Applying Convolutional Neural Networks for Inferring Migration Rates in Anopheles Mosquito Populations: Challenges and Insights

Jeremiah Mushtaq
Queen Mary University of London

A dissertation submitted for the degree of
Masters in Science
August 2024

Table of Contents

Abstract

Insecticide resistance in Anopheles mosquito populations poses a significant threat to malaria control efforts, particularly in sub-Saharan Africa. Understanding the genetic mechanisms underlying this resistance is crucial for developing effective strategies to combat malaria. This study explores the application of Convolutional Neural Networks to classify migration levels between two closely related mosquito species, *Anopheles gambiae* and *Anopheles coluzzii*, using simulated genetic data. While CNNs have shown potential in various genomic applications, the results of this project indicate that the model struggled to accurately infer migration levels in this context, achieving only moderate accuracy with frequent misclassifications. Several factors likely contributed to these limitations, including the inclusion of rare variants that introduced noise, the relatively small size of the training dataset, and the arbitrary ordering of individuals in the input data. These challenges underscore the complexity of applying deep learning techniques to demographic inference in population genomics. To address these issues, future research should focus on improving data pre-processing by filtering out rare variants, increasing the diversity and size of training datasets and sorting genetic data to reflect biological significance.

Introduction

The Problem Of Insecticide Resistance

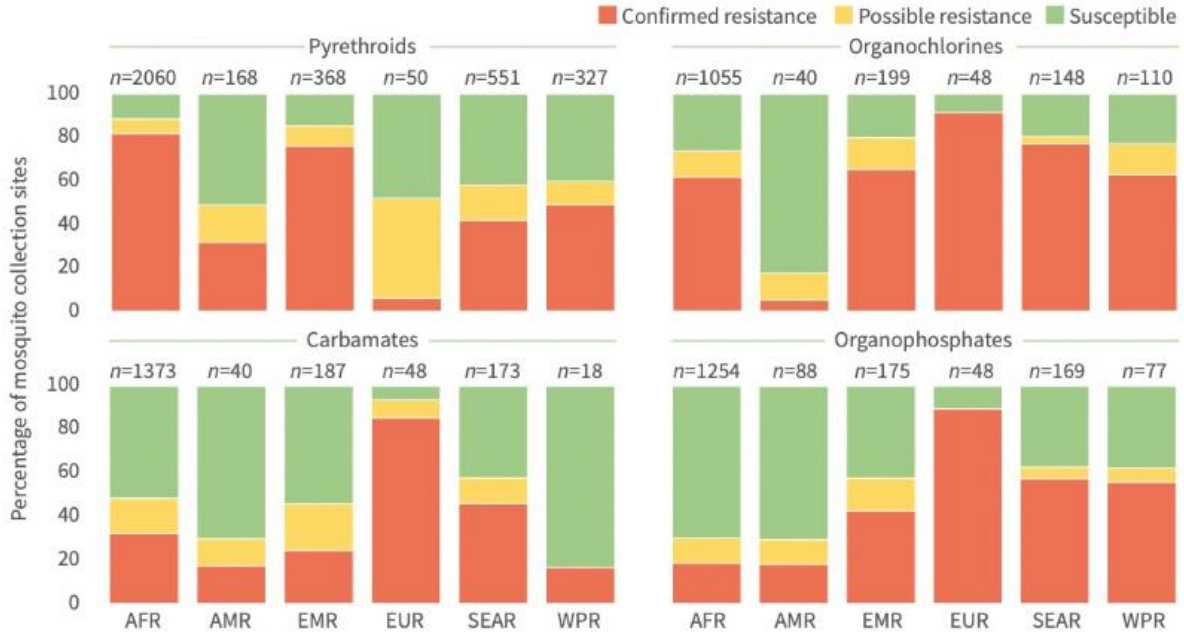


Figure 1. Reported insecticide resistance status of the Anopheles mosquito as a proportion of sites for which monitoring was conducted, by WHO region, 2010–2020, for pyrethroids, organochlorines, carbamates and organophosphates (1).

Malaria is a life-threatening disease caused by *Plasmodium* parasites, which are transmitted to humans through the bites of infected female *Anopheles* mosquitoes (2). Despite significant advances in prevention and treatment, malaria continues to pose a substantial public health challenge, particularly in tropical and subtropical regions of the world. According to the World Health Organization (WHO), there were an estimated 249 million malaria cases and 608,000 malaria-related deaths worldwide in 2022, with most of these cases occurring in sub-Saharan Africa (1). These staggering numbers underscore the persistent burden of malaria, despite considerable efforts in prevention and treatment over the past decades. A critical challenge in malaria control is the evolving resistance of both the malaria parasite and its mosquito vector to current interventions. *Plasmodium falciparum*, the deadliest species of the parasite, has developed resistance to multiple antimalarial drugs, including the highly effective artemisinin-based combination therapies (3). Similarly, *Anopheles* mosquitoes have shown increasing resistance to insecticides (see Figure 1), which are the cornerstone of vector control strategies such as insecticide-treated bed nets (ITNs) and indoor residual spraying (IRS) (4).

Insecticide resistance in mosquito populations poses a severe threat to malaria control and eradication efforts. As mosquitoes develop resistance to commonly used insecticides, the effectiveness of ITNs and IRS diminishes, leading to higher transmission rates and potentially more malaria cases and deaths. This resistance is driven by the widespread use of insecticides in agriculture and public health, which exerts selective pressure on mosquito populations, favouring the survival and reproduction of resistant individuals (5). Compounding this issue, climate change is increasingly recognized as a factor that could accelerate the development and spread of insecticide resistance. Warmer temperatures and altered precipitation patterns can expand the geographical range and breeding season of

Anopheles mosquitoes, leading to larger populations and more frequent exposure to insecticides (6). This increased exposure intensifies selective pressure, potentially speeding up the evolution of resistance. Furthermore, climate change can influence mosquito behaviour and life cycle dynamics, complicating the effectiveness of existing vector control measures.

Addressing insecticide resistance requires a multifaceted approach, including the development of new insecticides with novel modes of action, the implementation of insecticide resistance management strategies, and the exploration of alternative vector control methods such as genetic modification of mosquitoes. A deeper understanding of the molecular, ecological, and evolutionary processes driving these changes is crucial for extending the effective lifespan of current insecticides and speeding up the development of new strategies and tools for vector control.

Limitations To Current Approaches For Inferring Demography

One powerful approach to gaining this understanding is through population genomics, which examines the genetic variation within and between mosquito populations to infer the evolutionary processes at play, such as natural selection, genetic drift, migration, and recombination (7). By leveraging population genomics, researchers can unravel the complex dynamics that contribute to insecticide resistance, enabling the development of more effective and sustainable control measures. A critical component of population genomics is demographic inference, a method used to reconstruct the historical population dynamics of species. Demographic inference allows for the estimation of key parameters such as population size changes, migration patterns, and divergence times between populations (8). By understanding these demographic factors, researchers can better interpret the genetic signals observed in mosquito populations and identify the historical and contemporary forces shaping their evolution. This understanding is particularly important in the context of insecticide resistance, as it helps to reveal how past population bottlenecks, expansions, and gene flow events have influenced the spread of resistance alleles. To fully leverage the insights provided by demographic inference, it is crucial to understand the various methodologies available for reconstructing population histories and their limitations. The following are the broad categories of demographic inference approaches:

Full Likelihood Methods

These methods calculate the probability of observing DNA sequences under specific evolutionary models, typically assuming no recombination within DNA segments but accommodating complex mutation patterns (9). However, their utility is often constrained by the high computational demands involved in evaluating these models, which can make them impractical for large datasets or complex demographic scenarios (10). As a result, full likelihood methods are generally limited to analysing only a few hundred or thousand unlinked DNA segments. Moreover, the assumption of no recombination can oversimplify real genetic landscapes, potentially leading to inaccuracies when recombination plays a significant role.

Approximate Bayesian Computation (ABC) Methods

ABC methods circumvent the need for exact likelihood calculations by focusing on the probability of observing summary statistics that are informative about the model parameters (11). This probability is approximated through simulations that generate summary statistics closely resembling the observed data. The flexibility of this approach makes it applicable to a wide range of problems, including those involving selection and linkage disequilibrium.

Nonetheless, the method's reliance on the careful selection of summary statistics is both a strength and a challenge, as the accuracy of the results heavily depends on these choices (12). Furthermore, ABC can be computationally expensive, especially when applied to complex models or whole-genome data, and its approximate nature may lead to less precise inferences compared to full likelihood approaches.

Site Frequency Spectrum (SFS) Based Methods

These methods analyse the distribution of allele frequencies at various genomic sites within one or multiple populations to infer demographic history (13). The expected SFS can be calculated under simple models or through coalescent-based simulations for more complex scenarios. SFS methods are advantageous because they can scale to whole-genome data. However, they typically overlook linkage disequilibrium, which can result in a loss of critical information about genetic structure (14). Additionally, when applied to large sample sizes and multiple populations, the dimensionality of the SFS can become intractable, slowing down the analysis and limiting its effectiveness. The reliance on simplified models can also mean that SFS methods may not fully capture the complexity of real-world demographic events.

Hidden Markov Models (HMMs)

HMMs are employed to estimate the age of common ancestors of sampled genetic segments by analysing patterns of diversity along the genome (15). These methods are particularly useful for inferring past population sizes and migration rates between populations. However, HMMs typically assume a uniform recombination rate across the genome, which may not reflect the actual recombination landscape and can introduce biases into the inferred demographic parameters. Moreover, the requirement for accurate genome phasing adds another layer of complexity, particularly in non-model organisms (16). Additionally, while HMMs can provide valuable insights into historical demography, translating coalescence rates into population sizes can be problematic, especially when dealing with structured populations or those that exchange migrants.

