkah/proses yang digunakan oleh sistem untuk melaksanakan sesuatu fungsi lebih senang difahami



Rajah 3 Rajah Aliran Data

4.3.2 Reka Bentuk Algoritma

Bagi reka bentuk algoritma, perbandingan antara kaedah klasifikasi perlu dijalankan bagi menentukan kaedah klasifikasi yang paling sesuai bagi tugas kajian ini iaitu pengelasan spesies riang-riang. Selepas kaedah klasifikasi telah ditentukan, maka reka bentuk algoritma boleh dilaksanakan. *Hyperparameter* bagi setiap kaedah kasifikasi juga telah dioptimumkan semasa perbandingan dan kesemua ini dijalankan menggunakan perisian pembelajaran mesin, WEKA. Bagi projek ini, kaedah Random Forest telah memberikan nilai prestasi yang paling tinggi semasa proses perbandingan dan akan diaplikasikan dalam sistem sebagai kaedah klasifikasi yang akan digunakan untuk mengelas spesies riang-riang. Proses dan keputusan perbandingan antara kaedah klasifikasi akan dibincangkan pada bahagian hasil kajian manakala bahagian ini akan membincangkan reka bentuk algoritma *Random Forest*.

Rajah pseudokod di bawah menunjukkan kaedah untuk membina banyak *decision tree* yang membentukkan satu *random forest*. Untuk membina satu *tree*, algoritma pembinaan *tree* akan mengira nilai *information gain* bagi setiap nilai atribut yang wujud dalam dataset. *Information gain* menerupakan *measure of reduction in disorder* dan nilai yang tinggi akan memastikan bahawa kedua-dua kelompok data sampel yang dipecah menggunakan nilai atribut tersebut mempunyai perbezaan yang tinggi dan sebaliknya. Nilai atribut yang paling tinggi *information gain* akan dipilih sebagai kriteria pemecahan pada nod tersebut. Proses ini akan berulang sampai data sampel tidak dapat lagi dipecahkan kepada dua kelompok yang berbeza.

Bagi mengira *information gain* bagi sesuatu nilai atribut, *entropy* data sampel dan *entropy* nilai atribut tersebut perlu dikira dahulu. *Entropy* bererti *measure of disorder* dan formula untuk pengiraan *entropy* ialah:

$$E(x) = \sum_{i=1}^{n} -p_i * log_2 * p_i$$

p_i merujuk kepada kebarangkalian elemen kelas *i* dalam dataset. Selepas mengira *entropy* bagi *x* (nilai bagi sesuatu atribut), *information gain* bagi *x* boleh dikira dengan formula berikut:

$$IG(x) = E(dataset) - E(dataset|x)$$

Formula ini bermaksud bahawa *information gain* merupakan penolakan *disorder dataset* daripada *disorder dataset given x*. Sekiranya *x* mampu mengurangkan *disorder dataset* dengan banyak seperti yang diwakili oleh $(dataset|x)$, maka *information gain* untuk *x* akan lebih tinggi dan *x* akan mempunyai lebih banyak peluang untuk dipilih sebagai kriteria pemecahan.

```
buildForest(dataset,number of trees):

    from 0 to number of trees:
        create a randomized dataset
        buildTree(randomized dataset)
        add tree to forest



buildTree(dataset):

   for every feature in dataset:
     for every value in feature:
         computeInformationGain(dataset,feature value) for the feature value


   for feature value with highest information gain:
      split dataset into sub-datasets using feature value


   if highest information gain is not 0:
      for all sub-datasets:
         buildTree(sub-datasets)

   else if highest information gain is 0:
      return leaf node



computeInformationGain(dataset,feature value):

    calculate global entropy by computeEntropy(dataset)

    split dataset into sub-datasets using feature value

    for all sub-datasets:
        calculate entropy for the sub-dataset by computeEntropy(sub-dataset)

    information gain = global entropy - information gain of all sub-datasets

    return information gain



computeEntropy(dataset):

    for all classes in the dataset:
        count the occurence of each classes

    for all class counts gathered:
        calculate and update the value of entropy using the class count

    return entropy
```

Rajah 4 Pseudokod pembinaan *Random Forest*

4.3.3 Reka Bentuk Antaramuka Pengguna

Reka bentuk prototaip bagi antara muka sistem juga telah dihasilkan. Sistem ini mengandungi laman *Home*, laman *Classify Species*, laman *Result*, laman *Model Settings*, laman *Create Model* dan juga laman *Manage Existing Models*. Reka bentuk prototaip bagi antara muka sistem projek ini yang direka menggunakan aplikasi *Mockplus*. Penambahbaikan terhadap reka bentuk prototaip akan dilaksanakan pada fasa yang seterusnya jika perlu.

