IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

SPECIAL SECTION ON EMERGING DEEP LEARNING THEORIES AND METHODS FOR BIOMEDICAL ENGINEERING

Received August 9, 2020, accepted September 1, 2020, date of publication September 7, 2020, date of current version September 24, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3022274

A Novel Deep Neural Networks Model Based on Prime Numbers for Y DNA Haplogroup Prediction

JASBIR DHALIWAL, KEONG JIN, AND ZHE JIN, (Member, IEEE)
School of Information Technology, Monash University Malaysia, Bandar Sunway 47500, Malaysia

Corresponding author: Jasbir Dhaliwal (jasbirkaur.dhaliwal@monash.edu)

ABSTRACT Most of the Y chromosome (Ychr) region (approximately 95%) passes unchanged from father to son, except by the gradual accumulation of single-nucleotide polymorphism (SNP) mutations. This results in mutations being inherited together, where all males in the direct family will have an identical pattern of variations. These mutation patterns serve as markers and can be mapped into clusters known as Y DNA haplogroups. Besides lineage tracing, haplogroups have been associated with male infertility, semen parameters, and, more recently, disease progression in several populations. Thus, haplogroup prediction research is gaining importance because of the increasing interest in personalized medicine. Of note, there are two approaches to predicting haplogroups, where the difference lies in the genetic markers: short tandem repeats (STRs) or SNPs are inputs to the haplogroup prediction tools. STRs are not without limitations, as similar STR haplotypes exist between haplogroups, and this reduces the effectiveness of STR-based haplogroup prediction tools. By contrast, current SNP-based haplogroup prediction tools are computationally expensive. There have been no studies to date that leverage traditional machine learning and deep learning algorithms to identify mutation patterns using SNPs only, and this paper proposes a novel SNP-based deep neural networks (DNNs) model. However, DNNs suffer the curse of dimensionality and become computationally expensive with large datasets. Thus, this paper overcomes the limitation of the network by proposing a novel feature extraction method based on prime numbers that computes features in either the forward or reverse direction of the SNPs data. Our experimental results show that the model achieves a categorical cross-entropy loss value as low as 0.001 on the training dataset and as low as 0.039 on the test dataset.

INDEX TERMS Bioinformatics, feature extraction, multi-layer neural networks, deep learning.

I. INTRODUCTION

The Ychr is well-known for encoding sex-determining genes and a few other male-specific genes through translocation and transposition. Ychr is known to undergo genetic degradation: over the last 300 million years, the ancestral autosome that evolved to the human Ychr lost all (∼1500 genes) except for ∼78 of its genes. Thus, the Ychr was commonly regarded as a genetic wasteland. Jacobs *et al.* [10] first described a frequent loss of Ychr (LOY) in hematopoietic cells of aging men in 1963. For many years, researchers accepted the genetic wasteland view, and LOY was believed to be phenotypically neutral and an age-related phenomena [11], [12]. However, a recent study [13] seems to suggest the opposite; LOY may be involved with disease progression in various organs. These studies [14]–[17] show that the frequency of the LOY in the cancer genomes ranges between 15-80% in different types of cancer disease.

An interesting attribute of Ychr is that the male-specific region is a nonrecombining region, and constitutes approximately 95% of the chromosome [18]. This region is passed down from father to son unchanged, except by the gradual accumulation of SNP mutations [19]. As a result, any mutations that occur on a Ychr will always be inherited together. These mutations trace the lineage on the Ychr, where all males in the direct family will have an identical pattern of variations. Thus, Ychr is used exclusively in surname testing and forensic identification of male offenders or victims via lineage tracing [20].