Haplotype-Based Methods

This approach focuses on the sequence identity of chromosome segments (haplotypes) within or across individuals to infer recent ancestry, admixture events, and population structure (17). They are particularly effective for studying recent demographic events. However, haplotype-based methods may not accurately capture older events, which limits their utility for long-term demographic inference. The methods are also sensitive to population structure and admixture, which can complicate the interpretation of results (18). Furthermore, these methods often assume that haplotypes are neutral, an assumption that may not always hold true, potentially leading to biases in the inferred demographic history.

Deep Learning Challenges Current Limitations

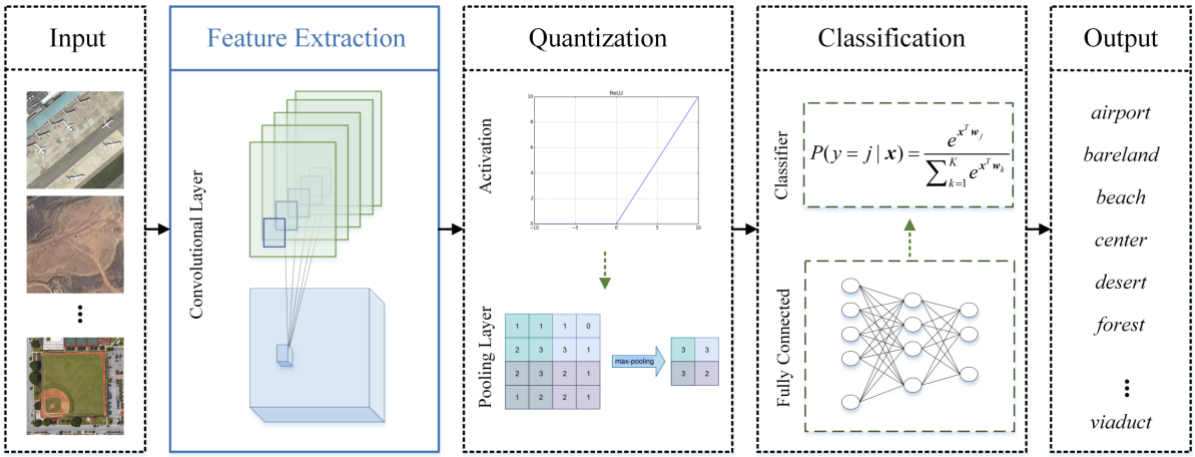
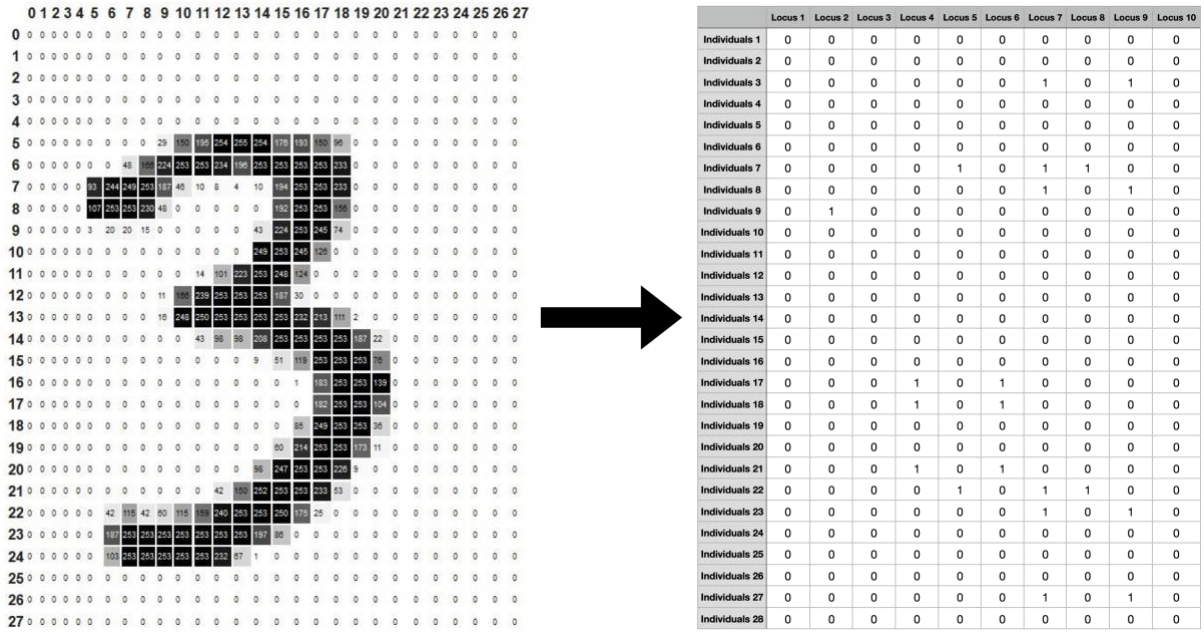


Figure 2. CNN framework consisting of three key components: a feature extraction module, a quantization module, and a tricks module. These modules are repeatedly stacked to construct the deep network architecture. At the end of the network, a classification module is applied to perform the specific classification task. The tricks module is integrated within the feature extraction and quantization modules and is not shown separately in this figure (19).

More recently, deep learning and machine learning approaches have emerged as promising tools for demographic inference. These approaches offer significant advantages over traditional methods by overcoming several key limitations. Unlike full likelihood methods, which are computationally intensive and require explicit likelihood calculations, deep learning models operate in a likelihood-free framework, making them scalable and suitable for large, complex datasets (20). They also eliminate the need for manually selected summary statistics, a central challenge in ABC methods, by learning directly from raw haplotype data, thereby reducing potential biases, and capturing more nuanced patterns (21). In contrast to HMMs, which depend on accurate genome phasing and assume uniform recombination rates, deep learning models can work with unphased data, simplifying analysis and reducing errors (22). While haplotype-based methods are effective for recent demographic events but struggle with older ones, deep learning can model both recent and ancient events, providing a broader temporal understanding of population history (23). Additionally, while SFS methods analyse allele frequency distributions but often do not directly incorporate linkage disequilibrium, deep learning models can integrate full genomic data, including LD information, thereby capturing more comprehensive genetic signals (24). Deep learning models handle entire datasets, including haplotype and genotype data from whole-genome sequencing, without reducing them to summary statistics, allowing for richer analysis. Once trained, these models deliver rapid predictions, making them highly efficient for large-scale or real-time applications. Therefore, deep learning provides a powerful, flexible, and efficient framework for demographic inference, particularly in complex scenarios like studying insecticide resistance in mosquitoes.

Artificial neural networks (ANNs) are a wide branch of deep learning, comprising inputs (referred to as features) and outputs (known as responses), interconnected by nodes across several hidden layers (25). These connections are optimized using data to minimize predictive errors. Once trained, an ANN can accurately predict responses for any new data it receives as input. The ability of this deep learning algorithm to process multiple features simultaneously has significantly advanced the field of image recognition (26)(27). Specifically, convolutional neural networks (CNNs), a specialised type of ANN, have leveraged this capability to handle high-dimensional haplotype data by treating it as “image-like” data, enabling more sophisticated analysis (28).

Figure 2 shows that for a standard CNN, the process begins with a convolutional layer, where small filters are applied across the entire input data, such as an image or a haplotype matrix (19). These filters slide over the input, performing a convolution operation that captures local features such as edges or textures in images, or specific patterns in haplotype data. The result is a feature map that highlights these features wherever they appear in the data. Following the convolutional layer, an activation function, typically a Rectified Linear Unit (ReLU), introduces non-linearity by setting all negative values in the feature map to zero. This allows the network to model complex patterns and interactions in the data. Next, a pooling layer reduces the spatial dimensions of the feature maps, making the computation more efficient and the model more robust to slight translations or distortions in the input (29). This is commonly achieved through max pooling, where only the highest value in each region of the feature map is retained. These convolutional, activation, and pooling layers are often repeated multiple times, each time capturing more complex and abstract features. For example, initial layers might detect simple edges in an image or basic patterns in haplotypes, while deeper layers can identify more sophisticated structures. The final part of a CNN involves one or more fully connected layers, where each neuron is connected to every neuron in the previous layer. These layers combine the features learned by the earlier layers to make a final prediction, such as classifying the input data into categories. The output layer, often using either a Sigmoid or SoftMax function, produces a probability distribution over the possible classes (30).



	Locus 1	Locus 2	Locus 3	Locus 4	Locus 5	Locus 6	Locus 7	Locus 8	Locus 9	Locus 10
Individuals 1	0	0	0	0	0	0	0	0	0	0
Individuals 2	0	0	0	0	0	0	0	0	0	0
Individuals 3	0	0	0	0	0	0	1	0	1	0
Individuals 4	0	0	0	0	0	0	0	0	0	0
Individuals 5	0	0	0	0	0	0	0	0	0	0
Individuals 6	0	0	0	0	0	0	0	0	0	0
Individuals 7	0	0	0	0	1	0	1	1	0	0
Individuals 8	0	0	0	0	0	0	1	0	1	0
Individuals 9	0	1	0	0	0	0	0	0	0	0
Individuals 10	0	0	0	0	0	0	0	0	0	0
Individuals 11	0	0	0	0	0	0	0	0	0	0
Individuals 12	0	0	0	0	0	0	0	0	0	0
Individuals 13	0	0	0	0	0	0	0	0	0	0
Individuals 14	0	0	0	0	0	0	0	0	0	0
Individuals 15	0	0	0	0	0	0	0	0	0	0
Individuals 16	0	0	0	0	0	0	0	0	0	0
Individuals 17	0	0	0	1	0	1	0	0	0	0
Individuals 18	0	0	0	1	0	1	0	0	0	0
Individuals 19	0	0	0	0	0	0	0	0	0	0
Individuals 20	0	0	0	0	0	0	0	0	0	0
Individuals 21	0	0	0	1	0	1	0	0	0	0
Individuals 22	0	0	0	0	1	0	1	1	0	0
Individuals 23	0	0	0	0	0	0	1	0	1	0
Individuals 24	0	0	0	0	0	0	0	0	0	0
Individuals 25	0	0	0	0	0	0	0	0	0	0
Individuals 26	0	0	0	0	0	0	0	0	0	0
Individuals 27	0	0	0	0	0	0	1	0	1	0
Individuals 28	0	0	0	0	0	0	0	0	0	0

Figure 3. Comparison of pixelated image and haplotype data represented as matrices, illustrating how both can be processed by CNNs for pattern recognition (Created with BioRender.com).

