Enhancing Understanding of Arbovirus Competence through Machine Learning and Feature Analysis

Wilson Phillips

A DISSERTATION

Submitted to

The University of Liverpool

in partial fulfilment of the requirements

for the degree of

MASTER OF SCIENCE

Student Declaration

I confirm that I have read and understood the University’s Academic Integrity Policy. I confirm that I have acted honestly, ethically and professionally in conduct leading to assessment for the programme of study. I confirm that I have not copied material from another source nor com mitted plagiarism nor fabricated data when completing the attached piece of work. I confirm that I have not previously presented the work or part thereof for assessment for another University of Liverpool module. I confirm that I have not copied material from another source, nor colluded with any other student in the preparation and production of this work. I confirm that I have not incorporated into this assignment material that has been submitted by me or any other person in support of a successful application for a degree of this or any other university or degree-awarding body.

SIGNATURE ______Wilson Phillips________________________________

DATE October 8, 2024

Acknowledgements

I would like to extend my deep gratitude towards Dr Maya Wardeh for providing me with the necessary data, as well as guidance and support during my project, which could not have been possible without herself.

Enhancing Understanding of Arbovirus Competence through Machine Learning and Feature Analysis

Abstract

The danger arboviruses pose cannot be underestimated; their ability to infect people through a vector of disease such as a mosquito causes a severe danger to public health, such as with malaria. To understand what factors cause the spread of disease, three levels of competence must be studied; infection of the vector with an arbovirus, in this case mosquitos, dissemination, which is the spread of the virus throughout the mosquito, and transmission of the virus from the mosquito to a human. A noisy dataset containing various features which are of interest in terms of competence has been cleaned, and three datasets have been constructed for each of these levels of competence. After this, machine learning algorithms were applied, with the target variable being composed of 3 classes. Class 0 contains 0%, indicating inability of the arbovirus to infect the mosquito (infection dataset), attain dissemination (dissemination dataset), or transmit to a host (transmission dataset). Class 1 contains 1-49% indicating a low ability of the arbovirus to infect the mosquito, attain dissemination or transmit to a host. Class 2 contains 50-100%, indicating a high ability of the arbovirus to infect the mosquito, attain dissemination or transmit to a host, comparing different models and tuning hyperparameters to obtain the best results. Features were then analysed to identify the 20 most contributing features for competence in each of the classes, for all three datasets.

Statement of ethical compliance

Data used falls under category B0; all published work with be met with strict accordance with the university's ethical guidelines.

Contents

Introduction and Background

Viruses transmitted by mosquitoes, arboviruses, pose major threats to animal and human health, leading to severe morbidity and wide range of symptoms in humans and animals. Such arboviruses that affect humans are Dengue and zika virus, and examples of those which affect animals are west Nile virus, Usutu virus. Overall, the measure of acquiring and transmitting a virus is called competence; in order for a mosquito to be competent the first stage would be the mosquito becoming infected, for the infection to disseminate (crossing the midgut barrier) and finally for the mosquito to replicate in the saliva.

This study aims at developing models to predict the ability of mosquito species to become infected and transmit a wide range of arboviruses. Then by using these models, the most influential features in predicting different levels of arbovirus competence can be better understood. By understanding the mechanisms behind arbovirus competence, development of mitigation strategies can be set in place for arboviral disease control (Agarwal *et al,* 2017).

Instances of literature regarding exploring features of arbovirus competence using machine learning is limited, however machine learning has previously been explored to determine driving mechanisms of arbovirus outbreaks (Alkhamis *et al,* 2021). This is similar to determining competence mechanisms in the sense that the driving forces behind outbreaks are complex, so using machine learning models in this fashion makes complete sense. Models used were spatially explicit, and included the algorithms Support Vector Machine, Random Forest and Extreme Gradient Boosting, all used in developing classification models.

Another similar project is by Jiang *et al,* 2018, where transmission risk of Zika virus was mapped using three machine learning models: backward propagation neural network, random forest and gradient boosting machine, with the backward propagation neural network obtaining the best predictive accuracy out of the three. Performance was measured using 10-fold cross-validation area under the curve. The reasoning behind the features being investigated were similar to that of this project, as they were investigating features which increases the chance of Zika transmission, however features were more to do with the external environment such as climate, humidity, etc. instead of mosquito/viral features.

Kaur *et al,* 2022, discusses the use of classification amongst other things, surmising various machine learning and deep learning approaches for predictive modelling of vector-borne diseases. Various metrics are used to evaluate the performance of models such as F1 score, area under the curve when dealing with binary classification and receiving operator characteristic curve. The paper also highlighted the importance of selecting the model selection and assessment procedure, and how this is crucial to developing a model which performs well.

Each instance in the dataset represents an individual experiment containing mosquitos. These experiments measure data about infection of the mosquito with the virus, dissemination of the virus in the mosquito, and transmission of the virus to a host. The features that come along with each experiment represents information surrounding the mosquito, such as species, what blood they feed on, where they are from and experimental conditions. There are also features which provide detail about the arbovirus, such as species, and experimental conditions. A total of 4 datasets were combined, with the original dataset containing the information stated previously and the other 3 datasets containing more in depth information. One being biological information surrounding the DNA in virus species, another being the proportion of 4 types of blood each species feeds on, this being amphibian, avian, mammalian and reptilian, and another providing details surrounding the environmental conditions for each location id, such as bioclimate, climate change velocity and human population. Origins of mosquitos in the dataset were from various biogeographical realms, which is shown on figure 1, with the exception of the group 'colony' which had mixed origins:

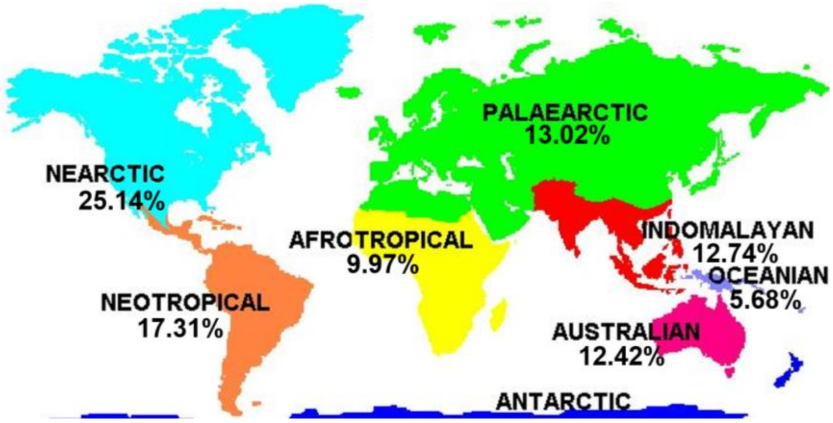


Figure 1- Distribution of mosquitos used in the dataset (Ritter, 2012)

Supervised machine learning models will be constructed and compared to each other with various scoring metrics such as Mathew's correlation coefficient (MCC), area under the curve (AUC), F1 score, and accuracy used to decide which is the best model to use. The models used are Random Forest, Support Vector Machine (SVM), Naïve Bayes, XGBoost (short for Extreme Gradient Boosting) and a multi layered perceptron (MLP). By applying these models to the data, a decision can be made to which model(s) will be used to highlight feature importance, depending on which model(s) produce the best metrics in predicting the correct classes.

Scikit-learn is used to import algorithms except for XGBoost, which is its own library. Random forests are widely used in many bioinformatic tasks due to its succession with complex datasets where the dimensionality is quite high (Boulesteix *et al,* 2012). Another benefit of random forest is the built in feature importance scores which allows for analysis of features and their contributions to the model, such as with classification tasks, their importance in terms of owning a distinguishable difference in value within the various classes. SVMs, known for their good generalisation ability and robustness, rely on a combination of empirical error and the complexity of the hypothesis space, with VC being a measure of this complexity (Awad & Khanna, 2015). While not often successful in probabilities, naïve bayes generally has an increased bias but reduced variance which generally allows it to work well as a classifier, however it is typically not the method of choice (Shonlau, 2023). XGBoost, like random forest, is a tree-based system which often outperforms other classifiers. It can handle high dimensional data as it is highly scalable and parallelizable, as well as being fast in its execution (Sinha *et al,* 2020). XGBoost also has a built-in feature importance score. MLPs can distinguish between nonlinearly separable data from its nonlinear activation functions present in each neuron, which makes it an ideal candidate for complex high dimensional datasets. They do however have a so called 'black box' identity when it comes to modelling due to its limited interoperability when it comes to analysing variables (Etemadi *et al,* 2023).

The outcome of this project is to gain insight into which features are important in determining distinct levels (no competence, low competence or high competence) of 3 different types of competence (infection, dissemination or transmission). By using SHAP (Shapley Additive exPlanations) features can be analysed (Lee *et al,* 2023) and insight into the non-linear relationships can be gained.

Aims and Requirements

The aim of this project is to successfully deal with the noisy dataset provided to identify which features, or combinations of features, link to arbovirus competence. This will be achieved through SHAP value analysis. Data preprocessing is an important part of this project due to the nature of the noisy data set; by attempting to remove as much noise as possible, these models will perform better and provide a greater understanding of which features are important in mosquito competence.