Correct interpretation of the mutation patterns can further improve our understanding of population and migration history. These mutations serve as markers and can be

The associate editor coordinating the review of this manuscript and approving it for publication was Yudong Zhang.

mapped into clusters known as Y DNA haplogroups (sometimes known as haplogroups or Y haplogroups) [19]. A haplogroup represents a group of people who have inherited common genetic markers from the same most recent common ancestor. Such information is useful not only to trace the paternal ancestry of an individual but also population events (e.g., migrations and bottlenecks) [21]–[23]. Moreover, various subgroups indicate different geographic signatures. For example, the following subgroups of haplogroup R represent particular geographic subregions: R2a for South Asia, R1b1c for Africa/Middle East, R1a1a1g for (Eastern) Europe and R1a1a1f for West Asia [24].

Besides lineage tracing, Y DNA haplogroups have been associated with male infertility [25]–[28], semen parameters [29], and, more recently, disease progression such as cardiovascular risk [30], [31], coronary artery disease [32], blood pressure [33] and prostate cancer [34], [35] in several populations. Thus, haplogroup prediction research is gaining importance because of the increasing interest in personalized medicine.

There are two approaches to predicting Y DNA haplogroups, where the difference lies in the genetic markers, STRs or SNPs, are inputs to the haplogroup prediction tools. Table 1 summarizes a few well-known STR-based and SNP-based prediction tools, including their advantages and limitations. STRs are not without limitations, as pointed out in more recent studies [2], [19], [36]–[38]. One major limitation is the existence of similar STR haplotypes between haplogroups [36], [39]. The study in [36] reported similar haplotypes between the haplogroups: B and I2, C1 and E1b1b1, C2 and E1b1a1, H1 and J, L and O3a2c1, O1a and N, O3a1c and O3a2b, and M1 and O3a2. As expected, such similarities reduce the accuracy of STR-based prediction tools. This conclusion is supported in separate studies [37], [38], [40], which suggest SNP analysis as a second validation step if accurate predictions are required. Thus, the most recent studies use a combination of STR and SNP as genetic markers [41], [42] for haplogroup predictions. The researchers in [41] used a phylogenetic tree of SNPs to represent haplogroups of samples from a known dataset. The tree was then used as a ground truth to facilitate variant findings in STR haplotypes.

However, with regard to phylogenetic trees, *yHaplo*, as indicated in Table 1, is the current state-of-the-art of the SNP-based haplogroup prediction tools but suffers from computational complexity. Moreover, the SNP alleles need to be on the correct strand as DNA is double-stranded. The definition of strand has been controversial [43]–[45]. The most intuitive definition of a strand uses the human genome reference as the forward strand; however, this has not been standard practice. Thus, the SNPs for Ychr need to be validated with the International Society of Genetic Genealogy (ISOGG) database [46] or other resources [47] to ensure correct reference and alternate allele labeling.

There is no doubt that the ISOGG database and the strand information are crucial for the ground truth labeling of haplogroups and subgroups of samples even in a machine learning model. However, upon labeling, a machine learning model can learn the distinctive mutation patterns of the training datasets in either direction (forward or reverse) of the sequence data and infer the new sample's haplogroup from the training set. This is possible because a machine learning model derives its power from its ability to differentiate patterns from the data itself. To the best of our knowledge, there have been no studies that leverage traditional machine learning and deep learning algorithms to identify those patterns using SNPs only, and this paper proposes to use deep learning. By contrast, the most recent work on haplogroup prediction uses a combination of STRs and SNPs with traditional machine-learning algorithms [42].

Deep learning is the emerging generation of artificial intelligence techniques and has grown immensely in applications in fields [48] ranging from computer vision to speech to signal processing to sequence and text prediction, and more recently, to bioinformatics and computational biology [49]–[54]. The basic models in deep learning are derived from artificial neural networks (ANNs), and deep neural networks (DNNs) is one such example. The contributions of this paper are summarized below:

TABLE 1. Y DNA haplogroup prediction tools and their advantages and limitations. Note that we have left out the STR-based predictor tools developed by Schlecht *et al.* [1], Felix Immanuel [2] and Vadim Urasin [3], as the links to their software are no longer available.

