Academia of Information Computing Research
Vol. 2, No. 1, pp. 1-8
ISSN: 2716-6465

COMPARATIVE ANALYSIS OF CLASSIFICATION BASED ON CELLULAR LOCALIZATION DATA USING MACHINE LEARNING

Rohayanti Hassan[1], Muhammad Luqman Mohd Shafie[1], Alif Ridzuan Khairuddin[1]
[1]*Faculty of Computing, Universiti Teknologi Malaysia, Johor, Malaysia*
*E-mail: rohayanti@utm.my, muhammadluqman.ms@utm.my, alifridzuan@utm.my

***Abstract*— Due to the pandemic caused by Covid-19, vaccine development has been a hot issue to be discussed and a lot of research was conducted to create a vaccine that is efficient in fighting against viral infection. Therefore, protein subcellular localization is one of the methods that are suitable to be used in studies of vaccine development. By recent technology, the protein subcellular localization is only able to handle single compartment prediction but in reality, there are multiple compartment predictions that need to be done in order to give an accurate prediction. Previously, we used DM3Loc pre-existing tools that were used to generate subcellular localization data from the FASTA Sequences to get the concentration of the viral protein inside the cell. Based on the result, we can conclude that the selected protein is highly possible to reside within the cells. For DM3Loc, we use CNN which is a Convolutional Neural Network as a framework. But what if we try to reverse-engineer the tools by using another machine learning model such as Decision Tree, Random Forest or Support Vector Machine? Is it still able to produce accurate prediction results? The dataset that will be used in this research was obtained from an online database and ran through DM3Loc to obtain the Subcellular Localization Dataset. Based on the findings, other machine learning methods can probably be another option than CNN for the future of subcellular localization.**

Keywords*— *Decision Tree, Random Forest, Support Vector Machine, DM3Loc, Subcellular Localization machine learning.

I. INTRODUCTION

Protein subcellular localization prediction involves the prediction of the location of protein situated inside a cell. Generally speaking, prediction tools take input information such as a fasta file of mRNA sequence of the protein and generate predicted location within the cell as output, such as the nucleus, endoplasmic reticulum, Golgi apparatus, extracellular space, or other organelles. The aim is to accurately predict the location of the protein of interest inside the cells.

Currently, we are still using the Single-Headed Self Attention Mechanism that still has some weaknesses such as only being able to be localized in a single compartment while in reality mRNA is located at multiple sites or locations in the cell. Therefore, it contributes to meaningless biological interpretation and reduced prediction power compared to the non-attention method which may be due to some drawbacks in the interpretation power caused by the weighted sum of hidden states derived from previous layers with its single attention-weight vector.

By using Machine Learning algorithms which are Decision Tree, Random Forest and Support Vector Machine, we attempted to reverse-engineer the process of predicting the type of virus by using a cellular localization dataset produced from the Self Attention Method to justify whether other machine learning can predict the type of virus from cellular location and that variable are credible enough to be used for prediction adder.

When it comes to problems such as differentiating between DNA sequences and classification of DNA sequences, the Machine Learning algorithm is a good choice (Srinivasa et al., 2020b). Machine Learning algorithms are commonly used in biological data classification to make predictions. Machine Learning is also used for clustering genes and the reference genome. Machine Learning becomes more essential in solving biological problems which is aided by rapid incline on biological data which is Big Data. There are also difficulties in translating raw data into biological knowledge. For this research, the proposed machine learning algorithms for solving Subcellular Localization Datasets to classify viral species include three methods: Decision Tree, Random Forest and Support Vector Machine. The details for each method will be described in the following subsection.

According to Wolpert and Macready (1997), the performance of any proposed algorithm over one set of issues is compensated if the performance over another set of problems is improved. To put it in another way, if an optimization strategy performs effectively in a certain situation, it is considered successful. It is possible that it will not be as effective in solving other difficulties. Recently, new approaches are expected to fill such a gap. To solve various biology problems, machine-learning approaches are frequently used. Recently, three machine-learning-based methods have been discussed to predict various viral species based on subcellular locations of which demonstrated that machine-learning methods have become experimental techniques to detect viral species as shown in Table 1. The method includes Decision Tree, Random Forest and Support Vector Machine.