In summary the aims are:

- Preprocess the dataset to effectively deal with noise and improve the quality of the dataset.
- Build machine learning models to highlight which features link to arbovirus competence.
- Highlight which features are the most important contributing factors towards different levels of arbovirus competence.

Design and Implementation

Preprocessing

Data was pre-processed thoroughly prior to applying any models to it:

1. Data integration- combining multiple datasets together
2. Domain definition- splitting the database into 3 separate datasets, infection, dissemination and transmission
3. Data cleaning- removal of redundant features and features which contained too many null values
4. Feature engineering of the target variable- target variables were not set up for classification tasks
5. Data transformation- performing one hot encoding on categorical features, normalisation on numerical features

Data, combined from 4 datasets, first is split intro 3 depending on the target variable. Target variables represented either infection of the mosquito with the virus, dissemination of the virus in the mosquito, or transmission of the virus from the mosquito to a host. These were represented as percentage, with the percentage showing the proportion of mosquitos successful in that specific level of competence.

After splitting the data into three separate datasets, noise needed attending to. To deal with said noise, various factors had to be taken into consideration, such as which models deal with imbalanced data, performing feature selection based on biological information, whether a certain feature is explained in another feature, or whether the number of values which are missing from a feature is an acceptable amount. This last one was especially important as there were some features where the missing data was as high as 98%, which obviously cannot be included. There must be a balance between what should be included as they are important features, and what should be excluded so the model has enough data to perform well.

The target feature then needed engineering, which was originally in the form of a percentage in the dataset. For the infection dataset, this was percentage of viruses which successfully infected the mosquitos in each experiment. For the dissemination dataset, this was percentage of viruses which achieved successful dissemination in the infected mosquitos in each experiment. For the transmission dataset, the percentage was the proportion of mosquitos which transmitted the virus to a host in each experiment. The feature engineering was carried out in two ways, via multiple classes and via binary classes. The multi-class system is the system intended for use however a binary option was made just to compare to each other and test which version attains the best metrics. The multi-class system was made using 3 classes:

- Level 0- percentage = 0% indicating there was no success in that type of competence (infection, dissemination or transmission).
- Level 1- percentage = 1-49% indicating there was low success in that type of competence (infection, dissemination or transmission).
- Level 2- percentage = 50-100% indicating there was high success in that type of competence (infection, dissemination or transmission).

The reasoning behind this multi-class approach is to evaluate important features for 3 levels of competence instead of just 2 levels, with these 2 levels being either 0%, or 1-100%. If this binary system were used, there would not be any information available to split features relating to high levels of competence and low levels of competence. By attaining features which are important in classifying all 3 levels of competence, it makes more sense.

Categorical features then required encoding into numerical forms via one hot encoding. This however had to be set with a limit, as the total number of categories was far too high to include every single category, which would have affected the models if the number of features were too many. Two ways were explored in dealing with this: either building a threshold where features which appeared below a certain percentage such as 2% or 3% were labelled as ‘other’, or the other option, only including the ten most frequent categories and labelling every other category as ‘other’. The decision was made to go with the former and include the top ten. The issue with the percentage threshold option was that some features included so many categories that every category present was quite rare and close to the threshold, therefore for some features, only two or three categories may be above the threshold.

Building and comparing models

With imbalanced datasets, a high amount of bias is present, which can affect scoring metrics making them more optimistic. Mathew’s Correlation coefficient (MCC) however, is a robust metric which presents the lowest bias in imbalanced data when considering errors (Luque *et al,* 2019). This is the primary metric which is used in tuning hyperparameters as well as comparing models as class imbalance is an issue in the data present.

Models are compared to each other with MCC being the deciding factor on which of the models should be used, as well as properties of the models themselves. Of the models chosen, hyperparameters are optimised using a Bayesian Optimisation method. Two ways of testing for good generalisation in these models are used, one is to compare the MCC obtained from the optimised models with training data, to the optimised models with testing data, to see whether the model is overfitting. Underfitting is mitigated by optimising hyperparameters. Another way of testing for good generalisation is to check the variation throughout the cross validated folds in the optimised models, to ensure no folds are being over/underrepresented during training. Standard deviation is calculated for MCC between each CV fold to ensure variance is low for each model.

Feature importance

Feature importance is then explored through built in feature importance methods if available in the models, and SHAP (SHapley Additive exPlanations) based methods for individual classes. Some models such as Random Forest and XGBoost have built in feature importance, however by using SHAP, the distribution of instances for each feature can be displayed through summary plots, providing essential information in how feature values contribute towards classification for each class, or how they contribute towards not classifying for that specific class. SHAP is based off game theory with the inputs acting as players and the prediction the payout (Ekanayake *et al*, 2022), with SHAP determining a score for the contribution each ‘player’ provided, with there being various versions of SHAP for types of machine learning model categories. This is explored further once it is decided which models to use and whether they have built in feature importance scores.

Testing and Evaluation

Throughout the code, print statements were used as well as examining any changes to datasets to ensure code worked properly. When using algorithms and open source libraries, APIs and user manuals from the creators of the algorithms were used to ensure code was used properly, and to provide explanation in how to use certain features of the algorithms. All graphical plots were constructed using matplotlib unless stated otherwise.

The distribution of classes was, as stated prior, shown to be imbalanced. These are as follows:

Table 1- Multi-class proportions

	Class 0 (not competent)	Class 1 (low competence)	Class 2 (high competence)
Training Infection	12.03%	32.69%	55.28%
Testing Infection	12.31%	34.27%	53.42%
Training Dissemination	23.83%	23.46%	52.71%
Testing Dissemination	21.33%	22.43%	56.24%
Training Transmission	40.28%	40.23%	19.49%
Testing Transmission	41.17%	39.22%	19.61%

As for binary versions which was compared to the multiclass models, these imbalances were more severe, with class 0 being a single class, and class 1 and 2 being a single class:

Table 2- Binary class proportions

	Class 0 (not competent)	Class 1 (competent)
Training Infection	12.03%	87.97%
Testing Infection	12.31%	87.69%
Training Dissemination	23.83%	76.17%
Testing Dissemination	21.33%	78.67%
Training Transmission	40.29%	59.72%
Testing Transmission	41.17%	58.83%

Comparing models

Classification models were tested and compared to each other to decide which options to go with for evaluating feature importance. Initially with all default settings, the results are as follows for all three datasets, with binary metrics also shown to compare:

Table 3- Infection (multi class) metrics

	MCC	AUC	Accuracy	Macro avg precision	Macro avg recall	Macro avg F1
Random Forest	0.527	0.856	0.73	0.68	0.66	0.67
SVM	0.398	0.815	0.67	0.66	0.55	0.57
Naïve bayes	0.110	0.652	0.30	0.44	0.41	0.26
XGBoost	0.573	0.877	0.76	0.73	0.69	0.71
MLP	0.507	0.862	0.72	0.73	0.73	0.73

Table 4- Dissemination (multi class) metrics

	MCC	AUC	Accuracy	Macro avg precision	Macro avg recall	Macro avg F1
Random Forest	0.493	0.845	0.71	0.66	0.65	0.65
SVM	0.411	0.792	0.68	0.65	0.56	0.58
Naïve bayes	0.161	0.678	0.30	0.44	0.41	0.26
XGBoost	0.515	0.857	0.72	0.68	0.66	0.67
MLP	0.498	0.835	0.72	0.71	0.72	0.72

Table 5- Transmission (multi class) metrics

	MCC	AUC	Accuracy	Macro avg precision	Macro avg recall	Macro avg F1
Random Forest	0.451	0.827	0.65	0.64	0.63	0.63
SVM	0.405	0.787	0.62	0.64	0.58	0.59
Naïve bayes	0.140	0.688	0.35	0.50	0.43	0.34
XGBoost	0.517	0.848	0.69	0.69	0.67	0.68
MLP	0.481	0.815	0.69	0.70	0.69	0.69

To compare with how the binary system worked, here are metrics for the binary version:

Table 6- Infection (binary) metrics

	MCC	AUC	Accuracy	Macro avg precision	Macro avg recall	Macro avg F1
Random Forest	0.513	0.879	0.90	0.78	0.73	0.75
SVM	0.386	0.848	0.90	0.83	0.61	0.65
Naïve bayes	0.087	0.640	0.23	0.54	0.54	0.23

XGBoost	0.577	0.896	0.92	0.83	0.75	0.78
MLP	0.516	0.892	0.91	0.90	0.91	0.90

Table 7- Dissemination (binary) metrics

	MCC	AUC	Accuracy	Macro avg precision	Macro avg recall	Macro avg F1
Random Forest	0.547	0.879	0.85	0.78	0.76	0.77
SVM	0.501	0.834	0.85	0.82	0.70	0.73
Naïve bayes	0.179	0.665	0.39	0.59	0.59	0.39
XGBoost	0.590	0.891	0.87	0.81	0.78	0.79
MLP	0.579	0.869	0.86	0.86	0.86	0.86