Software Tool	Description	Markers	
		STR	SNP
Athey (2005) [4]	Whit Athey's Y-Haplogroup Predictor. This predictor uses the Bayesian allele-frequency approach to estimate the probability of a given haplotype belonging to a particular haplogroup based on the allele frequencies of the STR haplotypes deposited in public databases [5]. Thus, according to the author, the scarcity of STR haplotypes for haplogroups C, H, L, N and Q may result in inaccurate predictions for those groups. This is because the software requires the selection of the geographic origin of the haplotypes being analyzed.	✓	
Cullen [6] (2008)	Jim Cullen's World Haplogroup & Haplo-I Subclade Predictor. This predictor was inspired by *Athey*, which uses a weighted genetic distance algorithm that is a variation of a goodness-of-fit test for predicting haplogroup I and its subgroups.	✓	
Nevgen [7] (2014)	NevGen's Y DNA Predictor. This predictor was also inspired by *Athey* and uses the Bayesian allele frequency approach. Thus, as mentioned by the author, a scarcity of STR haplotypes for certain haplogroups may result in inaccurate predictions. This tool works best on European or Near Eastern haplogroups.	✓	
yhaplo [8] (2016)	23andMe's Y Haplogroup Predictor. This predictor builds a phylogeny tree using major haplogroups and their relationships and a set of phylogenetically informative SNPs curated by the ISOGG and growing the tree as needed. The researcher modified the breadth-first search algorithm to overcome issues related to missing data, genotype errors and mutation recurrences. However, this predictor suffers a limitation of the traditional breadth-first search algorithm where the search becomes exhaustive if the number of offspring for each node is large [9] and for a large number of samples. However, such an algorithm is most effective if the tree has a uniform depth. In addition, multiallelic SNPs are excluded.		✓

- Proposes a novel SNP-based DNNs model to learn the patterns between haplogroups and subgroups.
- Proposes a novel feature extraction method based on prime numbers to select SNPs as features. This is because DNNs suffer from the curse of dimensionality and become computationally expensive when used for a large number of SNPs and samples.
- Provides a comprehensive analysis of the experimental results. We show that patterns learned in either direction (forward or reverse) of the SNPs data can be used to infer the haplogroup of a new sample.

The rest of this paper is organized as follows. Section II provides an overview of Ychr. This overview can be skimmed by readers already familiar with this field but may serve as a useful tutorial for those new to the problem. Next, Section III describes our DNNs model for the problem domain, including the selection of hyperparameters and the intuition behind them. Section IV presents our methodology, describes the experiments, and discusses the obtained results. Finally, we draw some conclusions in Section V.

II. OVERVIEW OF Y CHROMOSOME

In this section, we provide a brief overview of the cytogenetic structure of Ychr, the definition of mutation and the Y DNA haplogroups.

A. CYTOGENIC STRUCTURE

Ychr is the smallest chromosome in humans (~60 MB) [55] and is acrocentric. It has a short arm (Yp) and a long arm (Yq) that is separated by a centromeric region that is important for chromosome segregation during male meiosis [19]. There are three major regions in Ychr: pseudoautosomal, euchromatin and heterochromatin. Only pseudoautosomal regions, indicated by PARs in Figure 1, are involved in meiosis.

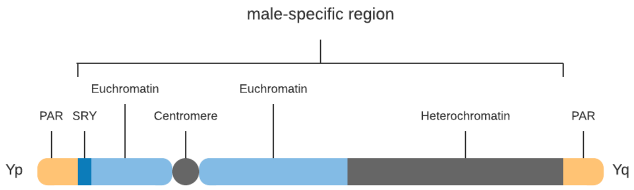


FIGURE 1. Cytogenic structure of Y chromosome. SRY refers to sex-determining region of the Y.

In theory, a chromosome's whole-genome sequence data allow us to construct reliable phylogeny where the length of the branch is proportional to the number of SNPs, which is then mapped to build a maximum parsimony tree to infer the phylogeny of the sequences [19]. However, in practice, the structure of Ychr is very different than those of other chromosomes, where even with the most advanced technology only certain regions of the Ychr can be mapped unambiguously. These regions are scattered across the chromosome and add up to a length of 9.9 Mb. This ~10 Mb is known as the callable region of Ychr [56]. Hereafter, we refer to SNPs in this region as callable SNPs.