4.4 Fasa Pengujian

Fasa ini bertujuan untuk menuji keteguhan sistem untuk memastikan sistem akhir yang dibangun tidak terdedah kepada kegagalan sistem. Kaedah pengujian berbeza bergantung kepada hasil projek. Hasil akhir bagi projek ini merupakan sistem pengelasan spesies riang-riang. Oleh itu, pengujian akan dilakukan terhadap modal klasifikasi spesies yang dibangun bagi menguji nilai prestasinya. Pengujian juga akan dilakukan terhadap sistem melalui antara muka sistem yang dibangun bagi menguji semua fungsian yang dibekalkan oleh sistem tersebut. Rajah-rajah di bawah menunjukkan pseudokod bagi menuji ketepatan modal klasifikasi yang dibangun. Keputusan pengujian terhadap modal dan antaramuka pengguna akan ditunjukkan pada hasil kajian.

```
# Import dataset from Excel & get feature list
ds = pd.read_excel('Dataset.xlsx')
feature_list = list(ds.drop('Label', axis=1).columns)

# Shuffle the dataset
ds_shuffled = ds.sample(frac=1).reset_index(drop=True)

# Get features,labels & convert to numpy array
labels = np.array(ds_shuffled['Label'])
features = ds_shuffled.drop('Label', axis=1)
features = np.array(features)

# Stratified 5-fold CV for Outer CV
skf = StratifiedKFold(n_splits=5)
cv_dataset = [(features[train_i], labels[train_i], features[test_i], labels[test_i])
              for train_i, test_i in skf.split(features,labels)]

# Hyperparameter Tuning
fold_count = 0
final_acc = 0

# 5-fold outer CV
for cur_fold in cv_dataset:
    fold_count += 1

    # Train & Test data for current fold
    feature_x = cur_fold[0]
    label_x = cur_fold[1]
    feature_y = cur_fold[2]
    label_y = cur_fold[3]

    # Estimate range for 'n_trees' & build param_grid
    # n_trees_grid = uf.getRange_nEstimator(feature_x,label_x)
    param_grid = {
        'n_estimators': [20, 40, 60, 80, 100],
        'max_features': np.arange(1, 2, 3),
        'max_depth': [20, 40, 60, 80, 100]
    }

    # Perform grid search in inner 5-fold CV
    rf = RandomForestClassifier()
    grid_search = GridSearchCV(estimator=rf, param_grid=param_grid, cv=5, n_jobs=-1, verbose=0)
    grid_search.fit(feature_x, label_x)

    # Get best model accuracy for current outer fold
    best_model = grid_search.best_estimator_
    predictions = best_model.predict(feature_y)
    results = predictions == label_y
    correct = np.delete(results, np.where(results == False))
    cur_acc = len(correct) / len(results)
    print('Fold: {} Accuracy: {:.4f}'.format(fold_count,cur_acc))
    final_acc += cur_acc

final_acc /= 5
print('\nFinal Accuracy:',final_acc)
```

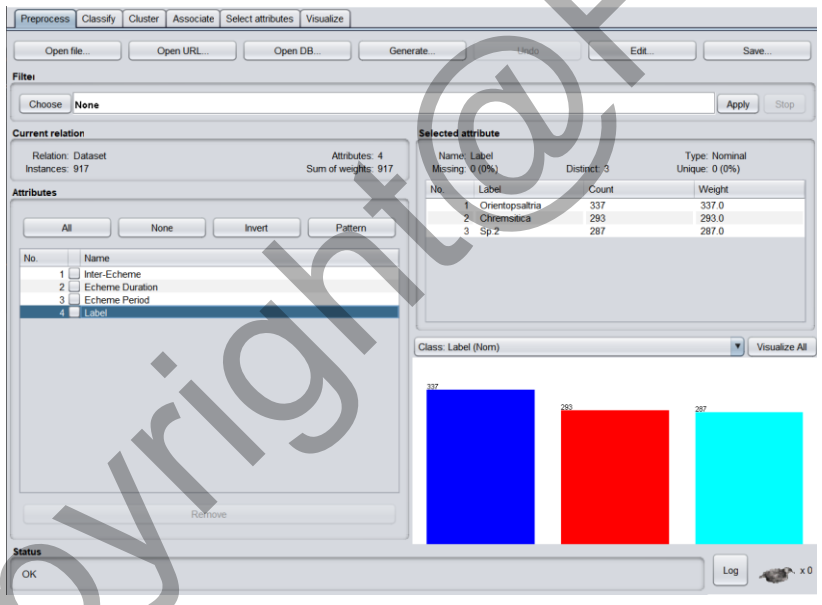
Rajah 5 Kod untuk menilai prestasi modal klasifikasi

5 HASIL KAJIAN

Bahagian ini akan membincangkan semua keputusan yang diperoleh sepanjang proses pembinaan sistem pengelasan spesies. Proses utama bagi membina sistem adalah mengenal pasti kaedah klasifikasi yang sesuai melalui perbandingan antara kaedah-kaedah klasifikasi. Selepas kaedah klasifikasi yang sesuai telah ditentukan, langkah pembangunan dan pengujian sistem perlu dijalankan.