Table 8- Transmission (binary) metrics

	MCC	AUC	Accuracy	Macro avg precision	Macro avg recall	Macro avg F1
Random Forest	0.565	0.853	0.79	0.79	0.77	0.78
SVM	0.509	0.817	0.76	0.78	0.73	0.74
Naïve bayes	0.340	0.739	0.69	0.71	0.64	0.63
XGBoost	0.599	0.876	0.81	0.81	0.79	0.80
MLP	0.534	0.840	0.78	0.77	0.78	0.77

Tables 3-8 show that in general, the binary system provides better metrics in every case apart from for table 3 and table 6, the infection dataset, where the tertiary classification system works better or similar in terms of the metrics provided. This may be because of how the binary infection class imbalance is more severe than the tertiary class imbalance, shown in tables 1 and 2. As for the dissemination and transmission dataset metrics, for tables 4, 5, 7 and 8, although the metrics are better for the binary versions, the project makes more sense if a multi-class system is used, as the feature importance scores will include features which are important in classifying classes with increasing levels of competence, therefore features which have a role when competence levels increase with 3 levels. A scenario where a binary system would be impractical in this project is if there's 2 categorical features, one which appears 300 times with a low transmission percentage (1-49%) and 100 times with a high transmission percentage (50-100%), and the other categorical feature which appears 50 times with a low transmission percentage and 350 times a high transmission percentage. If a binary system was used, there would be no information regarding which feature leads to higher transmission and both would be viewed as equals in this perspective.

After deciding that Random Forest and XGBoost provided the best metrics, hyperparameter tuning was performed. MLP was also considered however due to the 'black bock' nature of the algorithm (Etemadi *et al,* 2023) and the fact that it did not display much better metrics that XGBoost and Random Forest, it wasn't explored further. It was slightly better than random forest in the transmission dataset, but that is all.

Hyperparameter optimisation

Bayesian optimisation was used to tune hyperparameters for each model. It requires less computational resources to finish tuning than grid search; random search was also considered, but after comparing the MCC Bayesian optimisation could achieve, it was decided to go in that direction. While grid search is the optimum solution when it comes to tuning hyperparameters, alternatives are available when computational resources are scarce, and time is an issue. Both optimisation methods were tested for the infection dataset, Random Forest model, shown in figures 2 and 3, with Bayesian optimisation obtaining slightly better hyperparameters with an MCC of 0.559 when used on the test data, compared with random searches best hyperparameters which gave an MCC of 0.550 when used on the test data.

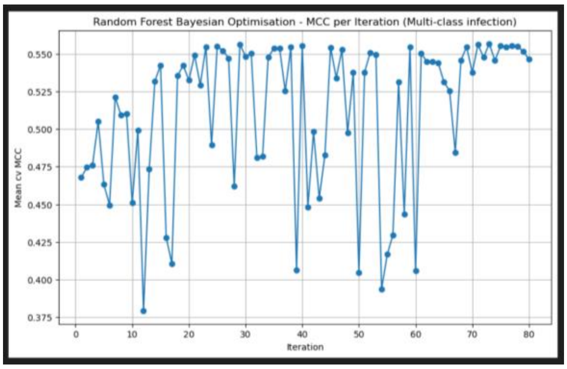


Figure 2- Bayesian optimisation of infection dataset, Random Forest multi-class model

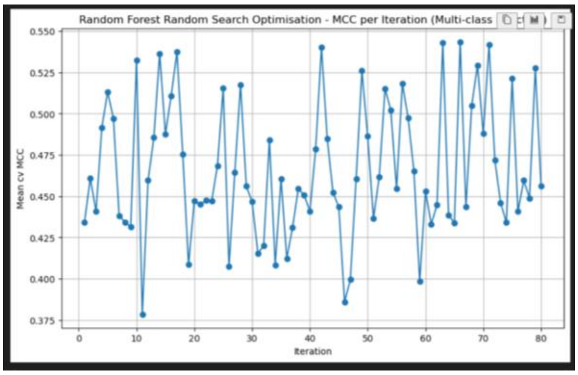


Figure 3- random search optimisation of infection dataset, Random Forest multi-class model

Hyperparameter tuning included 7 hyperparameters in Random Forest and 6 hyperparameters in XGBoost. For Random Forest the hyperparameters were:

- n_estimators
- criterion
- max_depth
- min_samples_split
- min_samples_leaf
- max_features
- bootstrap

A study into which hyperparameters contribute the most vairance in various datasets using Random Forest (van Rijn & Hutter 2018) shows that the most important features include min_samples_leaf and max_features. The range of these hyperparameters was therefore explored thoroughly with min_samples_split set between 2 and 40, and max_features testing sqrt, log2, 0.2, 0.4, 0.6 and 0.8.. The number of trees, n_estimators, was set between 100 and 1000, with this reprisenting the number of trees and the max_depth was set to between 10 and 30. Depending on these values, the model can overfit with high values, or underfit with low values. The values these take on dpends on the complexity of the dataset. Underfitting was not an issue as the hyperparameters used came from the best MCC, however overfitting was a concern. All 3 criterions were tested, these being gini, log_loss and entropy, and bootstrap was set to true, which enables bagging, shown to create a strong classifier from many weak classifiers (Datta & Ghosh, 2014).

For XGBoost, 6 hyperparameters were used, which are as follows:

- n_estimators
- max_depth

- learning_rate
- gamma
- reg_alpha
- reg_lambda

Again n_estimators and max_depth are important for setting the complexity of the model depending on how complex the data is. These were set between 100 and 1000 for n_estimators and between 4 and 20 for max_depth. Learning rate helps shrink the boosting process via weighting, making fitting more conservative (Shi *et al,* 2019), which was tested between 0.05 and 0.4. Gamma was set to between 0 and 1 and is the minimum loss reduction required to make a split on a leaf node, so the larger the gamma, the greater the conservativeness. The same goes for reg_alpha and reg_lambda with these making the model more conservative the higher their values. Lamda is L2 regularization and alpha is L1 regularization for weights. These were both tested between 0.3 and 0.6, with the range coming from some trial and error.

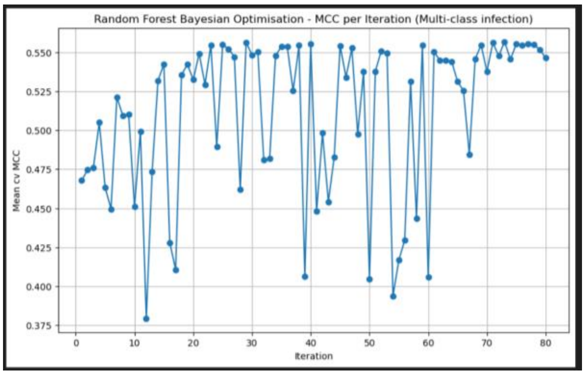


Figure 4- Bayesian optimisation of infection dataset, Random Forest multi-class model

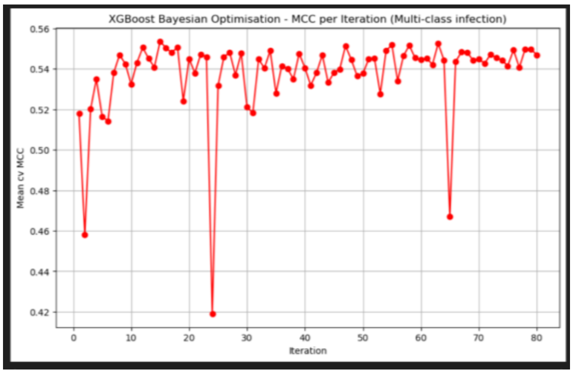


Figure 5- Bayesian optimisation of infection dataset, XGBoost multi-class model

Figure 4 and 5 shows the bayesian optimisation in work, comparing the infection multi-class models for both Random Forest and XGBoost. To test for overfitting, the optimul MCC score was calculated for the training data, from which the hyperparameters were optimised, and then calculated for the test data, and both values were compared to ensure the test score wasn't much lower than the score from the training data. There were some signs of overfitting in the dissemination dataset, which was mitigated by creating a new search space specifically for dissemination for both Random Forest and XGBoost. This decreased the difference in the two scores; for Random Forest the MCC was 0.549 from the training data and 0.526 from the test data, which was changed to a training score of 0.548 and a test score of 0.536. For the overfitted XGBoost dissemination model, the training score was 0.546 and test score 0.530, which was changed to 0.534 training score and 0.537 test score. The full list of training MCC, obtained from the mean of the cv, and test MCC, using the tuned hyperparameters on the test data, can be seen below:

Table 9- comparison of MCCs from the training data and the test data

	Train MCC (mean from cv)	Test MCC
RF Infection	0.557	0.559
RF Dissemination	0.548	0.536
RF Transmission	0.494	0.517
XGB Infection	0.554	0.576
XGB Dissemination	0.534	0.537

XGB Transmisson	0.497	0.519

XGBoost provided the best MCC for all three datasets when compared to Random Forest with both having optimised hyperparameters. The MCC for XGBoost was 0.576 for infection, 0.537 for dissemination and 0.519 for transmission. The full details of metrics can be found below:

Table 10- metrics from optimised infection models

	MCC	AUC	Accuracy	Macro avg precision	Macro avg recall	Macro avg F1
Random Forest	0.561	0.875	0.75	0.72	0.67	0.69
XGBoost	0.576	0.876	0.76	0.73	0.69	0.70

Table 11- metrics from optimised dissemination models

	MCC	AUC	Accuracy	Macro avg precision	Macro avg recall	Macro avg F1
Random Forest	0.530	0.860	0.73	0.69	0.66	0.67
XGBoost	0.537	0.855	0.74	0.70	0.66	0.68

Table 12- metrics from optimised transmission models models

	MCC	AUC	Accuracy	Macro avg precision	Macro avg recall	Macro avg F1
Random Forest	0.519	0.852	0.69	0.71	0.66	0.67
XGBoost	0.519	0.847	0.69	0.71	0.67	0.68

To further validate all models ability in generalisation, and to ensure there was minimal overfitting, MCC for the cross validated folds also were calculated for each optimised model, as was standard deviation between these MCC values. This was to ensure that variance was low, the results of which can be found in figure 6. The dispersion of individual cv folds indicates whether the model is overfitting or underfitting to any particular fold, and if this variation from the mean is low, it is a good indication of good generalisation.