B. MUTATION

Mutation (sometimes known as polymorphism) is the ultimate source of all genetic diversity and is any change in the DNA sequence. This can range from the substitution of a single base in the genome to small insertions and deletions of a few bases. A mutation only exists when at least two different alleles are present in a population, and both are present at $\geq 1\%$ frequency [57].

An SNP is the most common genetic variation among individuals and involves a single base difference in a single DNA building block that is commonly known as a nucleotide. For example, an SNP may replace the nucleotide adenine (A) with the nucleotide guanine (G). On the other hand, STRs are repeated units of 1-7 base pairs in length, and those with a useful degree of polymorphism have a frequency of 10-30 [57]. The mutation rate refers to the frequency of mutations in a single gene, chromosome or even in an individual over time.

C. Y DNA HAPLOGROUPS

A haplotype is a combination of allele states of polymorphisms on the same chromosome, whereas a haplogroup is a group of similar haplotypes that share a common ancestor with an SNP mutation [57]. In theory, if an individual has a different SNP than another, they can be said to be in different haplogroups. However, in practice, this requires keeping track of millions of groups. Therefore, the Y Chromosome Consortium (YCC) [58] was formed in 2002 to collate all phylogenetically informative SNPs and assign universal nomenclatures to each recognized haplogroup. YCC defined a single capital letter to indicate major haplogroups, where letters A to T have been used. For the subgroups, two nomenclature systems were proposed: lineage-based, where names are alphanumeric (e.g., E1b1b1b1a); and mutation-based, where terminal SNP mutation is used to define them (e.g., E-M81). Both examples refer to the same subgroup of major haplogroup E. To date, 20 major haplogroups have been identified with numerous subgroups [59]. Figure 2 shows the phylogeny tree of the Ychr's SNP data of the 1000 Genomes Project [60], where the tips of the branches represent the major haplogroups.

III. DEEP NEURAL NETWORKS

This section describes a novel DNNs model and the hyperparameters used in this study. To ensure that both the network and hyperparameters are independent of the datasets used in the same problem domain since overfitting is a serious problem in neural networks, we used the preprocessed SNPs of 39 individuals of the CEU population [61] of the International HapMap Project [62] for the initial exploration of various deep learning models.

We selected the DNNs as the deep learning model of interest. Furthermore, recent research has shown than ANNs can handle small datasets [63], and this characteristic of the

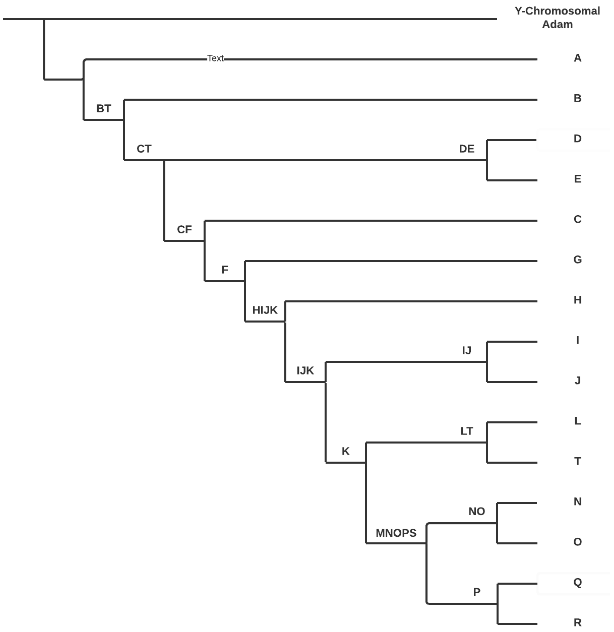


FIGURE 2. Tips of branches depict major haplogroups of populations from 1000 Genomes project. Y-Chromosomal Adam refers to most recent common ancestor on paternal line of all living males and is shown here for completeness.