The grayscale image in Figure 3, represented as a matrix of pixels with values based on brightness, is analogous to a binary haplotype matrix, where 0s and 1s represent different states. Just as a CNN processes the image to identify patterns, it can similarly analyse the binary matrix to detect patterns in genetic data, treating the haplotypes as "image-like" data. Once trained, CNNs can process entire datasets, including both haplotype and genotype data from whole-genome sequencing, without needing to reduce them to summary statistics, thus enabling richer and more detailed analysis. The efficiency of CNNs, combined with their ability to deliver rapid predictions once trained, makes them highly effective for large-scale applications, such as studying complex genetic traits or the spread of insecticide resistance in mosquito populations.

Inferring Migration Levels

The *Anopheles* gambiae 1000 Genomes Project (Ag1000G Project) is a large-scale genomics initiative aimed at understanding the genetic variation in populations of *A. gambiae* (31). The project is analogous to the 1000 Genomes Project in humans, but it focuses on mosquito populations across Africa, where malaria is most prevalent. It involves sequencing the genomes of thousands of *A. gambiae* and other closely related species from different regions across Africa. This is important because Ag1000G has used this genetic data to provide information of demographic parameters of the evolutionary history of the *Anopheles* mosquito, such as effective population sizes, mutation rates and divergence times. Currently, there is a need to understand the evolutionary processes of the Anopheles mosquitoes in Africa to combat the effect of increasing insecticide resistance. To this point, population genomics uses demographic inference to understand gene flow between sub-species of the Anopheles mosquito. Current approaches to inferring demography have limitations but recent advancement in deep learning, particularly in the form of CNNs, potentially offer a more efficient, accurate and flexible approach to demographic inference. The aim of this project is to leverage the demographic parameters provided by The Ag1000G Project to classify migration levels between two closely related *Anopheles* species, *A. gambiae* and *A. coluzzii*, as low, medium, or high using CNN technology.

Materials and Methods

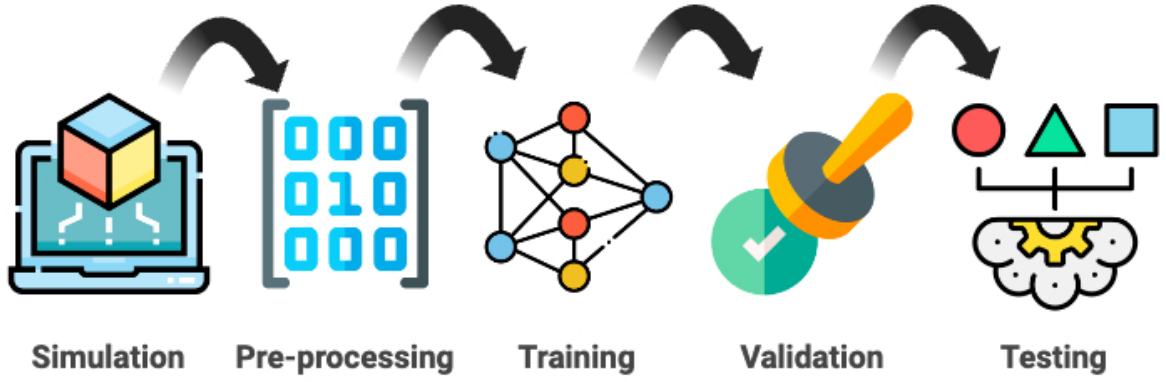


Figure 4. Schematic of overview of CNN workflow. The process begins with simulating haplotype data using msprime. The simulated data then undergoes pre-processing, which includes truncation and data augmentation. In the training phase, the pre-processed data is labelled, and the convolutional neural network model is designed and trained. During training, validation is performed simultaneously to provide feedback and guide the model's learning. Once the model is fully trained and validated, it proceeds to the testing phase, where its predictive performance is evaluated (Created with BioRender.com).

Simulation

Simulations were conducted using msprime, a software tool designed to simulate the genetic ancestry of populations based on the coalescent model (32). This approach allows for the generation of realistic haplotype data by modelling the genealogical history of a sampled set of individuals back to their most recent common ancestors, using user-defined parameters. For the CNN, 300 simulations were performed (100 per migration rate). The simulations were based on a two-population evolutionary model, as depicted in Figure 5, which was adapted from the pg-gan framework to reflect gene flow between *A. gambiae* and *A. coluzzi* (33). The model assumes instantaneous population growth from a shared ancestral population to the daughter populations, with a single pulse migration event occurring at Tsplit / 2, which can proceed in either direction. Although the recombination rate and mutation rate parameters are not depicted in Figure 5, their values, along with other variable parameters such as migration rates, are detailed in Table S1. All parameter values were derived from The Ag1000G Project data (31). Simulation process for exploratory analysis and investigation was nearly identical. The only difference lied in the output where the tree sequence data was converted into FST summary statistics instead of haplotype data. In contrast to the 300 simulations carried out for CNNs, the number of simulations for FST were 3000 (1000 per class).

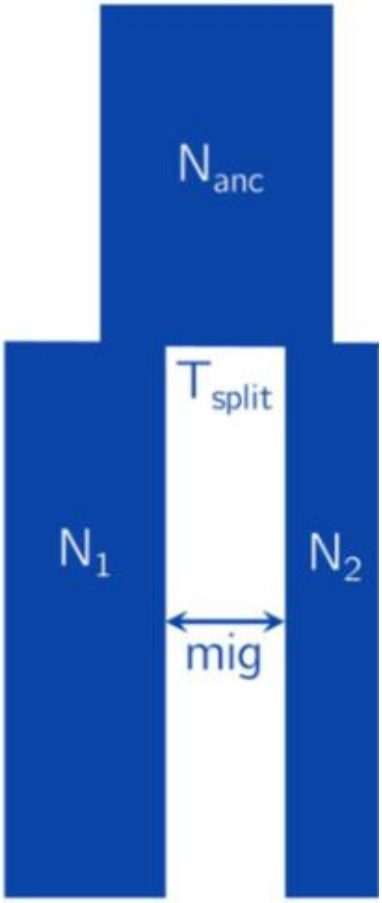


Figure 5. Two-population isolation with migration model from pg-gan (33).

Pre-processing

This step took place after haplotypes (or FST values) were simulated. The goal of this step was to reshape the data to be compatible with the input layer of the CNN. However, the difficulty lies in performing this transformation in a way that keep the integrity of the data. Standard to haplotype data, msprime simulated haplotypes as rows of individuals and columns of segregated sites or loci. Alleles were label encoded as either 0 or 1 for major or minor allele, respectively. However, due variable recombination and mutation of each simulation, the number of loci generate i.e., the number of columns varied. This meant that the haplotype length for individuals was not the same between simulations. This can either be dealt with padding empty spaces with arbitrary values like zeros (34). However, this created concerns about confounding the binary nature of the haplotype data and introducing unnecessary noise into the data. Hence, an alternative approach was employed by truncating the sequences to the minimum sequence length, which was 4882 haplotypes for the simulated training sequences (28). Sequences were truncated at the tail-end. Next, data augmentation was applied by mirroring the 3D haplotype array horizontally, vertically, and diagonally (35). This was done because data augmentation led to drastic improvements in training and validation accuracy (see Figure S1). The truncated sequences and the augmented data were then concatenated into a 3D array of dimensions as simulations x number of individuals x segregating sites. In contrast, pre-processing for exploratory analysis and SVM was not as intensive, only involving the removal of negative FST values. FST values were removed instead being set to zero because they greatly outnumbered non-negative FST values, skewing the data. For exploratory analysis, the processed FST values were plotted in R.

Training & Testing

To prepare the 3D haplotype array for input into the CNN, it was necessary to reshape the data into a 4D array, as 2D CNNs are designed to process image-like data with dimensions such as number of images, height, width, and number of channels. For instance, in the case of image data, typical dimensions for a batch might be 50,000 images with 32 pixels in height, 32 pixels in width, and 3 channels for coloured images (26). Similarly, for haplotype data, an image can be analogized to a haplotype matrix, as illustrated in Figure 3. The initial 3D haplotype data was reshaped into a 4D array with dimensions of 300 simulations × 100 individuals × 4882 SNPs × 1 channel (where the single channel corresponds to the binary nature of the data). With data augmentation, the batch size increased to 1,200 simulations (300 simulations × 4 augmentations). Pairwise labels were created to associate simulations to their corresponding migration level. The reshaped 4D haplotype array was subsequently converted into a TensorFlow tensor for processing. The CNN model was implemented and trained using Keras with TensorFlow as the backend. The input haplotype data had dimensions of 1,200 simulations × 100 individuals × 4882 SNPs × 1 channel. The model architecture comprised two 2D convolutional layers with 64 and 32 units, respectively, and kernel sizes of 3 × 3. Each convolutional layer was followed by a max-pooling layer with a 3 × 3 kernel size. ReLU activation was applied in each layer, and a mini-batch size of 16 was used during training. The output of the final fully connected layer was passed through a SoftMax function, with the unit size set to 3, corresponding to the three classes of migration rates (low, medium, and high). The model was optimised using the Adam algorithm, and the loss was calculated using the cross-entropy function. During training, 20% of the data was allocated as a validation set for each epoch. For testing, a new unseen batch of data comprising 100 simulations was generated, with the migration rates matching those used in the training phase (see Table S1 for details). The trained model's performance was evaluated on this test dataset to assess its predictive accuracy across the three migration rate categories.