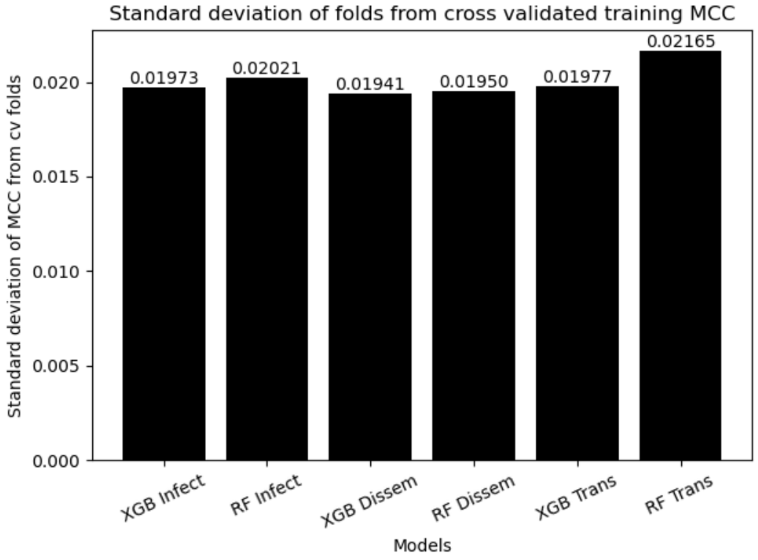


Figure 6- a plot of standard deviation for MCC calculated across cv folds for each model

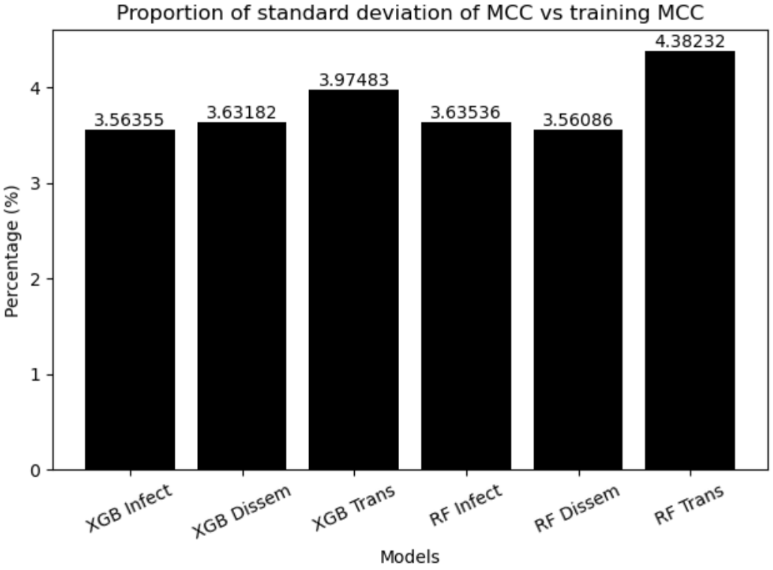
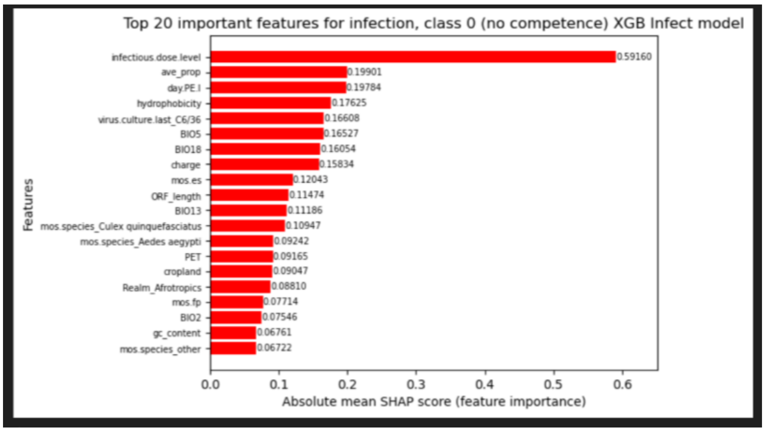


Figure 7- a plot of the proportion of standard deviation of MCC between cv folds vs the training MCC (derrived from the mean of the cv folds MCCs)

Figure 7 represents standard deviation of the cv folds' MCC for each optimised model, calculated as a proportion of the mean MCC. This provides a better idea of the amount of variation present, with the Random Forest transmission model having the highest variation from the mean, while still being reasonably low with all below 5%.

SHAP feature importance

Built in feature importance scores were tested, however the decision was to go with SHAP values for evaluating feature importance. SHAP allows for analysis of specific classes and the feature importance in these classes. Powerful visualisation tools are also available from SHAP which can aid understanding in how these features contribute to the classification. The plots used here are summary plots, which display the top 20 most important features for each class in each model as well as the spread of the instances for that feature, whether the feature value is high (red) or low (blue) for that instance, and the SHAP value, with positive values indicating contribution to that class prediction, and negative values indicating a reduction in confidence for predictions of that class. Other plots in the form of simple bar charts were also used detailing the absolute magnitude for SHAP values of features in specific classes for specific models. The models for each dataset is that of XGBoost due to its superiority in the MCC compared to Random Forest when both are optimised. The most important features, as well as their summary plots, is shown below, with the horizontal bar charts being constructed from matplotlib and the summary plot from SHAP:



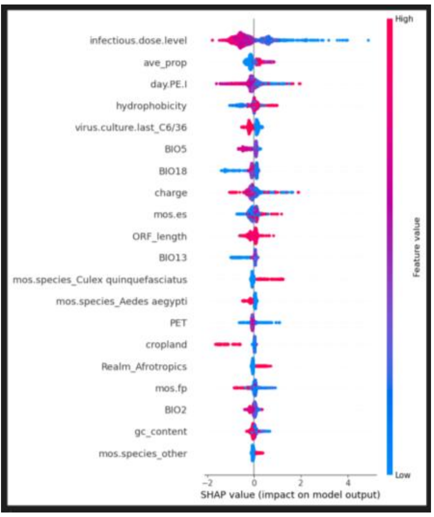
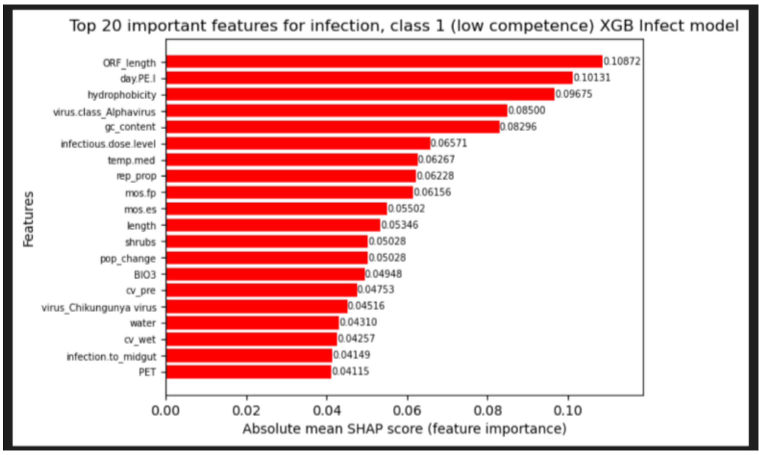


Figure 8a- a bar chart detailing the top 20 most imprtant features for class 0 in the XGBoost infection model.

Figure 8b- a SHAP summary plot detailing The distribution of instances in terms of SHAP value for class 0, XGBoost infection model.