5.1 Perbandingan kaedah klasifikasi

Rajah di bawah menunjukkan dataset yang digunakan bagi proses perbandingan. Dataset tersebut mengandungi tiga ciri suara panggilan – *Echeme Duration*, *Echeme Period* dan *Inter-Echeme Duration*.

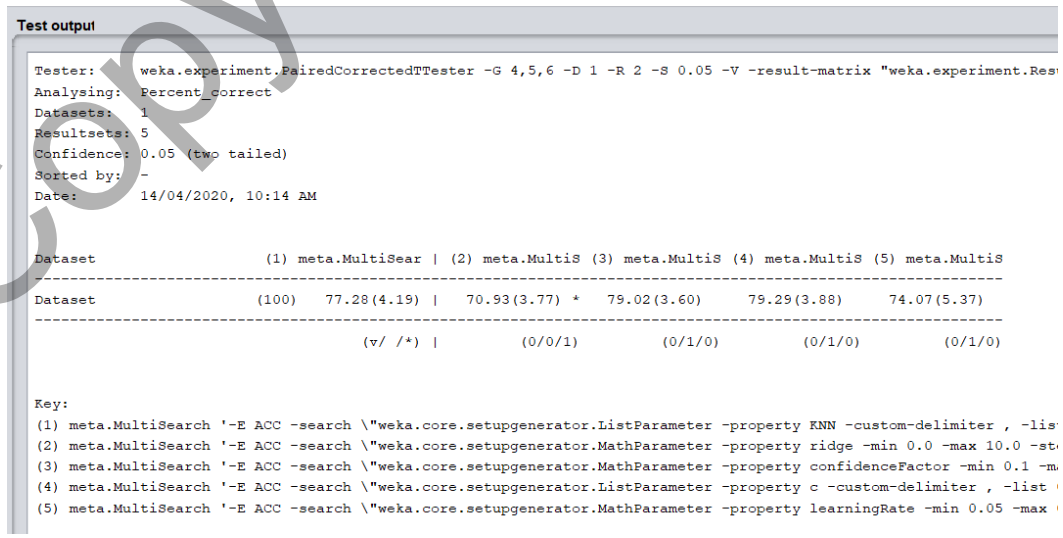


Rajah 6 Info dataset

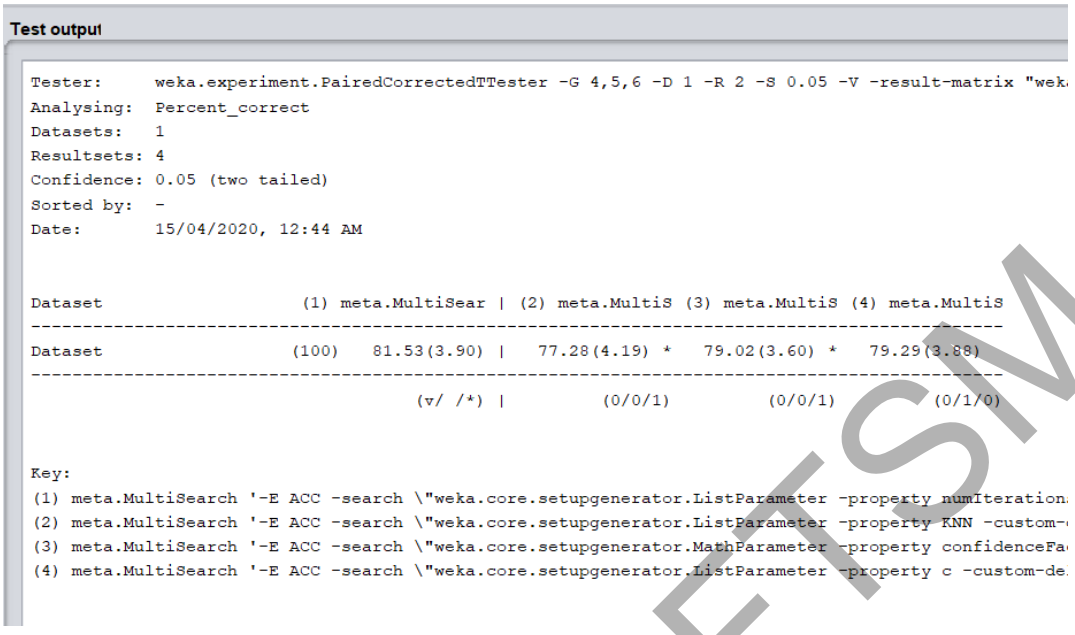
Hiperparameter merupakan parameter yang nilainya ditetapkan sebelum proses pembelajaran bagi modal klasifikasi bermula. Nilai optimum bagi hiperparameter tidak dapat ditentukan melalui proses pembelajaran menggunakan *training data* dan hanya dapat dioptimumkan menggunakan kaedah percubaan dan kesilapan. Walaupun proses pengoptimuman hiperparameter agak membebankan, ia memainkan peranan penting dalam mempengaruhi prestasi modal dan tidak boleh diabaikan supaya keputusan perbandingan yang diperolehi lebih menyakinkan. Rajah di bawah menunjukkan semua hiperparameter yang dioptimumkan bagi setiap kaedah klasifikasi bagi projek ini. Rajah yang seterusnya menunjukkan keputusan bagi kedua-dua fasa perbandingan yang dijalankan.