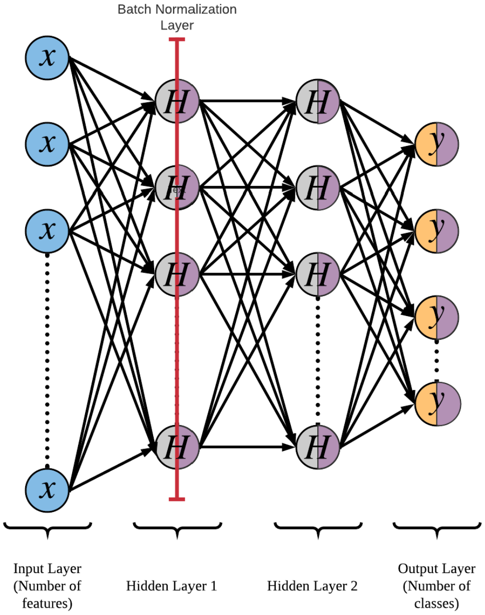

Batch Normalization Layer
x
x
x
x
H
H
H
H
H
H
H
H
y
y
y
y
Input Layer (Number of features)
Hidden Layer 1
Hidden Layer 2
Output Layer (Number of classes)

FIGURE 3. DNNs model for haplogroup and subgroup classification. There are 312 neurons with Tanh activation used on hidden layer 1 and 314 neurons with ELU activation used on hidden layer 2. Output layer used Softmax activation. Dropout rate of 20% is used for input neurons and of 50% for hidden neurons. See in text for details.

neural network fits our problem domain as it is challenging to obtain large datasets when compared to other big data domains. Upon selecting the DNNs as the model, we used the Myanmar dataset [61] (see Section IV for details) to search for the optimal hyperparameters.

Hyperparameter optimization is an area of research itself and several approaches ranging from grid search to genetic algorithms (GAs) [64] have been used. In this paper, we implemented a GA to search some of the optimized hyperparameters, while others are chosen intuitively by using a trial-and-error approach. This approach has been used in different problem domain studies [65]–[72]. However, in a new problem domain, it is unknown which hyperparameters are important and what values produce good performance. Thus, we also discuss the most important hyperparameters that we chose for our problem domain and the intuition behind these choices.

A. MODEL

The DNNs model of this paper is presented in Figure 3. The leftmost layer is known as the input layer, while the rightmost layer is known as the output layer. The input layer consists of input neurons that represent the number of SNP-based features. Meanwhile, the output neurons represent the number of unique classes based on either haplogroups or subgroups. By contrast, the hidden neurons in the two hidden layers form the network's bottleneck.

B. HIDDEN NEURONS

There is a trade-off between the number of hidden neurons with the training error rate, where using too few results in underfitting and too many results in overfitting [72]. There has been no census on the exact number; many studies have proposed several heuristics using a trial-and-error approach [69]–[71].

Sheela and Deepa [72] reviewed methods for fixing the number of hidden neurons in neural networks for the past 20 years and showed that their approach gave the lowest training error rate when measured using the Mean Squared Error (MSE) metric on weather datasets. We compared the network's MSE values using the number of hidden neurons obtained using the GA with their approach, and the comparison results are given in Table 2. The lower the MSE value, lower the error rate and the better the estimator.

TABLE 2. Number of hidden neurons and MSE values when compared to approach of Sheela and Deepa [72].

Literature	Hidden neurons on layer 1	Hidden neurons on layer 2	MSE
Sheela and Deepa [72]	4	4	0.099
This paper	312	314	0.075

Moreover, to prevent the saturation of hidden neurons, we explored z-score and min-max normalizations, found the latter giving better results. This paper uses the min-max normalization. Similarly, Aksu *et al.* [73] reviewed the effect of various normalization methods on educational sciences datasets with ANNs and found the min-max normalization yielding the best results.