Table 1 Comparison of Machine Learning Methods

Advantages	**DT**	**RF**	**SVM**
Computationally faster (Navlani, 2018)	/		/
Less Memory (Navlani, 2018)	/		/
Higher Accuracy (Penumudy, 2021)		/	
Low Overfitting (Penumudy, 2021)		/	
Suitable for Multi-Class Classification (This study)	/	/	

Disadvantages	**DT**	**RF**	**SVM**
Slow Prediction (Penumudy, 2021)		/	
Less Variation (Navlani, 2018)	/		
Sensitive to noise data (Navlani, 2018)	/		
Low Accuracy on Multi-Class Classification (This study)			/

II. METHODS

This section briefly describes the research framework, dataset, and performance measurement.

A. Research Framework

There were four phases in our research framework in which research planning was the first phase. The second phase was data preparation. The third one was Algorithm Development. The last one was Testing and Evaluation phase. For the classification technique, performance measurement was used to measure and compare both models.

B. Performance Measurement

Performance methods that were used to measure and evaluate the performance of the model during the classification of the dataset will be explained.

i. Average Area Under Precision-Recall (PR)

In the information retrieval process, precision is the measure of the relevancy of the results meanwhile recall is to measure how many of the relevant results are returned. The tradeoff was between precision and recall in PR Curve for different thresholds. The formulae that are used for calculating the precision and recall are shown as below:

a) Precision is calculated as the True Positive (TP) divided by the sum of True Positive and False Positive.

$$P = \frac{T_p}{T_p+F_p} \quad (1)$$

b) Recall is calculated as the True Positive (TP) divided by the sum of True Positive and False Negative.

$$P = \frac{T_p}{T_p+F_p} \quad (2)$$

c) The F1 score is given as follows:

$$P = \frac{T_p}{T_p + F_p} \qquad (3)$$

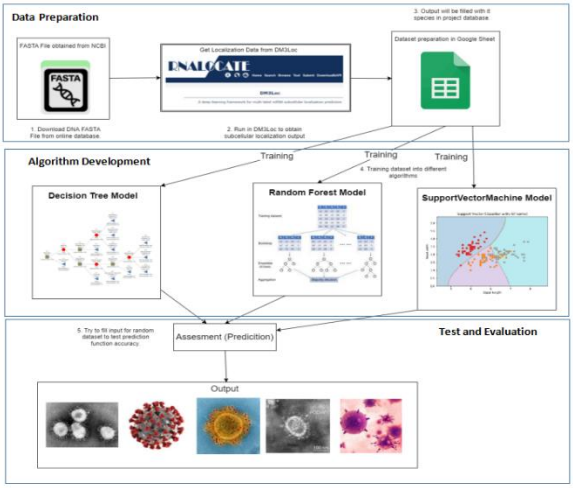

Figure 1: Experimental Workflow

ii. Matthews Correlation Coefficient (MCC)

Matthews Correlation Coefficient is used to measure the quality of binary classification in Machine Learning. The coefficient refers to True, False Positive and Negative that is generally regarded as balanced measures that are used in every class with different sizes. MCC can be calculated based on the Confusion Matrix and calculated by using the previously mentioned formulae.

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \qquad (4)$$

III. RESULTS AND DISCUSSION

The experimental design began with downloading and preparing FASTA File from any online database. The next step was to run all of the FASTA files into DM3Loc which is the Multihead Self Attention Method. This step was conducted to obtain subcellular localization output for training into a machine learning model. After that the output was filled manually into a csv file in Google Sheet until 500 sequence samples for training. Then, the datasets were run into both models of machine learning. The testing and evaluation process consisted of prediction where we filled any number into input and we predicted what viral species will be predicted for output. All methodology and coding used Python language and ran through Google Colaboratory which is an online available IDE.

A. Analysis on Machine Learning Models

1) Data Preparation

These datasets were created by us after we finished running them via DM3Loc. The output obtained during that process were filled in the Google Sheets until there were enough for 500 samples with 5 different species. The amount for each species may vary depending on the available sequences online. These five species which are SARS Cov2, MERS, Porcine Cov, Human coronavirus NL63 and 229E, were chosen due to them being coronavirus types, as shown in Figure 2.