- **K-Nearest Neighbour**
  k: [1,2,3,4,5,10,15,20,25,30,35,40,45,50]
- **Logistic Regression**
  ridge: [0,1,2,3,4,5,6,7,8,9,10]
- **Decision Tree**
  confidenceFactor: [0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9]
  minNumObj: [1,5,10,15,20,25,30,35,40,45,50]
  unpruned: [True,False]
- **Support Vector Machines (SVM)**
  c: [0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9]
  kernel: [PolyKernel, RBFKernel]
  gamma: [0,0.01,0.1,1,10,100]
- **Neural Network**
  trainingTime: [100,200,300,400,500,600,700,800,900,1000]
  learningRate: [0.05,0.1,0.15,0.2,0.25,0.3,0.35,0.4,0.45,0.5]
  hiddenLayers: [1,2,3,4,5,6,7,8,9,10]
- **Random Forest**
  numIterations: [10,20,30,40,50,60,70,80,90,100]
  numFeatures: [1,2,3]
  maxDepth: [10,20,30,40,50,60,70,80,90,100]

Rajah 7 *Hyperparameter* kaedah-kaedah klasifikasi yang dibanding



Rajah 8 Keputusan Perbandingan (1)



Rajah 9 Keputusan perbandingan (2)

Berdasarkan rajah 8, kaedah *Support Vector Machines* (SVM) mempunyai prestasi yang paling tinggi dengan ketepatan 79.29%. Namun, ketepatan *Decision Tree* juga amat dekat dengan 79.02%. Oleh itu, ini mencadangkan bahawa kaedah *Random* Forest yang merupakan modal ensembel *Decision Tree* berupaya untuk mencapai nilai prestasi yang lebih tinggi dan mengalahkan kaedah *Support Vector Machines*. Keputusan perbandingan di rajah 9 menunjukkan bahawa kaedah *Random Forest* memang berjaya memberikan nilai prestasi yang lebih tinggi dan berjaya mengalahkan semua kaedah klasifikasi yang lain. Oleh itu, kaedah *Random Forest* dipilih sebagai kaedah klasifikasi yang akan digunakan bagi projek ini untuk mengelas spesies riang-riang. Reka bentuk algoritma juga akan dihasilkan berdasarkan kaedah tersebut.

5.2 Pembangunan modal klasifikasi

Bahagian ini membincangkan kaedah untuk membangun modal klasifikasi *random forest*. Modal *random forest* untuk projek ini dibangun menggunakan bahasa pengaturcaraan *Python* dengan bantuan modul mesin pembelajaran seperti *pandas* dan *sklearn*. Rajah di bawah menunjukkan kod untuk membangun modal klasifikasi *random forest* yang akan digunakan oleh sistem.

Modal *random forest* dibina melalui fungsi *buildForest()*. Dalam fungsi tersebut, modal asas *random forest* dibina melalui *RandomForestClassifier()* dari modul *sklearn*. Selepas itu, *parameter grid* yang mengandungi semua kombinasi nilai hiperparameter akan