Regarding SVM, Scikit-learn was used to implement the model and the processed FST data was labelled according to the corresponding migration class. However, the removal of negative FST values reduced the sample size from 3000 to 1485, which introduced class imbalance in a random manner. Of the 1485 FST values, 70% (1039 samples) were allocated for training the SVM model, while the remaining 30% (446 samples) were reserved for testing. To ensure the robustness and consistency of the results, SVM training and testing were performed across 1000 iterations of bootstrap validation (see Table S2).

Evaluation Metrics & Statistical Testing

Several evaluation metrics and statistical tests were carried out to ensure the robustness of the results. Statistical tests included permutation testing for SVM significance testing and binomial testing for CNN outcome significance. Both techniques involve calculating a p-value and setting a threshold of 0.05 to determine the significance of the models' accuracies. Confusion matrices were determined by comparing true values to predicted values. The following equations indicate how precision, recall, F1-score, and macro-averaging was calculated.

$$Precision = \frac{True\ Positives}{True\ Positivies + False\ Positives}$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

$$Macro\ Precision = \frac{1}{n}\sum_{i=1}^{n} Precision_i$$

where n is the number of classes and $Precision_i$ is the precision of class i.

Code Availability

The code for this project involving simulations, CNNs, SVMs and evaluation metrics and testing and training data are available on GitHub in the following repository: https://github.com/jeremiahmushtaq/MSc-Research-Project

Results

Exploratory Analysis

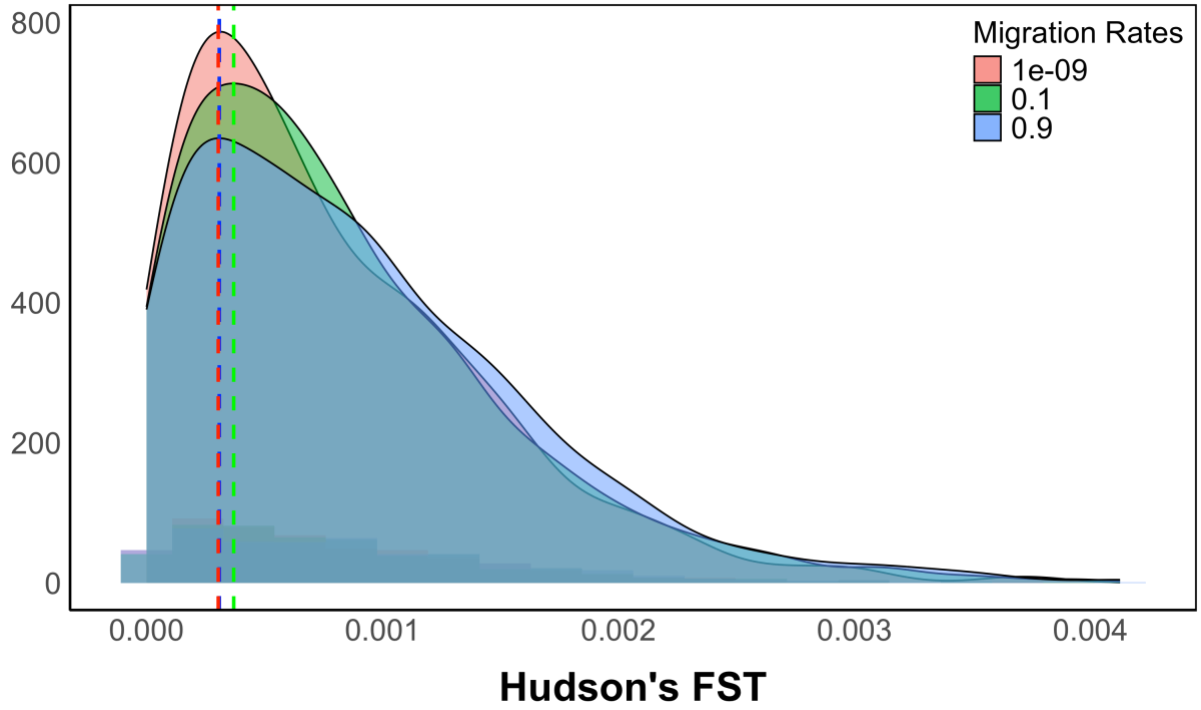


Figure 6. Distribution of FST statistics simulated using msprime for low (1×10^{-9}, red), medium (0.1, green), and high (0.9, blue) migration rates, based on 3,000 simulations. Non-zero FST values have been filtered out. The density plots represent the distribution of FST values for each migration rate category, with vertical dashed lines indicating the modal FST values: approximately 3.09×10^{-4} for low migration (red), 3.70×10^{-4} for medium migration (green), and 3.04×10^{-4} for high migration (blue).

FST is a widely used summary statistic for quantifying genetic differentiation among subpopulations, based on measures of heterozygosity (36). Specifically, it compares the genetic variance within subpopulations (intra-population variance) to the total genetic variance across the entire population, encompassing all subpopulations. This statistic indirectly measures gene flow or migration between populations, providing insights into the extent of genetic exchange. In this study, FST was employed to investigate the effects of varying migration levels by using genetically simulated data produced with msprime (Table S1). The simulations were based on a two-population evolutionary model designed to approximate the genetic dynamics between the *A. gambiae* and *A. coluzzii* species. The resulting distribution of FST values is presented in Figure 6, which illustrates the outcomes of 3,000 simulations conducted under three distinct migration rates: low (1 x 10-9), medium (0.1), and high (0.9). These simulations aimed to explore how different levels of migration influence genetic differentiation as measured by FST.

The graph shows three overlapping density plots corresponding to each migration rate, colour-coded in red (low migration), green (medium migration), and blue (high migration). Each curve represents the frequency distribution of FST values obtained from the simulations. Vertical dotted lines indicate the modal FST values for each migration rate, with the modal FST values for low, medium, and high migration rates occurring at approximately 3.09 x 10-4, 3.70 x 10-4, and 3.04 x 10-4, respectively. Despite the variation in migration

rates, the distribution of FST values largely overlaps across the three scenarios, suggesting that the differences in genetic differentiation between the low, medium, and high migration rates are not substantial when considering the modal FST values alone. This overlap indicates that, in this simulation context, changes in migration rates produce only minor shifts in the central tendency of FST values.

However, a notable difference is observed in the frequency of FST values within the range of 0 to 0.001. The low migration rate yields a higher frequency of FST values in this range compared to the medium and high migration rates, which produce progressively fewer FST values as migration increases. Specifically, there are 340, 337, and 312 FST values between 0 and 0.001 for the low, medium, and high migration rates, respectively. This pattern aligns with theoretical expectations, where reduced gene flow between populations—characteristic of low migration—leads to greater genetic differentiation. Consequently, lower migration rates result in more frequent occurrences of low FST values, reflecting the increased genetic variance between subpopulations due to limited interbreeding.

These findings suggest that while migration rate influences genetic differentiation, the effect is more pronounced in the distribution of lower FST values rather than in the modal FST value. In particular, low migration rates lead to higher genetic differentiation, as evidenced by the greater number of low FST values, which underscores the role of migration in maintaining genetic connectivity and reducing population divergence. These results provide valuable insights into the genetic dynamics between populations, with implications for understanding species divergence and the evolutionary consequences of gene flow.

SVM Performance Evaluation

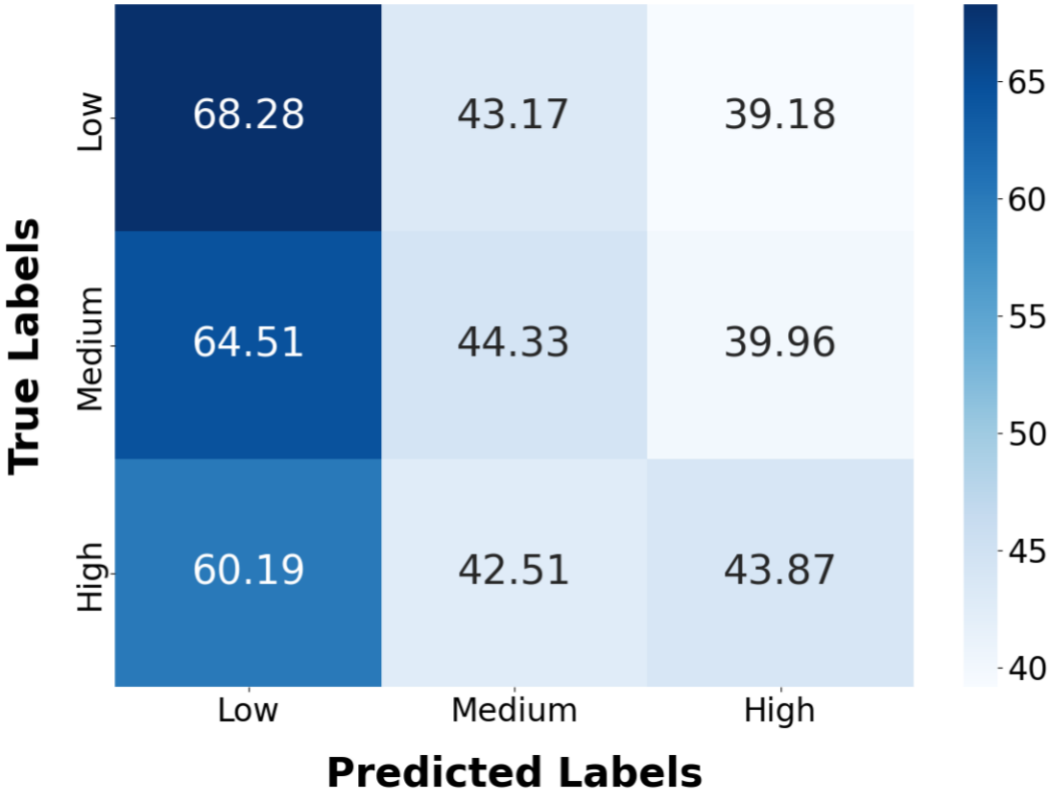


Figure 7. Confusion matrix showing the classification performance of the SVM model trained to predict migration levels—low (1×10^{-9}), medium (0.1), and high (0.9)—based on 3,000 FST values simulated using msprime (1,000 per class). The dataset was adjusted class imbalance by removing negative FST values. The SVM was trained on 70% of the data and tested on the remaining 30%. The confusion matrix highlights the distribution of correctly and incorrectly classified instances across the three migration categories, revealing that while the model was moderately successful in predicting low migration, it struggled significantly with medium and high migration, leading to substantial misclassification.