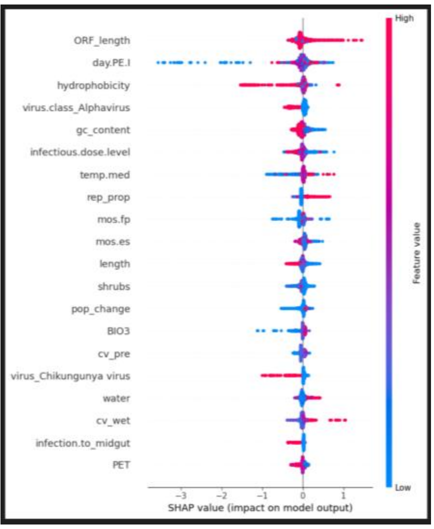
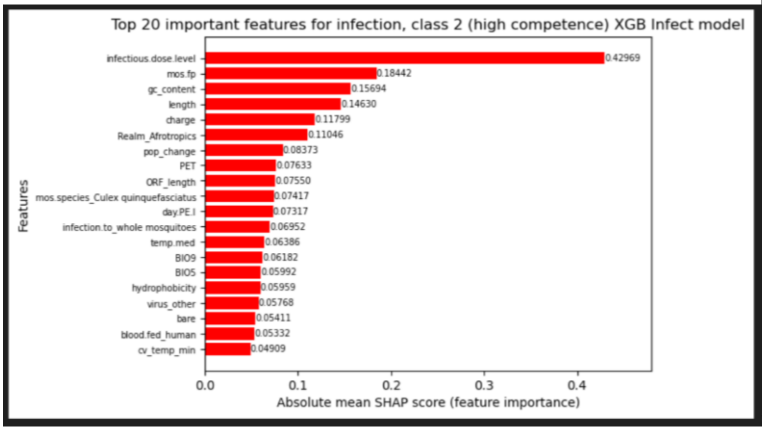


Figure 8c- a bar chart detailing the top 20 most imprtant features for class 1 in the XGBoost infection model.

Figure 8d- a SHAP summary plot detailing the distribution of instances in terms of SHAP value for class 1, XGBoost infection model.



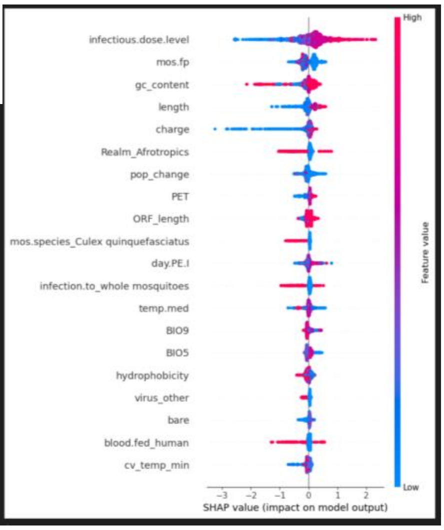
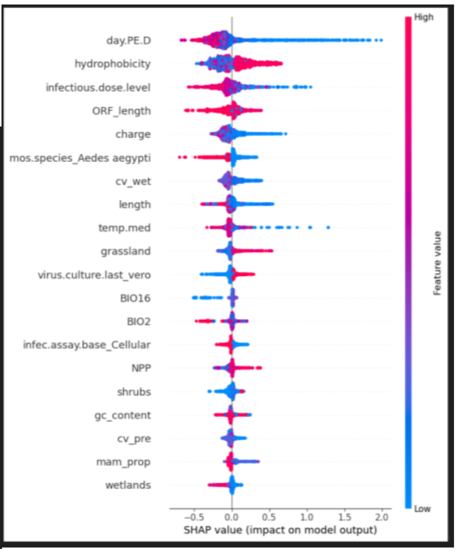


Figure 8e- a bar chart detailing the top 20 most imprtant features for class 2 in the XGBoost infection model.

Figure 8f- a SHAP summary plot detailing the distribution of instances in terms of SHAP value for class 2, XGBoost infection model.



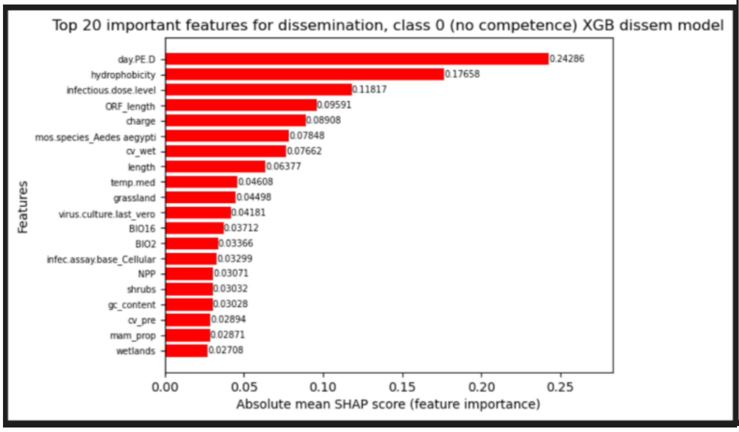


Figure 9a- a bar chart detailing the top 20 most imprtant features for class 0 in the XGBoost dissemination model.

Figure 9b- a SHAP summary plot detailing the distribution of instances in terms of SHAP value for class 0, XGBoost dissemination model.

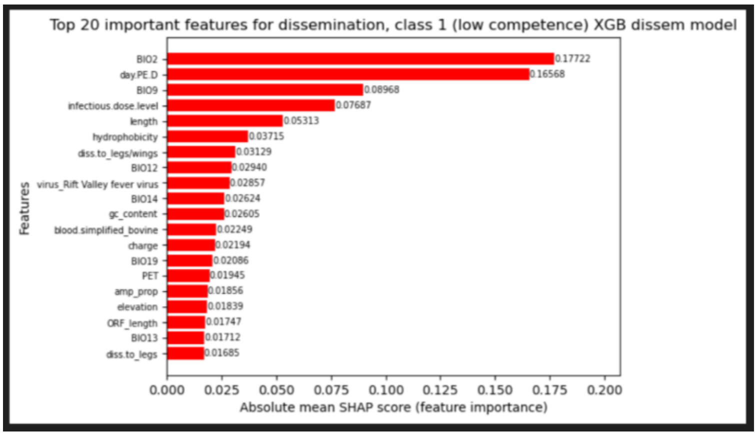


Figure 9c- a bar chart detailing the top 20 most imprtant features for class 1 in the XGBoost dissemination model.

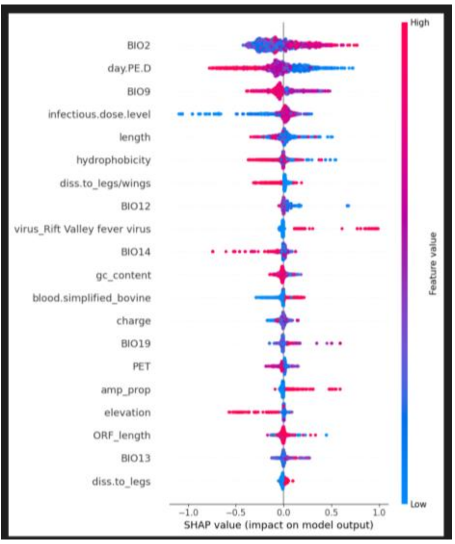
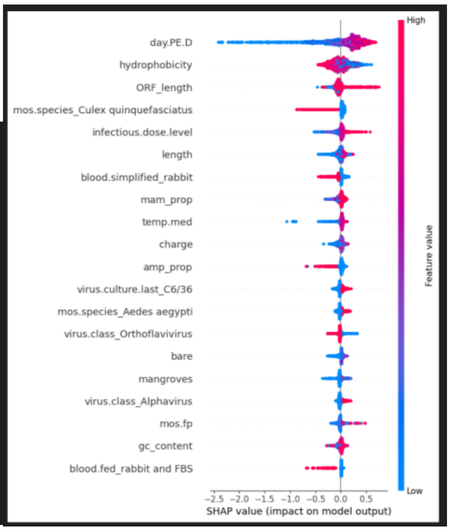


Figure 9d- a SHAP summary plot detailing the distribution of instances in terms of SHAP value for class 1, XGBoost dissemination model.



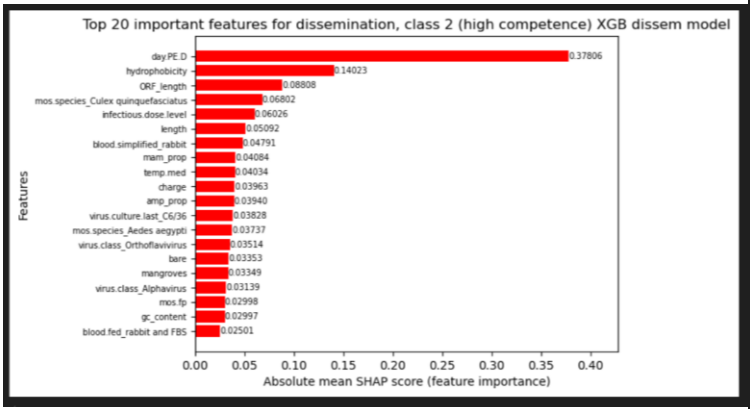


Figure 9e- a bar chart detailing the top 20 most imprtant features for class 2 in the XGBoost dissemination model.