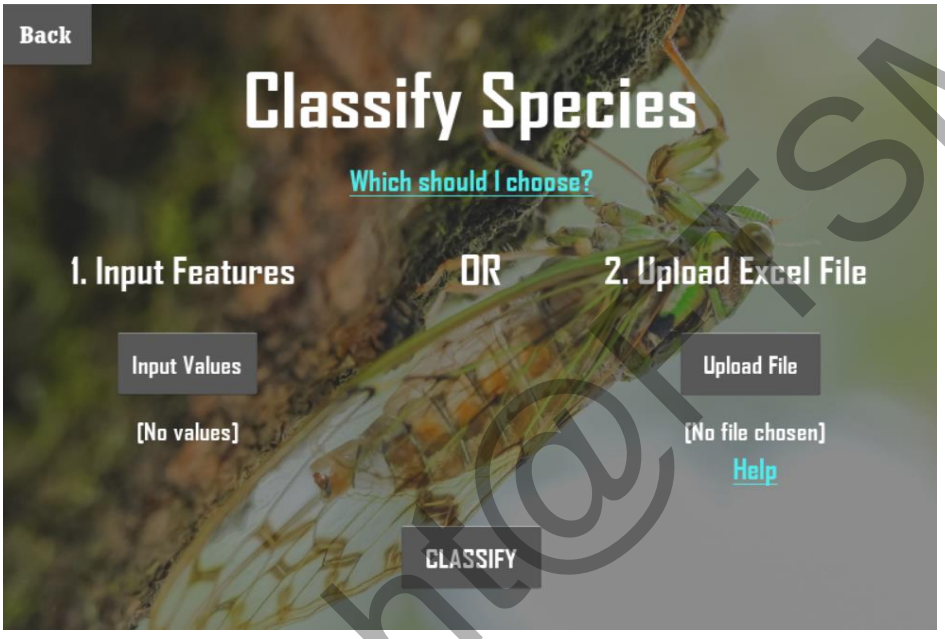
dibina. Pengoptimuman dan *training* modal akan dilaksanakan kemudian untuk mencari modal *random forest* dengan nilai hiperparameter yang optimum menggunakan pengesahan bersilang 10-lipatan. Dalam fungsi *main()*, dataset akan dipecahkan kepada 10 lipat untuk menjalankan pengesahan bersilang 10-lipatan. Bagi setiap lipat data, fungsi *buildForest()* akan dipanggil untuk membina modal *random forest* yang sekali lagi menggunakan pengesahan bersilang 10-lipatan untuk mencari nilai hiperparameter yang optimum. Proses ini dikenali sebagai *nested cross validation*. Oleh itu, sebanyak 10 modal *random forest* dengan nilai hiperparameter yang optimum akan dibina dan dinilai pada pengesahan bersilang luaran untuk mendapatkan ketepatan klasifikasi bagi setiap modal dan anggaran ketepatan akhir boleh dikira menggunakan purata ketepatan klasifikasi 10 modal tersebut. Akhirnya, satu modal *random forest* akan dibina menggunakan fungsi *buildTree()* dengan menggunakan semua data dalam dataset.

```
import pandas as pd
from sklearn.model_selection import StratifiedKFold
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV


buildForest(train_features, train_labels):
    # Create a base random forest model
    rf = RandomForestClassifier()

    # Create the parameter grid
    param_grid = {
        'max_depth': [80, 90, 100, 110],
        'max_features': [2, 3],
        'min_samples_leaf': [3, 4, 5],
        'min_samples_split': [8, 10, 12],
        'n_estimators': [100, 200, 300, 1000]
    }

    # Instantiate the grid search model
    grid_model = GridSearchCV(estimator = rf, param_grid = param_grid,
                              cv = 10, n_jobs = -1, verbose = 2)

    # Train the grid search model
    grid_model.fit(train_features, train_labels)



main():
    # Load data and split into stratified 10-folds
    load dataset using pd.read_excel(file_path)
    create 2 dataFrame object to contain dataset features(X) & dataset targets(y)
    Perform 10-fold stratification by StratifiedKFold(n_splits=10).split(X,y)

    averaged_acc = 0

    # Evaluate accuracy of tuned model for each fold
    FOR each fold of stratified data:
        get train_features, train_labels, test_feature, test_labels
        rf_model = buildForest(train_features ,train_labels)

        current_acc = evaluate(rf_model, test_feature, test_labels)
        averaged_acc += current_acc

    # Final estimated accuracy of the tuned random forest model
    averaged_acc /= 10

    # Build the final model with all the data after nested cross validation
    final_rf_model = buildTree(all_train_features, all_train_labels)
```