This study aimed to develop a deep-learning algorithm to classify migration levels in Anopheles mosquito populations in Africa. Initial analysis of simulated FST data (Figure 6) revealed patterns across migration rates, suggesting the potential for machine learning-based prediction. To assess the feasibility of this task with a simpler model, an SVM was used to classify FST values generated from migration rates of 1 x 10^{-9} (low), 0.1 (medium), and 0.9 (high). A dataset of 3,000 FST values (1,000 per class) was split into 70% for training and 30% for testing.

The SVM's performance, as shown in the confusion matrix (Figure 7), highlights the distribution of predictions across the three migration categories. The results indicate that the classifier demonstrated reasonable success in identifying cases of low migration, correctly classifying 68.28% of the instances. However, the model misclassified a notable proportion of low migration cases as medium (43.17%) and high migration (39.18%). This suggests some level of overlap or ambiguity in the FST data that challenges the model's ability to consistently distinguish low migration from the other categories. For medium migration, the classifier's performance was less satisfactory, with only 44.33% of medium migration cases being correctly identified. The majority of medium migration cases were misclassified as low migration (64.51%), while a smaller portion was incorrectly labeled as high migration (39.96%). These results reflect a significant challenge in accurately predicting medium migration rates, possibly due to the FST values not providing sufficiently distinct patterns that differentiate medium migration from low or high migration levels. The classifier

struggled most with cases of high migration, correctly identifying only 43.87% of instances. A substantial portion of high migration cases was misclassified, with 60.19% being incorrectly labeled as low migration and 42.51% as medium migration. This indicates that the SVM had considerable difficulty in distinguishing high migration from the other two categories, which may point to a lack of clear separation in the FST data for this class.

To ensure the robustness of these findings, bootstrap validation was employed (Table S2). Bootstrap validation is a resampling technique that involves repeatedly drawing samples from the original dataset, with replacement, to create multiple "bootstrap samples" (37). The model is then trained and tested on these samples over a large number of iterations. The primary advantage of bootstrap validation is that it provides an estimate of the model's performance that is less likely to be biased by the specific characteristics of a single training set. This method was crucial in obtaining a more reliable estimate of the model's accuracy across different potential variations of the dataset. Bootstrap analysis over 1,000 iterations revealed that the overall average accuracy of the model across all classes was 35.06% ± 2.41%. This relatively low accuracy suggests that the model's predictive power is limited in predicting migration levels. In addition to bootstrap validation, permutation testing was conducted to assess the statistical significance of the SVM classifier's performance (Table S2). Permutation testing is a non-parametric method used to determine whether the observed results are likely to have occurred by chance (38). In this process, the labels are randomly shuffled, and the model's accuracy is recalculated for these permuted datasets. This procedure is repeated many times to generate a distribution of accuracies that could be expected under the null hypothesis i.e. where there is no actual relationship between the FST data and migration levels. The p-value derived from this test indicates the proportion of permuted datasets that achieved an accuracy equal to or greater than that of the original model. In this study, permutation testing yielded a p-value of 0.47, which is above the significance threshold of 0.05. This suggests that the SVM classifier's performance is not significantly better than what could be expected by random chance, further questioning the reliability of the SVM in this context.

Thus, while the SVM classifier demonstrated some capability in predicting low migration levels, its overall performance was suboptimal, particularly for medium and high migration classes. The significant rates of misclassification, coupled with the low bootstrap accuracy and non-significant p-value from the permutation test, indicate that the model's ability to classify migration levels from the given FST data is inadequate. These findings highlight the need for either a more complex model to enhance classification accuracy.

CNN Performance Evaluation

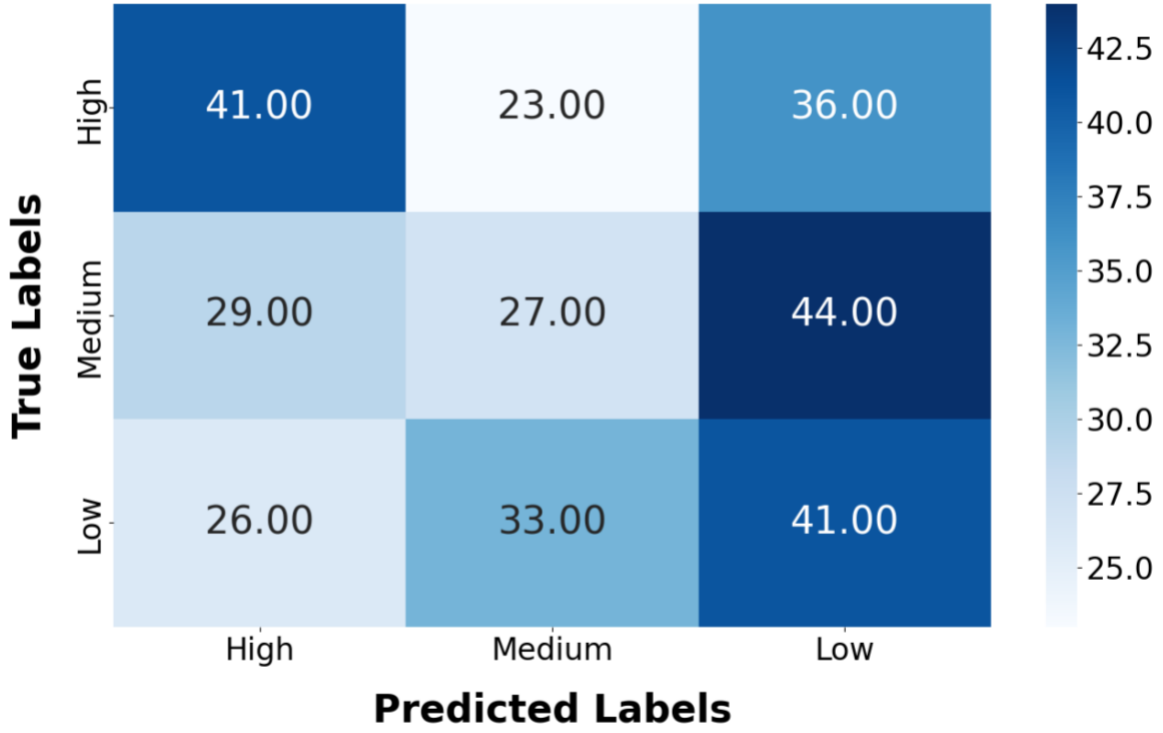


Figure 8. Confusion matrix depicting the performance of the CNN model trained to classify binary haplotype data, simulated using msprime, according to low (1×10^{-9}), medium (0.1), and high (0.9) migration rates. The dataset consisted of 300 simulations in total, with 100 simulations per class, ensuring no class imbalance. The matrix illustrates the distribution of predicted versus true migration levels, revealing that while the CNN achieved moderate accuracy across all classes, there were notable instances of misclassification, particularly between the low and high migration categories.

The SVM model initially used for classifying migration levels achieved an accuracy of 35.06% ± 2.41%. To enhance this accuracy, a more complex model was developed by employing a 2D CNN. This approach treated simulations as analogous to image-like data, which allowed the CNN to process the information effectively. The CNN requires input data structured with specific dimensions: batch size, height, width, and the number of channels. To align the simulated data with this format, haplotype data was used instead of FST data. Each simulation consisted of a dataset with dimensions of 100 individuals by 4882 haplotypes. To generate the necessary input data for the model, msprime was employed to conduct 300 simulations, with 100 simulations corresponding to each of the three migration rate classes. Like the SVM model, the migration rate classes were set at 1 x 10^{-9} (low), 0.1 (medium), and 0.9 (high). These simulations formed the input sample size used to train the CNN model.

The performance of the CNN in predicting levels of migration based on three migration rate classes was evaluated using a variety of metrics. A confusion matrix (Figure 8) was generated to visualise the model's accuracy in distinguishing between these classes. The matrix revealed that the model correctly predicted the High migration rate in 41.00% of cases, with 23.00% misclassified as Medium and 36.00% as Low. For the Medium migration rate, the model correctly identified only 27.00% of cases, with significant misclassifications into High (29.00%) and Low (44.00%) categories. The Low migration rate was correctly predicted 41.00% of the time, but there were notable misclassifications into High (26.00%) and Medium (33.00%) categories. These results suggest that while the model has some capability

to differentiate between migration rates, there is substantial overlap, particularly in the Medium migration class, which is frequently confused with the other two classes.

Further evaluation using precision, recall, and F1-scores provided additional insights into the model's performance (Table S3). Precision, defined as the ratio of true positive predictions to the total number of positive predictions made by the model, was 0.43 for the High migration rate, indicating a relatively balanced performance (39). Recall, which measures the ratio of true positives to the total number of actual positives, was 0.41 for the High migration rate. The F1-score, the harmonic mean of precision and recall, was 0.42 for the High migration rate. However, the Medium migration rate presented the greatest challenge, with a precision of 0.33, a recall of 0.27, and an F1-score of 0.30, reflecting the confusion observed in the confusion matrix. The Low migration rate demonstrated slightly better performance, with a precision of 0.34, a recall of 0.41, and an F1-score of 0.37, suggesting that the model is better at identifying Low migration rates when they occur but also tends to misclassify other rates as Low. The macro-averaged precision, recall, and F1-score were calculated to provide an overall measure of the model's performance across all classes (Table S3). Macro-averaging involves calculating the precision, recall, and F1-score for each class independently and then averaging these values (40). This method treats each class equally, regardless of the number of instances in each class, and is particularly useful when the classes are imbalanced. In this case, the macro-averaged precision, recall, and F1-score were all 0.36, indicating moderate overall performance across the migration rate classes.