C. ACTIVATION FUNCTION

The activation function provides nonlinear modeling capabilities for networks, and only by adding activation functions, DNNs possess hierarchical nonlinear mapping learning ability. Recently, Nwankpa *et al.* [68] reviewed the majority of activation functions in deep learning research, including their practical applications in deep learning models. The researchers proposed the use of Softmax activation on the output layer for the multiclass classification problem; this approach is also used in this paper.

By contrast, the activation functions for the two hidden layers are obtained using the GA, where the hyperbolic tangent (Tanh) is the activation function for hidden layer 1, and the exponential linear unit (ELU) is the activation function for hidden layer 2. The study in [74] showed that "Tanh-Tanh" combination of activation functions for both hidden and output neurons gave better training performance in multilayered perceptron architectures of neural networks [74]. By contrast, ELU has been used to speed up the training of deep neural networks [75] on computer vision datasets. Instead, this paper proposes "Tanh-ELU-Softmax" combination of activation function for both the hidden and output neurons.

To normalize the above activations in the intermediate layers of the network, a batch normalization layer was proposed by [76] and is used only with the first hidden layer, as indicated in Figure 3 as this network has only two hidden layers.

D. GRADIENT PROCESSING

A stochastic gradient with minibatches is used as the convergence depends on the updates and richness of the training distribution and not the size of the training dataset [65]. This characteristic fits our problem domain due to the small sample size issue described earlier.

Table 3 shows hyperparameters tuned for gradient processing as well as the recommended values. Bengio [65] researched gradient-based training for deep architectures and recommended values for the initial learning rate and batch size. Moreover, the study in [67] showed that small batch sizes improve the generalization performance of DNNs. By contrast, Goodfellow, Bengio and Courville recommended momentum ranges in their deep learning book [66].

TABLE 3. Recommended values of [65] and [66], and values used in the paper.

Hyperparameters	Recommended values	This paper
Initial learning rate	$> 1e - 6$ and < 1	$1e - 1$
Batch size	between 1 and a few hundred, where 32 is a good default value	32
Momentum	$5e - 1, 9e - 1, 99e - 1$	$1e - 1$
Learning rate decay	trial-and-error	$55e - 4$

The epoch is a hyperparameter related to the batch size. The epoch's value was 177, which was obtained using the GA. There are no recommended values for this parameter except through trial-and-error.

E. DROPOUT

Dropout is a technique proposed by [77] to prevent the network from coadapting too much by literally dropping neurons. The recommended dropout rate for the input neurons is 20%, and the rate for hidden neurons is 50%. These values are also used in this paper.

F. GENETIC ALGORITHM

We now describe the GA that we implemented for hyperparameters tuning. GA begins by creating an initial population of DNNs with random values assigned for epoch, hidden neurons, and activation functions. Then, the algorithm selects the top two networks based on a fitness criterion defined by machine learning metrics, described in Section IV-B4, to become parents while discarding the remaining networks. The parent networks are used for breeding children through the cross over and mutation steps. The pseudocode GA of the algorithm is presented as Algorithm 1.

Algorithm 1 Genetic Algorithm for Finding Hyperparameters

```
1  pseudocode GA (Data, $p$, $g$);
   Input  : Data, population size $p$ and number of
            generations $g$.
   Output : Optimized hyperparameter values for epoch
            (e), hidden neurons ($h^1$, $h^2$) and activation
            functions ($a^1$, $a^2$).
2  Create an initial $p$-sized population of DNNs.
3  Evaluate the population using fitness criteria defined by
   the machine learning metrics.
4  Select the top two networks with the highest fitness
   scores to become parents, $P$.
5  while current population size $< p$ do
6      Create child $C$ via cross over of $P$.
7      Mutate $C$ based on some randomness.
8      Add $C$ to new population.
9  end
10 repeat steps 3 - 9 until the $g$th generation.
```

Many variations of cross over and mutation steps exist. Of note, we present the pseudocode CO of the cross over step, that we implemented as Algorithm 2. We chose ratio values of between 0 and 1, particularly values 0.75 and 0.25, as they yielded good results while experimenting with various ratios on the training dataset.