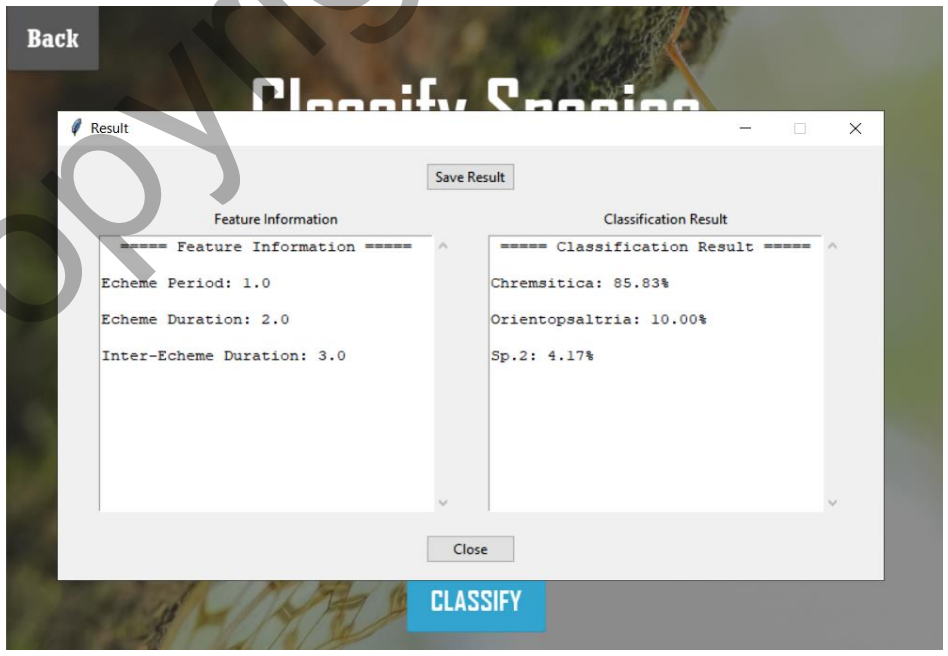
Rajah 10 Kod untuk membangun modal klasifikasi *random forest*

5.3 Pembangunan antaramuka pengguna (UI)

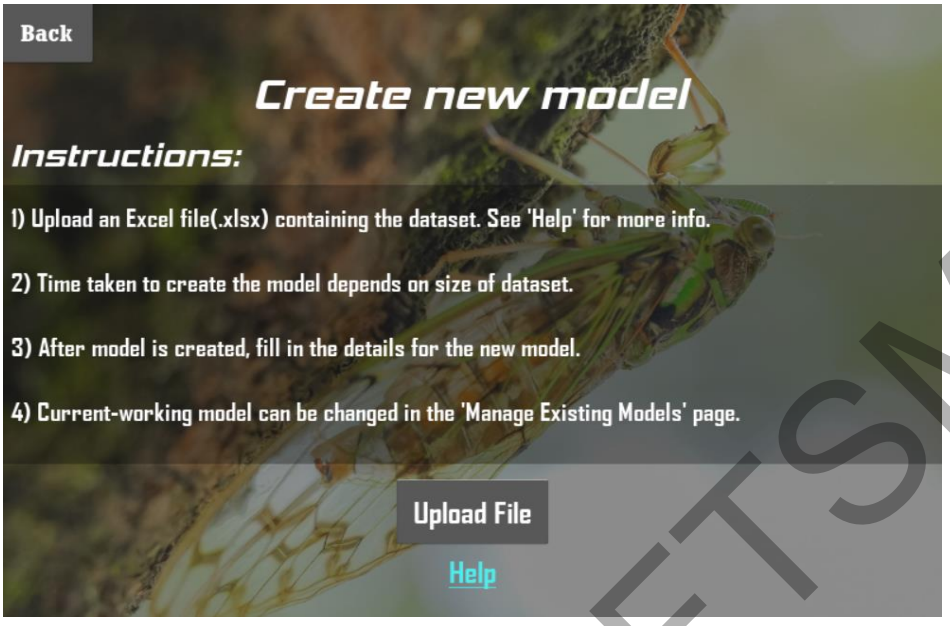
Antaramuka pengguna bagi sistem ini dibangun menggunakan *Kivy*, satu kerangka *Python* untuk pembangunan aplikasi. Beberapa perubahan telah diperkenalkan terhadap prototaip reka bentuk sebelum ini. Rajah di bawah menunjukkan beberapa laman penting bagi sistem ini dan fungsian yang dibekalkan pada laman tersebut.