To statistically assess the model's performance, binomial testing was conducted (Table S4). Binomial testing is a statistical method used to determine whether the observed success rate differs significantly from what would be expected by chance (41). The results showed that the model made 109 correct predictions out of 300, resulting in an overall accuracy of 36.33%. This accuracy is marginally better than the baseline accuracy of 33.33%, which would be expected from random guessing. The p-value obtained from the binomial test was 0.15, which exceeds the conventional significance threshold of 0.05. This suggests that the model's accuracy is not statistically significantly better than what would be expected by random chance, indicating that the CNN's classification of migration rates is not significantly superior to random guessing. Binomial testing was essential in this context to establish whether the model's performance was genuinely better than a baseline or whether it could be attributed to random variation.

In summary, while the CNN classifier exhibits some ability to differentiate between High, Medium, and Low migration rates, its performance, particularly for the Medium migration class, is limited. The lack of statistical significance and the moderate macro-averaged metrics further underscore the need for model refinement, potentially through the exploration of more complex architectures or the incorporation of additional data to enhance the classifier's predictive accuracy.

Discussion

Why do we need CNNs in demographic inference?

There are several computational approaches used in population genomics to infer demographic events from whole genome sequencing data. Traditional approaches to make evolutionary inferences aim to condense genetic information from various locations on the genome to make accurate inferences. Likelihood-based methods such as MLE use statistical models to estimate the likelihood of a set of evolutionary scenarios based on the observed genetic data (42). Probabilistic graphical models model dependencies among data points (43). Summary statistics reduce genomic data to a set of statistics such as FST that summarises patterns of genetic variation (44). While these traditional approaches are quite different, they all make use of population genomic theory to connect the various features of a genetic dataset to the underlying evolutionary process. However, in doing so, the complexity of the genetic data is simplified by focusing on a limited number of features or summary statistics due the assumption that these can effectively represent the foundational evolutionary processes. Another major challenge is that relying on a few selected features or summary statistics may compromise important information embedded in the data. The connection between these limited features and evolutionary processes is established through theoretically derived estimated or closed-form likelihood functions, which may not capture the full complexity of the data.

What are the key findings and does literature agree with the findings?

This study aimed to develop an alternative approach that reframes demographic inference as an image recognition problem, where the image is an alignment of population genetic sequences that is fed directly into the input layer of a CNN. This project ventured to test whether the deep learning technique could meet the challenge of using the entirety of the dataset, potentially offering a more nuanced and accurate approach to demographic inference. The results have shown that CNNs do not perform well in inferring levels of migration in the two-population evolutionary model used in this study. However, the findings of this study are not consistent with previous studies that demonstrate the potential for successful application of CNNs in evolutionary inference. For example, a study attempted to train a CNN to estimate the demographic parameters of a model, particularly looking at instantaneous effective population size changes (20). Inferences were made of three population sizes and of the two times the size changes occurred. The inferred values were then compared to the real-value parameters. The median root mean square error (RMSE) was 0.54 but several networks achieved an RMSE of <0.5 while the best score was 0.43. The study stated that the prediction accuracy was fairly encouraging considering the limitations of the study. Another study reports on a program called, ImaGene, that used CNNs to predict whether a genomic region is under neutral, weak-to-moderate, or strong natural selection pressures (34). This particular task achieved an accuracy of 83.84%, indicating that there is a strong potential of using CNNs for demographic inference. There are several other studies that suggest CNNs have the ability to provide more accurate inferences in population demographics (28,45,46).

Why was the prediction accuracy of the CNN low?

There is a disparity in the key findings of this study and what the literature reports. This study reports that CNNs are not able to infer demographic parameters but the evidence suggests that there is some potential for the applicability of CNNs in this field. As such, it is important

to identify the limitations in the methodology that might have led to a low prediction accuracy for the model developed in this study.

Rare variants

Comparing the methodology used in this study with several other studies, it is important to note that other investigations often remove rare variants from the simulated haplotype matrix when pre-processing the input data. Rare variants are alleles that occur at very low frequencies in the population (47). They become numerous in large populations because the likelihood of mutations is proportional to the population size. This is because larger populations tend to harbour more genetic variation simply because there are more individuals in which mutations can occur. However, this means that larger populations experience weaker genetic drift, meaning alleles, especially rare ones, are maintained in the population for longer periods. Hence, simulating populations with very large sizes introduces a significant number of rare variants. These rare variants contribute to the overall genetic variation, but their rarity adds a lot of complexity and "noise" to the data. This means the classification model might struggle to distinguish between meaningful patterns and the noise introduced by the rare variants (48). This is important because the effective population sizes used in this study for simulating haplotype data were $N1$ = 158,124,480 individuals, $N2$ = 54,259,530 individuals, N_{anc} = 7,148,911 individuals (see Table S1). When these population sizes are compared with effective population sizes used in other studies, it becomes clear that even though they use smaller population sizes than the ones used in this project, they still filter out low frequency alleles (34)(28). This indicates that this study could have greatly benefited from pre-processing rare variants to reduce noise in the simulated haplotype data used for training and testing the model, which could've potentially increased the prediction accuracy of the model.

Training data

Another notable difference lies within the amount of training data used to train CNNs on patterns in the data. A study investigated the application of CNNs in four primary tasks achieving impressive results (20). For each task, the number of simulations used were as follows: 100K for inferring population size histories; 239K for detecting selective sweeps and discriminating between modes of selection; 156K for recombination rate estimation; and 238K for introgression detection. The study shows that it is important to use a large number of simulations to train, validate and test the classifier. This is because with each iteration the model learns patterns within the data that may signal towards a particular class. Another study reported on the effect of different training dataset sizes and learning rates on the accuracy of CNNs in predicting either neutral or natural selection (49). The study tested four datasets of sizes 140K, 110K, 70K, 12K and found that the prediction accuracies were very similar upon at approx. 99% when testing with simulated data. This suggests that perhaps simulation batch sizes of the order 10^6 are excessive but there is no reporting of dataset sizes less than 12K. This project used a sample size of 300 simulations to train the classifier, which dwarfs in comparison to even the smallest dataset used in the aforementioned study. There are also other studies that use large simulation numbers to train their models (34)(28).

Exchangeability

In population genetic data, the order of individuals (rows in the input haplotype matrix) is arbitrary because these individuals are typically random samples from the population. There is no natural or meaningful sequence to this order that reflects their genetic relationships or spatial positioning. This can be an issue for CNNs because they rely heavily on spatial information i.e., they are designed to detect patterns based on the relative positioning of data

points. If the input data's order is arbitrary, this could negatively affect the CNN's performance because the network might interpret the arbitrary order as carrying some meaningful spatial information, leading to inaccurate or inconsistent results. One way to address this issue is by sorting the individuals in a "biologically meaningful" way, such as by genetic similarity or allele frequency (20)(34). This reordering helps to create a structure in the data that the CNN can more effectively learn from. There are other more complex methods of dealing with the issue of exchangeability e.g., deploying an exchangeable neural network architecture (50). This type of network includes convolutional layers combined with a permutation-invariant function. The permutation-invariant function (e.g., a mean operation) ensures that the network's output is insensitive to the order of the input data, effectively addressing the exchangeability issue by making the network robust to the arbitrary ordering of individuals. The CNN model developed in this project may have benefitted from incorporating an aspect of exchangeability into the input data to add a level of biological spatial awareness, which may have helped increase the predictive power of the model.

What are the implications to population genomic researchers?

The results of this project found that it was not possible to infer migration levels using CNNs. However, reviewing population genome literature about the application of CNNs for demographic inference tasks suggests that there are several steps that can be taken to ensure that the input data is pre-processed to optimise for better pattern recognition. Practitioners looking to design, develop and research CNNs for evolutionary inference should keep these steps in mind.

Firstly, it is important to remove rare variants from the input haplotype matrix (34). This is because rare variants lead to noisy data that can obscure signals of important patterns. Rare variants can be removed by calculating the minor allele frequency (MAF) of each locus (column) in the matrix (47). Then, choose a threshold below which an allele is considered rare. Common thresholds in population genetics are 1% (0.01) or 5% (0.05). Finally, remove columns of the matrix that have a MAF below the chosen threshold. This step reduces noise in the input data.

Secondly, researchers should use a relatively large number of simulations to train their models (49). This is necessary because CNNs learn patterns in the input data so a lack of training data can result in the CNN not having enough feature to inform predictions. Literature indicates that there is great variety in the number of simulations used for training, but most studies use a training size greater than 10K simulations (20)(34). Another approach called “simulation-on-the-fly” harnesses the power of simulators to generate infinite data. In this approach, fresh data is simulated at each training step, ensuring better-calibrated posteriors, and decreased risk of overfitting (50). Therefore, practitioners are advised to use simulation sizes of at least 10K or deploy a technique such as “simulation-on-the-fly” to continuously generate fresh training data until the model is adequately trained. However, it should be noted that it is computationally intensive to simulate and train large batches of in this order of magnitude (20). Thankfully, CNNs training can be divided amongst multiple GPUs on cloud-based platforms relatively cheaply to fasten the training process. This can be couple with rescaling techniques where population parameters and mutation rates can be rescaled to reduce computation time (51). For example, the mutation rate can be rescaled or adjusted to reduce the number of mutations that appear in the data. By artificially lowering the mutation rate, the number of polymorphic sites is decreased. This reduces the data complexity and the amount of information the model needs to process, speeding up computation without drastically affecting the model's ability to learn relevant patterns.