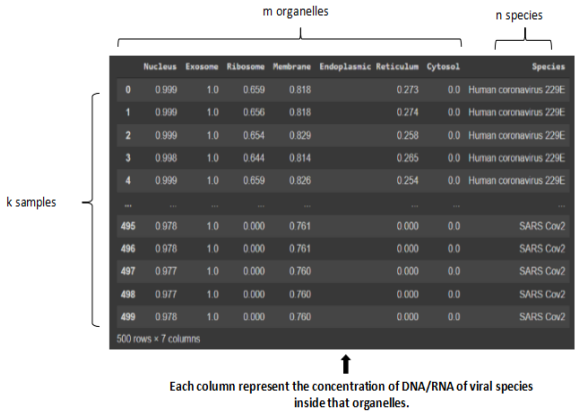

Figure 2: Sample Dataset Structure

2) Data Pre-processing

All the pre-processing and data cleaning processes were conducted during the input filling datasets. Therefore, there was no need to code for data cleaning and so on. To prove our claim, we included the code to see if there are any missing values in the datasets that can affect our model.

3) Algorithm Development

There are three machine learning models that were used for this classification which are Decision Tree, Random Forest and Support Vector Machine, which were implemented to evaluate viral species. More details will be provided in the next section.

4) Decision Tree

Decision Tree works like a tree structure where there are internal nodes that will denote a test on attributes and each branch represents the result of the test. The box represents the checkpoint for the computer to make a decision based on the input they obtained at that time.

5) Random Forest

Random Forest works similarly like a poll or election from a number of decision trees where it is called a sample. Then, all samples will produce predicted results from each decision tree. Next, from the result obtained from each sample, the majority vote will be decided as the final outcome or final prediction.

6) Support Vector Machine

Support Vector Machine works by creating a line or hyper-plane that will differentiate data into separate classes. For this research, the prediction could only be done by choosing two from six inputs available. The prediction is done by clustering the data based on coordinates on the plane and grouping it by a specific group. For example, if the sample is located near to each other in the plane and also grouped across the plane, therefore there is a higher possibility they are a similar species. As a result, the Support Vector Machine method of prediction is hugely different from previous algorithms which are Decision Tree and Random Forest.

A. Evaluation on Machine Learning Model

This section focuses on the results and analysis of the model performance of Decision Tree, Random Forest and Support Vector Machine in classifying viral species in terms of accuracy, sensitivity, and precision.

1) Material

Before training the model, the first step was to declare and partition the dataset into training and testing datasets. Separation of the dataset are divided into 80 percent training and 20 percent testing, with actual 401 and 101 samples respectively. Data partitioning was implemented to train the model with several data (testing dataset). The models were tested with testing data to predict the class based on the training dataset. Table 5.1 below shows the data partitioning of the subcellular localization datasets for both models. After the dataset was partitioned into the decided amount, the models were trained and tested to obtain the prediction result.

Table 2 Data Partitioning

Dataset	**Data Partition**		**Total**
	Training (80%)	**Testing (20%)**	
Number of Sample	401	101	502

2) Model Performance

To facilitate the observation of the model accuracy, recall, precision, F1-score, MCC and Confusion Matrix results were obtained by using formal definition as mentioned in the previous subsection, the model of confusion matrix based on testing dataset.

3) Decision Tree

The Decision Tree in Figure 3 represents five different types of viral species according to DNA concentration inside the organelles obtained from subcellular localization data. The instance will be sorted down to the bottom of the decision tree and eventually classified as Human NL3. To facilitate the observation of the model accuracy, recall, precision, F1-score, MCC and Confusion Matrix results were obtained by using formal definition as mentioned in the previous subsection, the model of confusion matrix based on testing dataset.