Figure 9f- a SHAP summary plot detailing the distribution of instances in terms of SHAP value for class 2, XGBoost dissemination model.

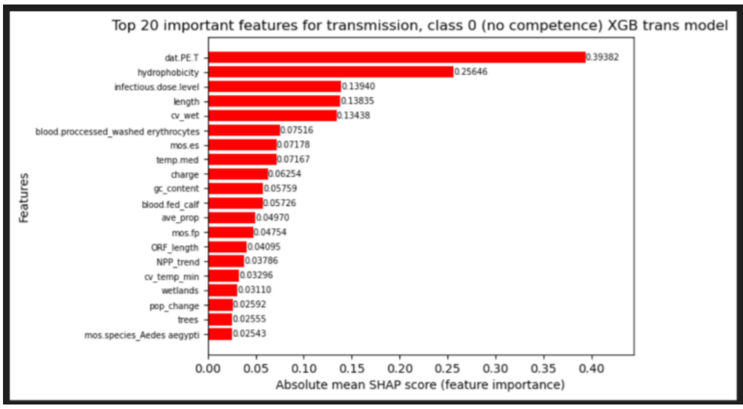


Figure 10a- a bar chart detailing the top 20 most imprtant features for class 0 in the XGBoost transmission model.

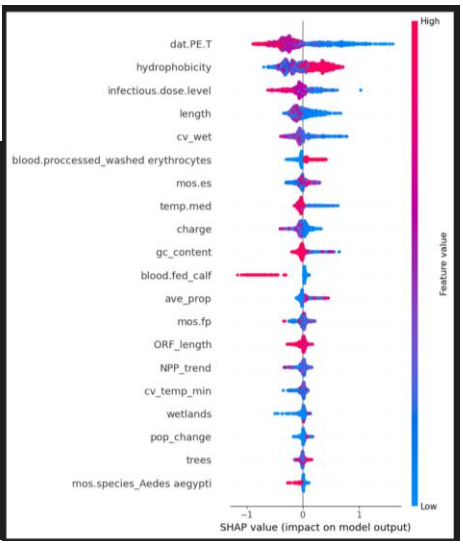
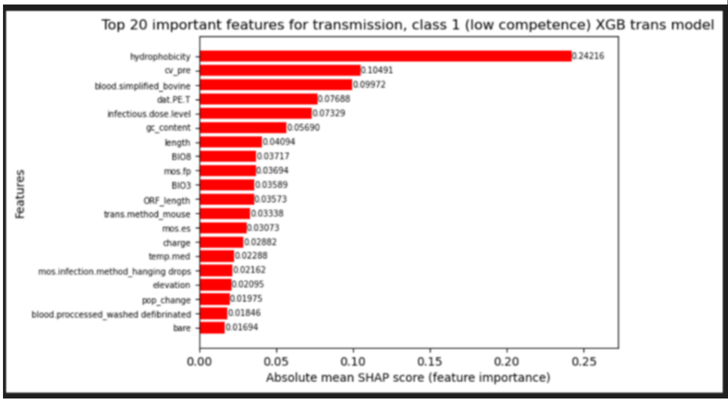


Figure 10b- a SHAP summary plot detailing the distribution of instances in terms of SHAP value for class 0, XGBoost transmisson model.



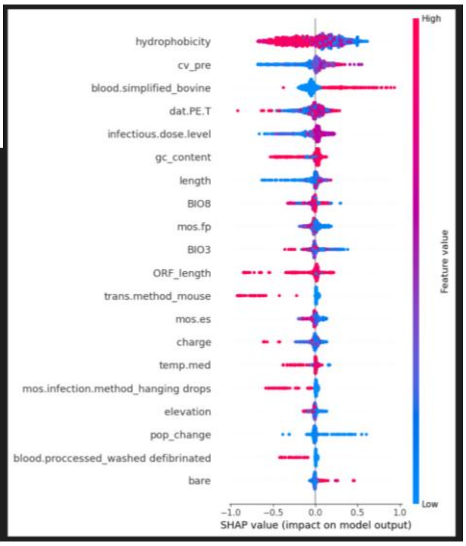
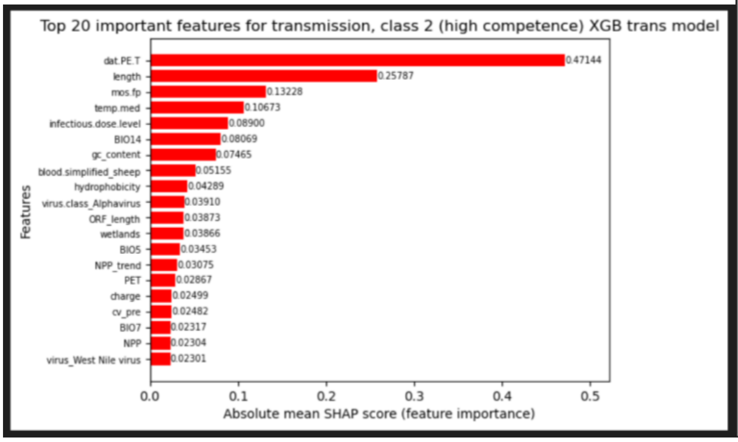


Figure 10c- a bar chart detailing the top 20 most imprtant features for class 1 in the XGBoost transmission model.

Figure 10d- a SHAP summary plot detailing the distribution of instances in terms of SHAP value for class 1, XGBoost transmisson model.



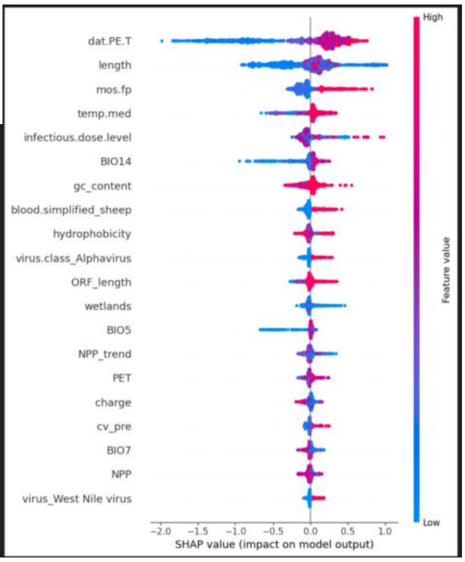


Figure 10e- a bar chart detailing the top 20 most imprtant features for class 2 in the XGBoost transmission model.

Figure 10f- a SHAP summary plot detailing the distribution of instances in terms of SHAP value for class 2, XGBoost transmisson model.

By examining features which are expected to be influencial in certain directions for certain classes, the validity of the other SHAP values can be confirmed. For example figure 8b and 8f shows summary plots for class 0, the 0% class, and class 2, the 50-100% class for the infection dataset. Both classes would be expected to have one of the most important features as infectious.dose.level, which is true. Class 0 would be expected to have high feature values with negative SHAP values, and class 2 would be expected to have high feature values with positive SHAP values, which is the case for both summary plots. Another example of such a feature which is expected to be important in the transmission dataset is dat.PE.T. which is average incubation time for transmission. This would be an important feature in class 0 and 2, and be expected to have negative SHAP values with high feature values in class 0, and positive SHAP values with high feature values in class 2. This is also the case as can be shown in figures 10b and 10f, so there is confidence in other features to be correct in their retrospective feature distribution. Explanations of features shown in the above graphs and plots can be found in table 16 in the appendices.

Further testing of XGBoost

Finally, the XGBoost model was tested for all 3 variations (infection, dissemination and transmission) to examines their abilities to predict classes in unseen data after 2022 to simulate what it would be like in predictions for future data entries.

Table 13- Infection data metrics from 2022 onwards, trained prior to 2022

	MCC	AUC	Accuracy	Macro avg precision	Macro avg recall	Macro avg F1
Optimised XGBoost	0.397	0.781	0.65	0.64	0.59	0.60
Default XGBoost	0.446	0.788	0.68	0.69	0.61	0.64

Table 14- Dissemination data metrics from 2022 onwards, trained prior to 2022

	MCC	AUC	Accuracy	Macro avg precision	Macro avg recall	Macro avg F1

Optimised XGBoost	0.420	0.798	0.67	0.62	0.56	0.57
Default XGBoost	0.449	0.816	0.68	0.64	0.58	0.59

Table 15- Transmission data metrics from 2022 onwards, trained prior to 2022

	MCC	AUC	Accuracy	Macro avg precision	Macro avg recall	Macro avg F1
Optimised XGBoost	0.563	0.869	0.72	0.71	0.73	0.71
Default XGBoost	0.574	0.866	0.73	0.72	0.73	0.72

Results were good for the transmission dataset, table 15, achieving even better results than the transmission MCC with the original test split, however the same cannot be said for the dissemination and infection datasets. In addition, the optimised models do not gain as good metrics as the default version of XGBoost in all models. As overfitting does not seem to be an issue, shown through figure 7, it's possible trends in data change throughout years, which would cause difficulty in predicting data from 2022 onwards when trained on data prior to 2022.