Finally, taking the issue of exchangeability is important to produce robust training data for the model (50). This is because overtime researchers have identified that CNNs rely on spatial relationships to detect patterns in the training data. This implies that it is important to sort training data in a biologically meaningful manner. Here, researchers have several options such as using sorting data by allele frequency, using PCA to cluster data based on genetic similarity or deploying an exchangeable neural network architecture. Either way, the aim is to sort the individuals/chromosomes of individuals so the model can recognise patterns between similar rows in the input matrix.

What are some other limitations to the methodology?

A major limitation occurred in the pre-processing of FST data for exploratory analysis. As briefly mentioned in the materials and methods section, there were a great number of FST values simulated using msprime that were negative. This posed an issue because the FST statistic ranges from 0 to 1, where 0 indicates a population in complete panmixia and 1 indicates no shared genetic variation. Therefore, negative FST values do not fit within the typical interpretation of the statistic. Negative FSTs can be caused by random fluctuations in allele frequencies, which in the case of this study is entirely possible do the random noise generated by unfiltered rare variants. Often, researchers deal with non-positive FSTs by setting them to zero but in this study approximately half of the FST values were negative. Therefore, setting these values to zero resulted in a massive bias. Therefore, all non-positive FST values were removed, however it was not considered that this would create a class imbalance in downstream analysis. Removing all negative FSTs dropped the input data to 50% of its original size of which 70% was allocated to training. This left only 446 samples of FST values for testing. Even within these samples, a class imbalance was found of 150.63, 148.77 and 146.57, for low medium and high migration classes. This potentially confounded the data to produced misleading predictions that indicate classes with more FST values have a higher accuracy. To this point, macro-averaging could have been employed to further investigate model predictions within the context of class imbalance.

On the other hand, further limitations in the CNN approach including truncating haplotype sequences that could potentially result in loss of genetic diversity and not performing any tests with real data from The Ag1000G project. Although, given the accuracy of the model with simulated data, it would be unlikely that any meaningful outcomes would have resulted from testing on real Anopheles haplotypes. Finally, while this study focused more on implementing a CNN for migration rate inference, the two-population evolutionary model used to simulate haplotypes and FSTs is simplistic due to the assumptions it makes about instantaneous growth in population size. A more detailed model could be employed to further replicate real-life population growth patterns.

Conclusion & Future Work

This study explored the use of CNNs to classify migration levels between *A. gambiae* and *A. coluzzii* based on simulated genetic data. While CNNs have shown promise in other applications, our findings indicate that the model struggled to accurately infer migration levels in this specific context. The moderate accuracy and frequent misclassifications suggest that the CNN faced difficulties in recognising the complex genetic patterns associated with different migration scenarios. Several factors likely contributed to the model's limitations. The inclusion of rare variants in the dataset may have introduced noise, obscuring the patterns the model needed to identify. Moreover, the relatively small size of the training dataset may not have provided enough examples for the CNN to effectively learn the relevant features. Additionally, the arbitrary ordering of individuals in the input data could have confused the model, which relies on spatial information to detect patterns.

Looking ahead, there are several avenues for future research that could enhance the predictive power of CNNs in demographic inference. First, improving data pre-processing by filtering out rare variants could help reduce noise and allow the model to focus on more informative patterns within the genetic data. Second, increasing the size and diversity of the training dataset could improve the model's ability to generalize and detect subtle patterns. Techniques such as "simulation-on-the-fly," which continuously generates fresh data during training, could provide a richer learning environment for the CNN. To this point, it might be interesting to build on the standard CNN architecture and use a Generative Adversarial Neural Network framework. GANs pit two CNNs against each other in a zero-sum game to to generate synthetic data. There are already some interesting applications of this in populations genomics using data from The Human Genome Project (33).

Another important consideration is addressing the issue of exchangeability in the input data. Implementing methods to reorder individuals based on genetic similarity or exploring neural network architectures that are robust to arbitrary input orders, could enhance the model's performance. Additionally, while CNNs have been the focus of this study, exploring alternative deep learning architectures, such as recurrent neural networks (RNNs) or transformer models, may yield better results in capturing the temporal and sequential nature of genetic data. Finally, integrating additional data sources, such as environmental factors or phenotypic data, could provide a more comprehensive understanding of the forces driving migration and other demographic events. Multimodal models that combine these diverse data sources could offer new insights and improve the accuracy of demographic inferences.

By addressing these areas in future research, one can better leverage the potential of deep learning techniques to understand complex evolutionary processes. This, in turn, could contribute to more effective strategies for malaria control and the management of insecticide resistance, ultimately aiding global health efforts.

References

1. World malaria report 2023. Geneva: World Health Organization; 2023.

2. Poespoprodjo JR, Douglas NM, Ansong D, Kho S, Anstey NM. Malaria. The Lancet. 2023 Dec;402(10419):2328–45.

3. Dondorp AM, Nosten F, Yi P, Das D, Phyo AP, Tarning J, et al. Artemisinin Resistance in *Plasmodium falciparum* Malaria. N Engl J Med. 2009 Jul 30;361(5):455–67.

4. Suh PF, Elanga-Ndille E, Tchouakui M, Sandeu MM, Tagne D, Wondji C, et al. Impact of insecticide resistance on malaria vector competence: a literature review. Malar J. 2023 Jan 17;22(1):19.

5. Reid MC, McKenzie FE. The contribution of agricultural insecticide use to increasing insecticide resistance in African malaria vectors. Malar J. 2016 Dec;15(1):107.

6. Agyekum TP, Arko-Mensah J, Botwe PK, Hogarh JN, Issah I, Dadzie SK, et al. Relationship between temperature and Anopheles gambiae sensu lato mosquitoes' susceptibility to pyrethroids and expression of metabolic enzymes. Parasit Vectors. 2022 Dec;15(1):163.

7. Johnston HR, Keats BJB, Sherman SL. Population Genetics. In: Emery and Rimoin's Principles and Practice of Medical Genetics and Genomics [Internet]. Elsevier; 2019 [cited 2024 Aug 22]. p. 359–73. Available from: https://linkinghub.elsevier.com/retrieve/pii/B9780128125373000123

8. Marchi N, Schlichta F, Excoffier L. Demographic inference. Curr Biol. 2021 Mar;31(6):R276–9.

9. Felsenstein J. Evolutionary trees from DNA sequences: A maximum likelihood approach. J Mol Evol. 1981 Nov;17(6):368–76.

10. Nielsen R, Wakeley J. Distinguishing Migration From Isolation: A Markov Chain Monte Carlo Approach. Genetics. 2001 Jun 1;158(2):885–96.

11. Beaumont MA, Zhang W, Balding DJ. Approximate Bayesian Computation in Population Genetics. Genetics. 2002 Dec 1;162(4):2025–35.

12. Sunnåker M, Busetto AG, Numminen E, Corander J, Foll M, Dessimoz C. Approximate Bayesian Computation. Wodak S, editor. PLoS Comput Biol. 2013 Jan 10;9(1):e1002803.

13. Nielsen R. Estimation of Population Parameters and Recombination Rates From Single Nucleotide Polymorphisms. Genetics. 2000 Feb 1;154(2):931–42.

14. Excoffier L, Dupanloup I, Huerta-Sánchez E, Sousa VC, Foll M. Robust Demographic Inference from Genomic and SNP Data. Akey JM, editor. PLoS Genet. 2013 Oct 24;9(10):e1003905.

15. Li H, Durbin R. Inference of human population history from individual whole-genome sequences. Nature. 2011 Jul;475(7357):493–6.

16. Schiffels S, Durbin R. Inferring human population size and separation history from multiple genome sequences. Nat Genet. 2014 Aug;46(8):919–25.

17. Browning SR, Browning BL. Accurate Non-parametric Estimation of Recent Effective Population Size from Segments of Identity by Descent. Am J Hum Genet. 2015 Sep;97(3):404–18.

18. Lawson DJ, Hellenthal G, Myers S, Falush D. Inference of Population Structure using Dense Haplotype Data. Copenhaver GP, editor. PLoS Genet. 2012 Jan 26;8(1):e1002453.

19. He C, Shi Z, Qu T, Wang D, Liao M. Lifting Scheme-Based Deep Neural Network for Remote Sensing Scene Classification. Remote Sens. 2019 Nov 13;11(22):2648.

20. Flagel L, Brandvain Y, Schrider DR. The Unreasonable Effectiveness of Convolutional Neural Networks in Population Genetic Inference. Kim Y, editor. Mol Biol Evol. 2019 Feb 1;36(2):220–38.

21. Sheehan S, Song YS. Deep Learning for Population Genetic Inference. Chen K, editor. PLOS Comput Biol. 2016 Mar 28;12(3):e1004845.

22. Spence JP, Steinrücken M, Terhorst J, Song YS. Inference of population history using coalescent HMMs: review and outlook. Curr Opin Genet Dev. 2018 Dec;53:70–6.

23. Fournier R, Tsangalidou Z, Reich D, Palamara PF. Haplotype-based inference of recent effective population size in modern and ancient DNA samples. Nat Commun. 2023 Dec 1;14(1):7945.

24. Schrider DR, Kern AD. Supervised Machine Learning for Population Genetics: A New Paradigm. Trends Genet. 2018 Apr;34(4):301–12.

25. Hopfield JJ. Neural networks and physical systems with emergent collective computational abilities. Proc Natl Acad Sci. 1982 Apr;79(8):2554–8.

26. Kasar MM, Bhattacharyya D, Kim T hoon. Face Recognition Using Neural Network: A Review. Int J Secur Its Appl. 2016 Mar 31;10(3):81–100.

27. Sharma N, Jain V, Mishra A. An Analysis Of Convolutional Neural Networks For Image Classification. Procedia Comput Sci. 2018;132:377–84.

28. Cecil RM, Sugden LA. On convolutional neural networks for selection inference: Revealing the effect of preprocessing on model learning and the capacity to discover novel patterns. Fariselli P, editor. PLOS Comput Biol. 2023 Nov 27;19(11):e1010979.