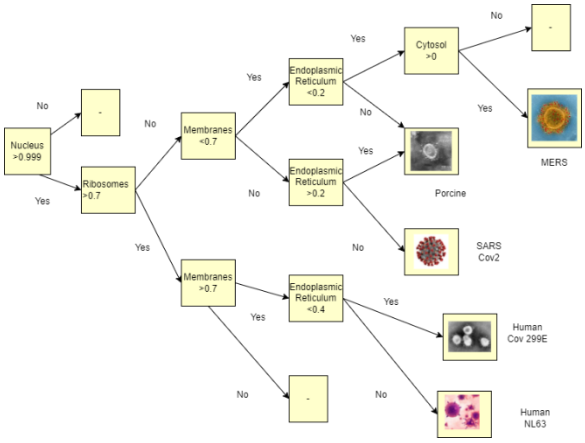

Figure 3: Decision Tree Architecture

```
In [55]: #From Random Dataset
         model.predict([[0.962,1,0,0.645,0.239,0]])
         #The input accurately predicted for >MH025764.1 Porcine deltacoronavirus strain CH/JXJGS01/P50, complete genome

Out[55]: array(['Porcine Cov'], dtype=object)

In [56]: #From Random Dataset
         model.predict([[0.978,1,0,0.761,0,0]])
         #The input accurately predicted for >NC_045512.2 Severe acute respiratory syndrome coronavirus 2 isolate Wuhan-Hu-1,

Out[56]: array(['SARS Cov2'], dtype=object)

In [57]: #From Random Dataset
         model.predict([[0.993,1,0.396,0.754,0.233,0]])
         #The input accurately predicted for >KF745068.1 Middle East respiratory syndrome coronavirus isolate FRA/UAE, complet

Out[57]: array(['MERS'], dtype=object)

In [58]: #From Random Dataset
         model.predict([[0.999,1,0.737,0.835,0.515,0]])

         #The input accurately predicted for >LC687394.1 Human coronavirus NL63 Fukushima_H257_2018 RNA, complete genome

Out[58]: array(['Human coronavirus NL63'], dtype=object)
```

Figure 4: Decision Tree Prediction Result

```
In [64]: print(classification_report(y_test,predictions))
```

	precision	recall	f1-score	support
Human coronavirus 229E	0.80	1.00	0.89	8
Human coronavirus NL63	0.88	0.78	0.82	18
MERS	0.91	0.95	0.93	22
Porcine Cov	1.00	1.00	1.00	23
SARS Cov2	1.00	0.97	0.98	30
accuracy			0.94	101
macro avg	0.92	0.94	0.93	101
weighted avg	0.94	0.94	0.94	101

Figure 5: Decision Tree Classification Report

4) Random Forest

Random Forest is an example of how our prediction system works. After training the data, they were tested on separate Decision Tree models with different rows of training and testing for each decision tree. After the prediction result was produced, it went through the voting process first. As shown, Covid-19 had the majority compared to MERS with 2 votes, therefore Covid-19 was the final prediction. This contributed to accuracy, robustness and credibility due to the participation of more decision trees for the final result.