Referring back to the aims, the dataset was successfully preprocessed and features which were useless, described in other features or contained too many N/A values were pruned. Doing so decreases dimensionality and increases the amount of instances available, as instances which contain any N/A values for any features cannot be used. The quality of the dataset was therefore improved. Machine learning models were then fitted to the data, with the objective being to identify the most important features used in classification. Hyperparameters were optimised for the two best models, according to MCC, and the model with the best MCC after optimisation when it came to predicting classes was used for SHAP values analysis. By using SHAP, the 20 most important features for each class in each type of competence were highlighted, with each class referring to no competence (class 0), low competence (class 1) and high competence (class 2).

Conclusion and future work

The features presented represent the most important features for each class. By honing in on the graphs which represent class 2 (the high competence class) insight into which features are responsible for high competence is possible, with these being the features with more red distributed towards the positive SHAP value side in the summary plots, indicating higher feature values. Features which are not responsible for high competence can also be identified if there is more red towards the negative SHAP value side. In addition, by evaluating class 0 graphs which represents no competence, features which are responsible for 0% competence can be shown the same way as mentioned before, by examining the summary plots. It is also important to evaluate features with more red dots in the negative SHAP value side as these indicate features which are responsible for predicting both class 1 and 2 (the competent classes). All in all, these features are great for providing researchers in arboviruses with general information, in which features link to competence, and which don't, with there being 3 types of competence: infection, dissemination and transmission.

XGBoost has accumulated various awards in Kaggle machine learning competitions since its introduction in 2014; in 2015, 17 of the 29 winning solutions of that year used XGBoost, with 11 solutions being deep neural networks (Chen & Guestrin, 2016). It is therefore not surprising that the model performed the best in terms of MCC with the available data. There are several reasons which contribute towards its success; Newton boosting is deployed ahead of gradient boosting which has been shown to outperform gradient boosting in terms of predictive accuracy, as well as less generalization errors (Sigrist, 2021). Another feature of XGBoost is the penalisation techniques through L1 and L2 regularisation which can add different ‘shrinkage factors to individual leaf weights, which places it ahead of MART, another tree boosting model, making it more adaptable than MART (Nielsen, 2016). Nielsen also explains how tree boosting is so effective due to its ability to fit additive tree models which use adaptively determined neighbourhoods, containing strong representational abilities.

Dealing with class imbalance on a data level was not something explored in this project, which would be interesting to explore in future work. Resampling techniques have been shown to improve performance in other imbalanced classification models, however this is dataset dependent, so should be part of the preprocessing stage (Liu, 2004). Doing so could improve models reducing the class imbalance and increase confidence on the feature importance scores.

Depending on the SHAP values present some features may score as an absolute mean of 0, or close to 0, across all 3 classes. These features which do not contribute to classification may be worth removing from the datasets which could increase amount of data available if these features contain N/A values, as well as decrease dimensionality, which therefore could improve the models. In addition, small subsets of data including only the top 20 features for each model could be explored which could produce more insightful results without other features interfering with the SHAP scores.

As for why the yearly predictions for 2022 onwards performed poorly for the infection and dissemination datasets, one could speculate that trends in certain feature values could be different in the past compared to more recently as stated in the evaluation. This would also explain why the optimised models don’t work as well as the default models on this data, however this is merely speculation and further tests need to be carried out to confirm this such as comparing mean feature values with the data available before 2022 and the data available after and including 2022. Additionally, by examining which features are important through SHAP value analysis, ones which are important for transmission may have stayed similar throughout the years but some which are important for classifying infection and dissemination may have different results. This would be a good avenue for future work.

Project Ethics

Ethical issues in this project are not present. Data used does not use any information regarding human participants.

BCS Project Criteria & Self-Reflection

I am confident that the six outcomes listed by the BCS Project have been fully met during the completion of this project. An explanation of each outcome is provided below:

- An ability to apply practical and analytical skills gained during the degree programme

Data science techniques are used throughout the code, successfully building machine learning models and optimising hyperparameters, both skills gained during the degree program. In addition, analytical skills were shown through analysing optimised machine learning models for any overfitting, and understanding how to decrease signs of overfitting through tuning different ranges of hyperparameters.

- Innovation and/or creativity

By separating the original dataset into three, for three types of competence (infection, dissemination and transmission), as well as engineering, from a percentage feature target variable, a tertiary classification system for each dataset which made sense given existing domain knowledge, innovation and creativity were both displayed.

- Synthesis of information, ideas and practices to provide a quality solution together with an evaluation of that solution.

By combining domain knowledge of features which were expected to have been important, with analysing the percentage of N/A values for each feature, as well as whether these features were represented in other features already, dimensionality was reduced and the total number of instances without any N/A features were kept to a high enough amount to build successful models out of. These models were scored based off various metrics. In addition, by using two ways of measuring overfitting (calculating standard deviation and comparing training/testing data metrics), confidence in the generalisation of the optimised models was gained.

- That your project meets a real need in a wider context.

This project aims to increase understanding of the mechanisms behind competence. By doing this, scientists and researchers in this field can use this information to put in place measures of mitigation in arboviral disease control.

- An ability to self-manage a significant piece of work.

I was able to effectively deal with preprocessing a large dataset and gain some useful insight out of it. Timings wasn’t an issue with enough time allocated towards coding as well as the write up. By analysing results, any anomalies were addressed adequately, whether it was because of the code itself being wrong, or techniques not applied properly. If the anomaly couldn’t be resolved, it was mentioned in the report.

- Critical self-evaluation of the process.

I believe that the project worked well, enhancing understanding of arbovirus competence through SHAP value analysis. I would however, if given more time, like to explore more into neural networks, and perhaps other types of models. In addition, more methods of improving the quality of data should have been explored such as resampling techniques to deal with the class imbalance.

References

Agarwal, A., Parida, M. & Dash, P.K., 2017. Impact of transmission cycles and vector competence on global expansion and emergence of Arboviruses. *Reviews in Medical Virology*, 27(5). https://doi.org/10.1002/rmv.1941 [Accessed 27/11/2024]

Alkhamis, M.A. *et al.,* 2021. Environment, vector, or host? using machine learning to untangle the mechanisms driving arbovirus outbreaks. *Ecological Applications*, 31(7). https://doi.org/10.1002/eap.2407 [Accessed 10/08/2024]

Awad, M. & Khanna, R., 2015. Support Vector Machines for classification. *Efficient Learning Machines*, pp. 39-66. https://doi.org/10.1007/978-1-4302-5990-9_3 [Accessed 03/11/2024]

Boulesteix, A. *et al.,* 2012. Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics. *WIREs Data Mining and Knowledge Discovery*, 2(6), pp. 493-507. https://doi.org/10.1002/widm.1072 [Accessed 02/11/2024]

Chen, T. & Guestrin, C., 2016. XGBoost. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785-794. https://doi.org/10.1145/2939672.2939785 [Accessed 14/11/2024]

Datta, J. & Ghosh, J.K., 2014. Bootstrap—an exploration. *Statistical Methodology*, 20, pp. 63-72. https://doi.org/10.1016/j.stamet.2013.08.003 [Accessed 17/11/2024]

Ekanayake, I.U., Meddage, D.P.P. & Rathnayake, U., 2022. A novel approach to explain the black-box nature of machine learning in compressive strength predictions of concrete using Shapley additive explanations (SHAP). *Case Studies in Construction Materials*, 16. https://doi.org/10.1016/j.cscm.2022.e01059 [Accessed 13/11/2024]

Etemadi, S., Khashei, M. & Tamizi, S., 2023. Etemadi reliability-based multi-layer perceptrons for classification and forecasting. *Information Sciences*, 651, p. 119716. https://doi.org/10.1016/j.ins.2023.119716 [Accessed 12/11/2024]

Jiang, D. *et al.,* 2018. Mapping the transmission risk of zika virus using machine learning models. *Acta Tropica*, 185, pp. 391–399. https://doi.org/10.1016/j.actatropica.2018.06.021 [Accessed 01/09/2024]

Kaur, I., Sandhu, A.K. & Kumar, Y., 2022. Artificial intelligence techniques for predictive modeling of vector-borne diseases and its pathogens: A systematic review. *Archives of Computational Methods in Engineering*, 29(6), pp. 3741-3771. https://doi.org/10.1007/s11831-022-09724-9 [Accessed 05/09/2024]

Lee, Y-G. *et al.,* 2022. Shap value-based feature importance analysis for short-term load forecasting. *Journal of Electrical Engineering &amp; Technology*, 18(1), pp. 579-588. https://doi.org/10.1007/s42835-022-01161-9 [Accessed 12/11/2024]

Liu, A.Y-C., 2004. The Effect of Oversampling and Undersampling on Classifying Imbalanced Text Datasets. *The University of Texas at Austin*. https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=cade435c88610820f073a0fb61b73dff8f006760 [Accessed 21/11/2024]

Luque, A. *et al.,* 2019. The impact of class imbalance in classification performance metrics based on the binary confusion matrix. *Pattern Recognition*, 91, pp. 216-231. https://doi.org/10.1016/j.patcog.2019.02.023 [Accessed 09/11/2024]