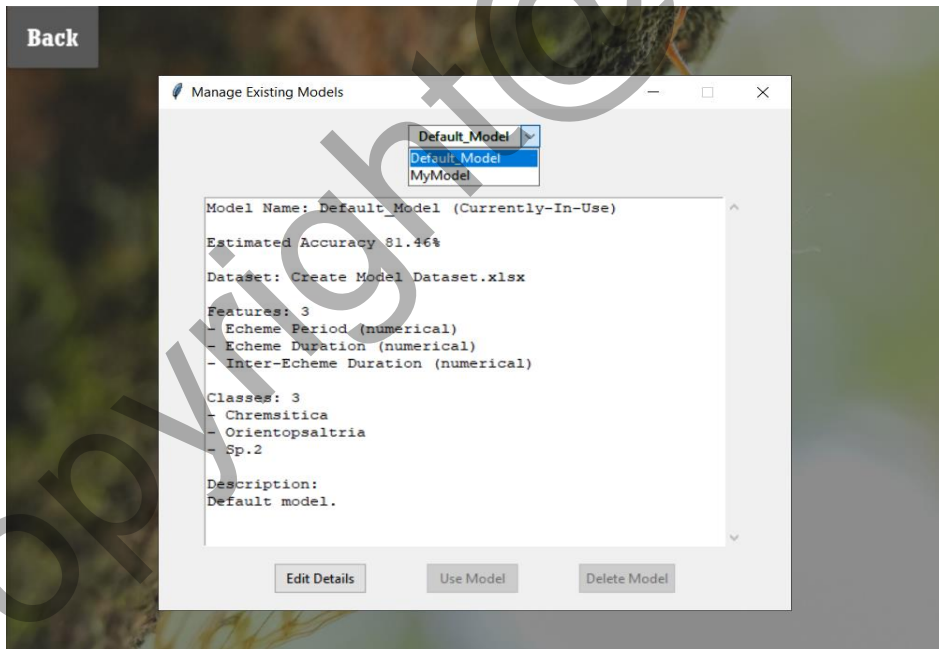
Rajah 11 Laman *Classify Species*



Rajah 12 Laman *Result*



Rajah 13 Laman *Create Model*



Rajah 14 Laman *Manage Existing Models*

Laman *Classify Species* memberikan dua kaedah bagi menjalankan pengelasan spesies riang-riang. Kaedah pertama akan melakukan klasifikasi terhadap satu data dengan mudah dan cepat. Selepas habis proses klasifikasi, sistem akan menunjukkan laman *Result* di mana keputusan klasifikasi akan dipaparkan kepada pengguna. Kaedah kedua pula membolehkan

klasifikasi terhadap banyak data dan memerlukan pengguna untuk memuat naik fail *Excel*(.xlsx). Satu fail *Excel* akan dijana bagi mengandungi semua keputusan klasifikasi dan disimpan dalam lokasi direktori yang dipilih oleh pengguna.

Laman *Create Model* membolehkan pengguna untuk membina modal klasfikasi baru. Pengguna perlu memuat naik fail Excel(.xlsx) yang mengandungi dataset yang akan digunakan untuk membina modal klasifikasi. Selepas modal klasifikasi siap dibina, pengguna akan diminta untuk mengisi butiran bagi modal tersebut dan fail modal akan disimpan dalam direktori *Models*.

Laman *Manage Existing Models* pula membolehkan pengguna untuk menjalankan pelbagai operasi terhadap modal klasifikasi sedia ada seperti megubah butiran modal, mengubah modal klasifikasi yang dipakai oleh sistem dan memadam modal lama. Jika nama modal yang diubah bertindih dengan nama modal lain, *popup error* akan ditunjuk. Modal yang sedang dipakai oleh sistem tidak boleh dipadam oleh pengguna.

5.4 Pengujian sistem dan maklum balas pengguna

Pengujian telah dijalankan terhadap modal klasifikasi yang dibangun dan antaramuka pengguna bagi menguji kekukuhan sistem serta memadam pepijat dari sistem. Rajah-rajah di bawah menunjukkan keputusan pengujian.

```
Fold: 1 Accuracy: 0.8152
Fold: 2 Accuracy: 0.8152
Fold: 3 Accuracy: 0.8087
Fold: 4 Accuracy: 0.8361
Fold: 5 Accuracy: 0.8087

Final Accuracy: 0.8167973390354003
```

Rajah 15 Keputusan penilaian prestasi modal klasifikasi

LAMAN	UJIAN	KEPUTUSAN
Home	Navigasi ke laman *Classify*	LULUS
	Navigasi ke laman *Model Settings*	LULUS
Classify	Tunjukkan *Help popup* jika hiperteks *Which should I choose?* ditekan	LULUS
	Muat naik fail *Excel* melalui butang *Upload File*	LULUS
	Tunjukkan *Error popup* sekiranya format fail yang dimuat naik tidak sah	LULUS
	Klasifikasi menggunakan kaedah 1 (*Input Features*)	LULUS
	Klasifikasi menggunakan kaedah 2 (*Upload File*)	LULUS
Result	Menyimpan keputusan dalam fail *Excel* melalui butang *Save Result*	LULUS
Model Settings	Navigasi ke laman *Create Model*	LULUS
	Navigasi ke laman *Manage Existing Models*	LULUS
Create Model	Membina modal klasifikasi selepas fail *Excel* dimuat naik	LULUS
	Tunjukkan *Error popup* sekiranya format fail yang dimuat naik tidak sah	LULUS
Manage Existing Models	Edit butiran modal klasifikasi	LULUS
	Menunjukkan *error popup* jika nama modal klasifikasi bertindih	LULUS
	Mengubah modal klasifikasi yang dipakai oleh sistem	LULUS
	Memadam modal klasifikasi	LULUS