29. Alzubaidi L, Zhang J, Humaidi AJ, Al-Dujaili A, Duan Y, Al-Shamma O, et al. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. J Big Data. 2021 Mar 31;8(1):53.

30. Taye MM. Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions. Computation. 2023 Mar 6;11(3):52.

31. The Anopheles gambiae 1000 Genomes Consortium. Genetic diversity of the African malaria vector Anopheles gambiae. Nature. 2017 Dec;552(7683):96–100.

32. Baumdicker F, Bisschop G, Goldstein D, Gower G, Ragsdale AP, Tsambos G, et al. Efficient ancestry and mutation simulation with msprime 1.0. Browning S, editor. Genetics. 2022 Mar 3;220(3):iyab229.

33. Wang Z, Wang J, Kourakos M, Hoang N, Lee HH, Mathieson I, et al. Automatic inference of demographic parameters using generative adversarial networks. Mol Ecol Resour. 2021 Nov;21(8):2689–705.

34. Torada L, Lorenzon L, Beddis A, Isildak U, Pattini L, Mathieson S, et al. ImaGene: a convolutional neural network to quantify natural selection from genomic data. BMC Bioinformatics. 2019 Nov;20(S9):337.

35. Poojary R, Raina R, Kumar Mondal A. Effect of data-augmentation on fine-tuned CNN model performance. IAES Int J Artif Intell IJ-AI. 2021 Mar 1;10(1):84.

36. Jakobsson M, Edge MD, Rosenberg NA. The relationship between F(ST) and the frequency of the most frequent allele. Genetics. 2013 Feb;193(2):515–28.

37. Savvides R, Mäkelä J, Puolamäki K. Model selection with bootstrap validation. Stat Anal Data Min ASA Data Sci J. 2023 Apr;16(2):162–86.

38. Nichols K, Holmes A. Non-parametric procedures. In: Statistical Parametric Mapping [Internet]. Elsevier; 2007 [cited 2024 Aug 23]. p. 253–72. Available from: https://linkinghub.elsevier.com/retrieve/pii/B9780123725608500218

39. Goutte C, Gaussier E. A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation. In: Losada DE, Fernández-Luna JM, editors. Advances in Information Retrieval [Internet]. Berlin, Heidelberg: Springer Berlin Heidelberg; 2005 [cited 2024 Aug 23]. p. 345–59. (Hutchison D, Kanade T, Kittler J, Kleinberg JM, Mattern F, Mitchell JC, et al., editors. Lecture Notes in Computer Science; vol. 3408). Available from: http://link.springer.com/10.1007/978-3-540-31865-1_25

40. Mathew J, Kshirsagar R, Abidin DZ, Griffin J, Kanarachos S, James J, et al. A comparison of machine learning methods to classify radioactive elements using prompt-gamma-ray neutron activation data [Internet]. 2023 [cited 2024 Aug 23]. Available from: https://www.researchsquare.com/article/rs-2518432/v1

41. Wörz S, Bernhardt H. Towards an uniformly most powerful binomial test. Stat Pap. 2020 Oct;61(5):2149–56.

42. Fu YX, Li WH. Maximum likelihood estimation of population parameters. Genetics. 1993 Aug 1;134(4):1261–70.

43. Mourad R, Sinoquet C, Leray P. Probabilistic graphical models for genetic association studies. Brief Bioinform. 2012 Jan 1;13(1):20–33.

44. Hejase HA, Dukler N, Siepel A. From Summary Statistics to Gene Trees: Methods for Inferring Positive Selection. Trends Genet TIG. 2020 Apr;36(4):243–58.

45. Zhao H, Pavlidis P, Alachiotis N. SweepNet: A Lightweight CNN Architecture for the Classification of Adaptive Genomic Regions. In: Proceedings of the Platform for Advanced Scientific Computing Conference [Internet]. Davos Switzerland: ACM; 2023 [cited 2024 Aug 19]. p. 1–10. Available from: https://dl.acm.org/doi/10.1145/3592979.3593411

46. Smith CCR, Tittes S, Ralph PL, Kern AD. Dispersal inference from population genetic variation using a convolutional neural network. Novembre J, editor. GENETICS. 2023 May 26;224(2):iyad068.

47. Goswami C, Chattopadhyay A, Chuang EY. Rare variants: data types and analysis strategies. Ann Transl Med. 2021 Jun;9(12):961.

48. Lau Y, Sim W, Chew K, Ng Y, Arabee Z, Salam A. Understanding how noise affects the accuracy of CNN image classification. In 2021. Available from: https://api.semanticscholar.org/CorpusID:245739577

49. Nguembang Fadja A, Riguzzi F, Bertorelle G, Trucchi E. Identification of natural selection in genomic data with deep convolutional neural network. BioData Min. 2021 Dec 4;14(1):51.

50. Chan J, Perrone V, Spence JP, Jenkins PA, Mathieson S, Song YS. A Likelihood-Free Inference Framework for Population Genetic Data using Exchangeable Neural Networks. Adv Neural Inf Process Syst. 2018 Dec;31:8594–605.

51. Hoggart CJ, Chadeau-Hyam M, Clark TG, Lampariello R, Whittaker JC, De Iorio M, et al. Sequence-level population simulations over large genomic regions. Genetics. 2007 Nov;177(3):1725–31.

Supplementary Tables & Figures

Table S1. FST Simulation Parameters

Parameter	Symbol	Value	Unit
Population 1	*N1*	158,124,480	Individuals
Population 2	*N2*	54,259,530	Individuals
Ancestral Population	N_{anc}	7,148,911	Individuals
T Split	T_{split}	1018	Generations
Mutation Rate	*mut*	3.5×10^{-9}	Per base per generation
Sequence Length	*L*	10,000	DNA base
Sample Size	N_{sample}	50	Individuals
Recombination Rate	*reco*	8.4×10^{-9}	Per base per generation
Number of Simulations	N_{sims}	1000	Simulations per migration rate
Low Migration Rate	m_1	1×10^{-9}	Fraction of migrating individuals per generation
Medium Migration Rate	m_2	0.1	Fraction of migrating individuals per generation
High Migration Rate	m_3	0.9	Fraction of migrating individuals per generation

Table S2. SVM Bootstrap & Permutation Metrics

Evaluation Metric	Value
Number of Bootstrap/Permutation Iterations	1000
Average Bootstrap Accuracy	35.06%
Bootstrap Standard Deviation	2.41%
Average Permutation Accuracy	34.84%
Permutation Standard Deviation	2.54%
P-value	0.47
Significance Threshold	0.05
Is Result Significant?	Insignificant

Table S3. CNN, Precision, Recall & F1-Score Metrics

Migration Rate Class	Precision	Recall	F1-Score
High	0.43	0.41	0.42
Medium	0.33	0.27	0.3
Low	0.34	0.41	0.37
Macro Average	0.36	0.36	0.36

Table S4. CNN Binomial Testing Metrics

Evaluation Metric	Value
Correct Predictions	109
Total Predictions	300
Model Accuracy	36.33%
Baseline Accuracy	33.33%
P-value	0.15
Significance Threshold	0.05
Is Result Significant?	Insignificant

Table S5. CNN Model Training Statistics

Layer	Output Shape	Number Of Parameters
First 2D Convolutional Layer	(None, 98, 4880, 64)	640
First 2D Max Pooling Layer	(None, 32, 1626, 64)	0
Second 2D Convolutional Layer	(None, 30, 1624, 32)	18,464
Second 2D Max Pooling Layer	(None, 10, 541, 32)	0
FC Layer: Flattening	(None, 173120)	0
FC Layer: First Dense	(None, 64)	11,079,744
FC Layer: Second Dense	(None, 3)	195
Total Parameters	11,099,045	
Trainable Parameters	11,099,043	
Non-Trainable Parameters	0	
Optimiser Parameters	2	

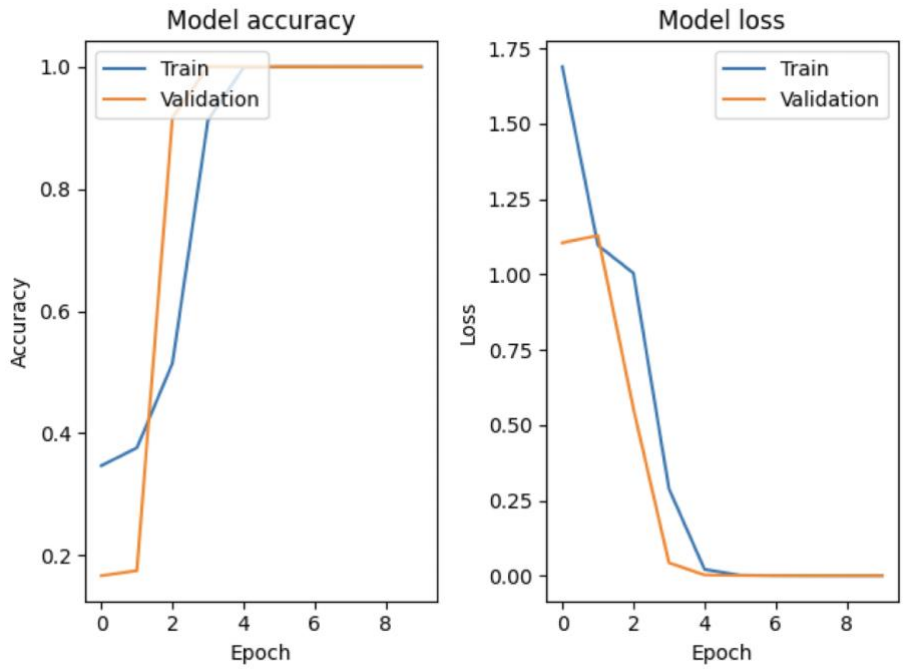


Figure S1. Effect of data augmentation on CNN training and validation accuracy