Nielsen, K., 2016. Tree Boosting With XGBoost. Why Does XGBoost Win “Every Machine Learning Competition?. *Norwegian University of Science and Technology.* https://pzs.dstu.dp.ua/DataMining/boosting/bibl/Didrik.pdf [Accessed 21/11/2024]

Ritter, M.E., 2006. The Physical Environment: an Introduction to Physical Geography. https://www.thephysicalenvironment.com/Book/biomes/biogeographical_realms.html [Accessed 10/11/2024]

Schonlau, M., 2023. The naive bayes classifier. *Statistics and Computing*, pp. 143-160. https://doi.org/10.1007/978-3-031-33390-3_8 [Accessed 08/11/2024]

Shi, X. *et al.,* 2019. A feature learning approach based on XGBoost for driving assessment and risk prediction. *Accident Analysis &amp; Prevention*, 129, pp. 170-179. https://doi.org/10.1016/j.aap.2019.05.005 [Accessed 19/11/2024]

Sinha, N.K., 2020. Developing a web based system for breast cancer prediction using XGboost classifier. *International Journal of Engineering Research and*, V9(06). https://doi.org/10.17577/ijertv9is060612 [Accessed 09/11/2024]

van Rijn, J.N. & Hutter, F., 2018. Hyperparameter importance across datasets. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery &amp; Data Mining*, pp. 2367-2376. https://dl.acm.org/doi/pdf/10.1145/3219819.3220058 [Accessed 17/11/2024]

Appendices

Table 16- A table explaining each feature used in the 3 datasets

Feature	Explanation	Percent NA values (%)	Datasets used in: I/D/T (Infection, Dissemination or Transmission)
amp_prop	Proportion amphibian blood the mosquito species feeds on.	0.000000	I/D/T
ave_prop	Proportion of avian blood the mosquito species feeds on.	0.000000	I/D/T
bare	Fraction of area with the corresponding landcover for that location.	3.687589	I/D/T
BIO1	Annual mean temperature for that location.	3.687589	I/D/T
BIO10	Mean temperature of warmest quarter for that location.	3.687589	I/D/T
BIO11	Mean temperature of coldest quarter for that location.	3.687589	I/D/T
BIO12	Annual precipitation for that location.	3.687589	I/D/T
BIO13	Precipitation of wettest month for that location.	3.687589	I/D/T

BIO14	Precipitation of driest month for that location.	3.687589	I/D/T
BIO15	Precipitation seasonality (coefficient of variation) for that location.	3.687589	I/D/T
BIO16	Precipitation of wettest quarter for that location.	3.687589	I/D/T
BIO17	Precipitation of driest quarter for that location.	3.687589	I/D/T
BIO18	Precipitation of warmest quarter for that location.	3.687589	I/D/T
BIO19	Precipitation of coldest quarter for that location.	3.687589	I/D/T
BIO2	Mean diurnal range (mean of monthly (max temp - min temp)) for that location.	3.687589	I/D/T
BIO3	Isothermality (BIO2/BIO7) (×100) for that location.	3.687589	I/D/T
BIO4	Temperature seasonality (standard deviation ×100) for that location.	3.687589	I/D/T
BIO5	Maximum temperature of warmest month for that location.	3.687589	I/D/T
BIO6	Minimum temperature of coldest month for that location.	3.687589	I/D/T
BIO7	Temperature annual range (BIO5-BIO6) for that location.	3.687589	I/D/T
BIO8	Mean temperature of wettest quarter for that location.	3.687589	I/D/T
BIO9	Mean temperature of driest quarter for that location.	3.687589	I/D/T
blood.fed	Type of blood fed to mosquito (if any).	10.748930	I/D/T
blood.processed	How the bloodmeal is processed.	33.459344	I/D/T
blood.simplified	Simplified blood.	10.748930	I/D/T
built	Fraction of area with the corresponding landcover for that location.	3.687589	I/D/T
charge	Charge of virus DNA (based off virus data id, a unique id for the virus/strain/accession combination).	0.292439	I/D/T
cropland	Fraction of area with the corresponding landcover for that location.	3.687589	I/D/T
cv_pre	Velocity of change in precipitation for that location.	3.687589	I/D/T
cv_temp_mean	Velocity of change in mean temperature for that location.	3.687589	I/D/T

cv_temp_min	Velocity of change in minimum temperature for that location.	3.687589	I/D/T
cv_wet	Velocity of change in wet days for that location.	3.687589	I/D/T
dat.PE.T	Average incubation period for transmission.	0.256776	T
day.PE.D	Average incubation period for dissemination infection.	0.256776	D
day.PE.I	Average incubation period for infection.	0.256776	I
diss.to	Which organ used to determine disseminated infection.	27.888730	D
dissemination.assay.base	Cleaned and simplified disseminated infection assay.	30.413695	D
elevation	Elevation in meters for that location.	3.687589	I/D/T
gc_content	GC (guanine & cytosine) proportion of virus DNA (based off virus data id, a unique id for the virus/strain/accession combination).	0.292439	I/D/T
grassland	Fraction of area with the corresponding landcover for that location.	3.687589	I/D/T
hydrophobicity	Hydrophobicity of virus DNA (based off virus data id, a unique id for the virus/strain/accession combination).	0.292439	I/D/T
infec.assay.base	Cleaned and simplified infection assay.	24.493581	I/D/T
infection.dose.level	Level of dose (-1 to 11) for infection of mosquito.	4.486448	I/D/T
infection.to	Which organ used to determine infection.	24.614836	I/D
length	Length of virus DNA (based off virus data id, a unique id for the virus/strain/accession combination).	0.292439	I/D/T
mam_prop	Proportion of mammalian blood the mosquito species feeds on.	0.000000	I/D/T
mangroves	Fraction of area with the corresponding landcover for that location.	3.687589	I/D/T
mos.es	Evolutionary distinctiveness variable equal splits, for that location.	0.135521	I/D/T
mos.fp	Evolutionary distinctiveness variable fair proportion, for that location.	0.135521	I/D/T

mos.infection.method	Simplified infection method.	7.097004	I/D/T
mos.species	Species name of mosquito used.	0.000000	I/D/T
moss	Fraction of area with the corresponding landcover for that location.	3.687589	I/D/T
NPP	Average NPP (net primary productivity) between 2000 and 2023.	3.687589	I/D/T
NPP_trend	Change in NPP (net primary productivity) between 2000 and 2023.	3.687589	I/D/T
ORF_length	Open reading frame length of virus DNA (based off virus data id, a unique id for the virus/strain/accession combination).	0.292439	I/D/T
perc.trans.infected	Proportion of mosquitoes that transmitted the virus (of those infected) as listed in the primary source (used for target variable of transmission dataset).	67.995720	T
percent.dissem.Infected	Proportion of mosquitoes with a disseminated infection, of those infected, as listed in the primary source (used for target variable of dissemination dataset).	55.149786	D
percent.infected	Proportion of mosquitoes with an infection as listed in the primary source (used for target variable of infection dataset).	27.425107	I
PET	Potential evapotranspiration	3.687589	I/D/T
pop	Average human population between 2000-2020 for that location.	3.687589	I/D/T
pop_change	Human population change between years 2000-2020 for that location.	3.687589	I/D/T
pop00	Human population density in year 2000 for that location.	3.687589	I/D/T
pop05	Human population density in year 2005 for that location.	3.687589	I/D/T
pop10	Human population density in year 2010 for that location.	3.687589	I/D/T
pop15	Human population density in year 2015 for that location.	3.687589	I/D/T
pop20	Human population density in year 2020 for that location.	3.687589	I/D/T
Realm	Realm of the mosquito - note that some have COLONY as	0.000000	I/D/T

	realm, for long established colonies of mixed origins.		
rep_prop	Proportion of reptile blood the mosquito species feeds on.	0.000000	I/D/T
segments	Number of segments in Virus DNA(based off virus data id, a unique id for the virus/strain/accession combination).	0.000000	I/D/T
shrubs	Fraction of area with the corresponding landcover for that location.	3.687589	I/D/T
snow	Fraction of area with the corresponding landcover for that location.	3.687589	I/D/T
temp.constant	Whether a constant temperature (1) or fluctuating range was used (0) in incubation.	1.134094	I/D/T
temp.med	Average incubation temperature.	1.134094	I/D/T
trans.assay.base	cleaned and simplified transmission assay.	52.703281	T
trans.method	cleaned transmission methods.	52.225392	T
trees	Fraction of area with the corresponding landcover for that location.	3.687589	I/D/T
virus	Virus name.	0.000000	I/D/T
virus.class	Classification of the virus (genus/family).	0.014265	I/D/T
virus.culture.last	Last culturing method performed before experiment.	3.352354	I/D/T
virus.culture.last.insect	1 if last is in invertebrate cell, 0 otherwise.	3.352354	I/D/T
water	Fraction of area with the corresponding landcover for that location.	3.687589	I/D/T
wetlands	Fraction of area with the corresponding landcover for that location.	3.687589	I/D/T
year	Year of publication.	0.000000	I/D/T