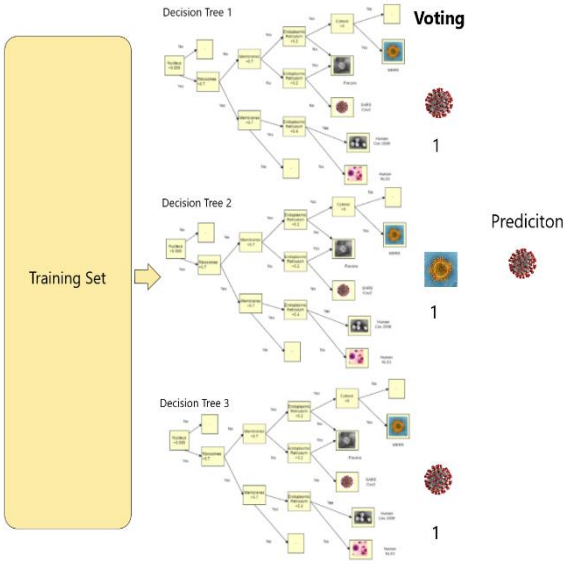

Figure 6: Random Forest Architecture

```
In [91]: #From Random Dataset
         model2.predict([[0.962,1,0,0.645,0.239,0]])
         #The input accurately predicted for >HH025764.1 Porcine deltacoronavirus strain CH/JXJGS01/P50, complete genome

Out[91]: array(['Porcine Cov'], dtype=object)

In [92]: #From Random Dataset
         model2.predict([[0.978,1,0,0.761,0,0]])
         #The input accurately predicted for >NC_045512.2 Severe acute respiratory syndrome coronavirus 2 isolate Wuhan-Hu-1,

Out[92]: array(['SARS Cov2'], dtype=object)

In [93]: #From Random Dataset
         model2.predict([[0.993,1,0.396,0.754,0.233,0]])
         #The input accurately predicted for >KF745068.1 Middle East respiratory syndrome coronavirus isolate FRA/UAE, complet

Out[93]: array(['MERS'], dtype=object)

In [94]: #From Random Dataset
         model2.predict([[0.999,1,0.737,0.835,0.515,0]])

         #The input inaccurately predicted for >LC687394.1 Human coronavirus NL63 Fukushima_H257_2018 RNA, complete genome

Out[94]: array(['Human coronavirus NL63'], dtype=object)
```

Figure 7: Random Forest Prediction Result

```
[263] print(cr2)

                          precision    recall  f1-score   support

    Human coronavirus 229E       0.83      1.00      0.91        10
    Human coronavirus NL63       0.93      0.81      0.87        16
                      MERS       0.95      1.00      0.98        21
               Porcine Cov       0.95      0.90      0.93        21
                 SARS Cov2       0.88      0.88      0.88        33

                  accuracy                           0.91       101
                 macro avg       0.91      0.92      0.91       101
              weighted avg       0.91      0.91      0.91       101
```

Figure 8: Random Forest Classification Report

5) Support Vector Machine

The problem with SVM is that it is hard for SVM to handle classification for multiclass that have more than 2 feature data for classification prediction. One vs All (OVA) is usually used for SVM for Multiclass Classification, where a number of binary classification needs to be conducted to classify a number of species. Therefore, to increase the accuracy of SVM, Crammer & Singer method for Multiclass Classification was used to improve the accuracy result for SVM. This method helps to reduce memory and training time.

Random Forest outperformed Decision Tree and Support Vector Machine with 95 percent. In terms of MCC, Random Forest also outperformed Decision Tree and Support Vector Machine with 93.6 percent. This result was an expected result due to Random Forest being one of the most powerful methods. Based on our observation of the final results, the reason for Support Vector Machine accuracy being lower compared to other algorithms is due to several factors.

- SVM is not suitable for large datasets.
- SVM performs poorly in imbalance data.

For this research,we used categorical data to classify viral species. Support Vector Machine is mostly used to solve the linear and graphical problems. Even though the Decision Tree was the second highest in accuracy and MCC score, it is also not recommended to be used for prediction to avoid bias due to less variation of data.

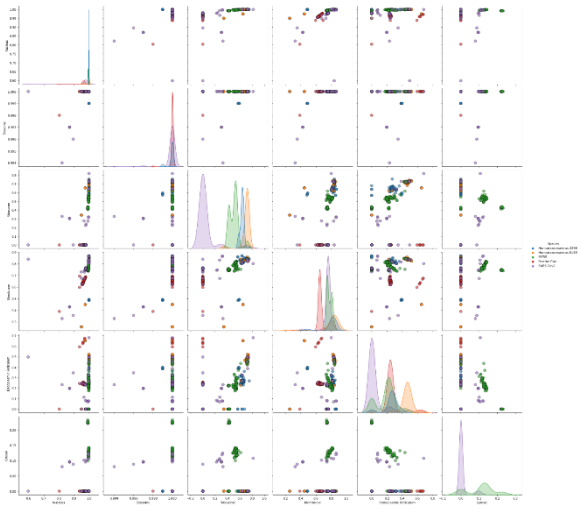

Figure 9: SVM Architecture

```
In [118..  #From Random Dataset
           model3.predict([[0.962,1,0,0.645,0.239,0]])
           #The input accurately predicted for >HM025764.1 Porcine deltacoronavirus strain CH/JX3GS01/P50, complete genome
Out[118.. array(['Porcine Cov'], dtype=object)

In [119..  #From Random Dataset
           model3.predict([[0.978,1,0,0.761,0,0]])
           #The input accurately predicted for >NC_045512.2 Severe acute respiratory syndrome coronavirus 2 isolate Wuhan-Hu-1,
Out[119.. array(['SARS Cov2'], dtype=object)

In [120..  #From Random Dataset
           model3.predict([[0.993,1,0.396,0.754,0.233,0]])
           #The input accurately predicted for >KF745068.1 Middle East respiratory syndrome coronavirus isolate FRA/UAE, complet
Out[120.. array(['MERS'], dtype=object)

In [121..  #From Random Dataset
           model3.predict([[0.999,1,0.737,0.835,0.515,0]])

           #The input accurately predicted for >LC687394.1 Human coronavirus NL63 Fukushima_H257_2018 RNA, complete genome
Out[121.. array(['Human coronavirus NL63'], dtype=object)
```