Rajah 16 Keputusan ujian antaramuka pengguna (UI)

Rajah 15 menunjukkan prestasi modal random forest yang dibangun menggunakan modul *sklearn* dan diuji menggunakan kod yang ditunjukkan di rajah 5. Prestasi modal tersebut adalah hampir sama dengan seperti yang ditunjukkan di perisian WEKA pada fasa perbandingan kaedah klasifikasi (~81.50%). Ini menunjukkan modal yang dibangun betul dan berfungsi seperti yang dijangka. Rajah 16 pula menerangkan keputusan pengujian bagi setiap fungsian sistem melalui anataramuka pengguna sistem. Pengujian ini menunjukkan bahawa sistem yang dibangun adalah kukuh dan bebas ralat.

	Easy to Look					Easy to Understand					Easy to Use				
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
Main Screen					X					X					X
Comment: Very good															
Classify Screen					X					X					X
Comment: Very good															
Result Window					X					X					X
Comment: Very Good															
Settings Screen					X					X					X
Comment: Very Good															
Create Screen					X					X					X
Comment: Very Good															
Manage Model Window					X					X					X
Comment: Very Good															

Rating Scale:

1 - Very Bad 4 - Good

2 - Bad 5 - Very Good

3 - Neutral

15June2020

Dr. Johari Bin Jalinas
Senior Lecturer
Faculty of Science & Technology
Universiti Kebangsaan Malaysia

Rajah 17 Maklum balas pengguna

Maklum balas yang diberi oleh Dr. Johari adalah sangat positif. Beliau mengatakan bahawa sistem tersebut mampu mencapai objektif utama iaitu memaparkan kebarangkalian kepunyaan spesies diberikan ciri-ciri suara panggilan. Selain itu, kebebasan untuk membina modal klasifikasi baru amat berguna dan memanfaatkan. Ini merupakan sebab maklum balas yang positif dan baik telah diberikan. Namun, beliau berharap bahawa sistem tersebut boleh ditukar ke aplikasi mudah alih pada masa yang akan datang.

6 KESIMPULAN

Projek ini telah menghasilkan satu sistem cerdas, *Cicada Master* yang bertujuan untuk mengautomatikkan proses pengelasan spesies riang-riang. Objektif bagi kajian ini telah dicapai dan masalah pengelasan spesies secara manual yang membebankan telah diatasi walaupun wujudnya pelbagai kekangan projek. Penambahbaikan projek iaitu menyari ciri-ciri suara panggilan melalui sistem juga dicadangkan sekiranya usaha mampu dilaksanakan pada masa depan.

7 RUJUKAN

Agarwal, A. 2019. Software engineering | structure charts. URL https://www. geeksforgeeks.org/software-engineering-structure-charts/.

Athuraliya, A. 2017. Sequence diagram tutorial: Complete guide with examples. URL https://creately.com/blog/diagrams/sequence-diagram-tutorial/.

Eriksson, U. 2012. Functional vs non functional requirements. URL https://reqtest.com/ requirements-blog/functional-vs-non-functional-requirements/.

Fonseca, P.J. & Revez, M.A. 2002. Song discrimination by male cicadas cicadabarbara lusitanica (homoptera, cicadidae). *Journal of Experimental Biology* 205(9): 1285–1292. URL https://jeb.biologists.org/content/205/9/1285.

Guru99.com. 2019. Prototyping model in software engineering: Methodology, process, approach. URL https://www.guru99.com/ software-engineering-prototyping-model.htmlhttps://www.guru99.com/ software-engineering-prototyping-model.html.

Inflectra. 2018. What are system requirements specifications/software (srs)? URL https://www.inflectra.com/ideas/topic/requirements-definition.aspx.

InterServer. 2019. What is mvc? advantages and disadvantages of mvc. URL https: //www.interserver.net/tips/kb/mvc-advantages-disadvantages-mvc/.

Mogzai, D. 2019. What are cicadas? URL https://www.cicadamania.com/.

ProwebUltimaERP. 2019. Pengertian data dictionary. URL https://www.ultima-erp.id/ article/sia/data-dictionary/.

Visual-Paradigm. 2019. What is use case diagram? URL https://www.visual-paradigm. com/guide/uml-unified-modeling-language/what-is-use-case-diagram/.

VisualParadigm. 2019a. What is class diagram? URL https://www.visual-paradigm. com/guide/uml-unified-modeling-language/what-is-class-diagram/.

VisualParadigm. 2019b. What is data flow diagram? URL https://www.visual-paradigm.com/guide/data-flow-diagram/what-is-data-flow-diagram/.

Yiu, T. 2019. Understanding random forest. URL https://towardsdatascience.com/understanding-random-forest-58381e0602d2.