Figure 10: SVM Prediction Result

```
[288] from sklearn.metrics import classification_report,confusion_matrix

      cr3 = classification_report(y_test,svmprediction3)

      print(cr3)
```

	precision	recall	f1-score	support
Human coronavirus 229E	0.89	0.80	0.84	10
Human coronavirus NL63	0.80	1.00	0.89	16
MERS	1.00	1.00	1.00	21
Porcine Cov	1.00	0.95	0.98	21
SARS Cov2	0.97	0.91	0.94	33
accuracy			0.94	101
macro avg	0.93	0.93	0.93	101
weighted avg	0.95	0.94	0.94	101

Figure 11: SVM Classification Report

Table 3 The Result of Model Performances

A) Accuracy

	DT	**RF**	**SVM**
Accuracy (%)	94.1	95	93.1

B) Matthews Correlation Coefficient

	DT	**RF**	**SVM**
MICC (%)	92.4	93.6	91.1

C) Precision

	Precision (%)				
Species	A	B	C	D	E
DT	80	88	91	100	100
RF	83	93	95	95	88
SVM	89	80	100	97	100

D) Recall

	Recall (%)				
Species	A	B	C	D	E
DT	100	78	95	100	97
RF	100	81	100	90	88
SVM	89	100	100	95	91

E) F1-Score

	F1-Score (%)				
Species	A	B	C	D	E
DT	89	82	93	100	98
RF	91	87	98	93	88
SVM	84	89	100	98	94

Legend for species:
A – Human Coronavirus 229E
B – Human Coronavirus NL63
C – MERS
D – Porcine Cov
E –SARS Cov2 (Covid-19)

IV. CONCLUSION

There are several achievements made through this study. For now, it can be concluded that the Random Forest Method performance is more outstanding and helpful compared to Decision Tree and Support Vector Machine as expected from the beginning of this research. Therefore, Random Forest can be considered if you want to develop a classifier for the Subcellular Localization Data.

ACKNOWLEDGEMENT

This work was supported by the UTM Fundamental Research Grant (UTMFR), Vot No: PY/2023/01586 and PY/2022/01312 from Universiti Teknologi Malaysia.

REFERENCES

[1] P. J. Thul *et al.*, "A subcellular map of the human proteome," *Science (80-.).*, vol. 356, no. 6340, 2017.
[2] S. Raschka and V. Mirjalili, "Python machine learning : machine learning and deep learning with python, scikit-learn, and tensorflow 2," *Int. J. Knowledge-Based Organ.*, vol. 11, no. 1, p. 741, 2021.
[3] W. Mckinney, "Wes McKinney-Python for Data Analysis-O'Reilly Media (2012)," *Eff. Br. mindfulness Interv. acute pain Exp. An Exam. Individ. Differ.*, vol. 1, pp. 1689–1699, 2015.
[4] K. A. Shastry, H. A. Sanjay, In:, K. Srinivasa, and G. Siddesh, "Statistical Modelling and Machine Learning Principles for Bioinformatics Techniques, Tools, and Applications. Algorithms for Intelligent Systems," in *Machine Learning for Bioinformatics*, K. G. Srinivasa, G. M. Siddesh, and S. R. Manisekhar, Eds. Singapore: Springer Singapore, 2020.
[5] Navlani, "Python Decision Tree Classification with Scikit-Learn DecisionTreeClassifier - DataCamp," 2018.
[6] N. Tyagi, "Understanding the Gini Index and Information Gain in Decision Trees", 2020.
[7] E. R. Sruthi, "Random Forest | Introduction to Random Forest Algorithm", 2021.
[8] T. Penumudy, "Random Forest: Simplified.," *Medium.*
[9] N. Donges, "Random Forest Algorithms: A Complete Guide," *Built In*, 2021.