Similarly, we present the pseudocode Mut of the mutation step that we implemented as Algorithm 3, where a child is selected randomly. We chose a ratio of 0.50 as it worked sufficiently well on the training dataset.

IV. EXPERIMENTAL RESULTS

This section describes the steps taken in the experiments of this study as well as the justification given for each decision made for each method used, and then provides a comprehensive analysis of the results.

Algorithm 2 Cross Over Algorithm for Creating a Child *C* With Some Properties From Parents *P*

```
pseudocode CO($P$);
Input  : Parents $P$.
Output : Child $C$ with values for epoch (e), hidden
         neurons ($h^1$, $h^2$) and activation functions ($a^1$,
         $a^2$).
Randomly assign father and mother from $P$ to $F$ and $M$,
respectively.
$C[e] = F[e] * 0.25 + M[e] * 0.75$
$C[h^1] = F[h^1] * 0.75 + M[h^1] * 0.25$
$C[h^2] = F[h^2] * 0.25 + M[h^2] * 0.75$
$C[a^1] = F[a^1]$
$C[a^2] = M[a^2]$
```

Algorithm 3 Mutate Algorithm for Mutating Some of the Properties of the Child

```
pseudocode Mut($C$);
Input  : Child $C$.
Output : Mutated child $C$ with updated values for
         epoch (e) and hidden neurons ($h^1$, $h^2$).
$C[e] = C[e] + (C[e] * 0.50)$
$C[h^1] = C[h^1] + (C[h^1] * 0.50)$
$C[h^2] = C[h^2] + (C[h^2] * 0.50)$
```

A. DATA

Apart from the full Ychr sequences, which is the highest level of phylogenetic resolution, there are Ychr SNP datasets with low and medium resolutions as a different set of SNPs might have been sequenced depending on the objective of their study. Moreover, unlike other big data problem domains, it is not easy to acquire large sample sizes to obtain a good representative of the haplogroups and their subgroups.

Thus, to represent our problem domain accurately, we used the Myanmar (*Myan*) [61] data consisting of 106 samples (i.e., individuals) as the training dataset. Phase 3 of the 1000 Genomes project (*1000 Genomes*) [60] data, consisting of 1,233 samples, was used as the test dataset. Both these datasets use the hg19 build. We have separated the training and test datasets so that the test set is invisible to the network's training to confirm the network's actual predictive power.

B. METHODOLOGY

In this section, we present our experimental framework of preprocessing, feature extraction methods, classification and evaluation metrics.

1) PREPROCESSING

For the *Myan* data, we used a Python script to preprocess the genotype data to a sequence of 0s and 1s as the files were in PLINK format. A "0" indicates that the SNP is similar to the reference allele, whereas a "1" indicates that a mutation has occurred at that position. However, such preprocessing was not required for the *1000 Genomes* data, as it was already in a sequence of 0s and 1s. Thus, we used BCFtools to extract the required fields from the VCF file before using a Python script for further processing. Moreover, if a sample has a missing value at a particular position, the said position is removed from all of the samples. Therefore, *1000 Genomes* data contains 58,732 SNPs, and *Myan* data contains 2,041 SNPs. Hereafter, we refer to the above preprocessed SNPs (in the form of 0s and 1s) as simply SNPs.

As we are using stratified 3-fold cross-validation, we ensure each haplogroup or subgroup has at least four samples. Of note, the stratified K-fold is a commonly accepted cross-validation technique. Stratification is a process that ensures each fold is a good representative of each class, which is dependent on the number of samples. This technique splits the dataset into groups known as folds and, in our case, three folds, where two folds are used for training the model while the remaining fold is used for testing the model.

This resulted in 1,216 samples and 27 classes of haplogroups, and 985 samples and 76 classes of subgroups, for *1000 Genomes* data. For the *Myan* data, there were 76 samples and 4 classes of subgroups. The samples were labeled based on the ground truth information of their datasets.

2) PRIME-NUMBER-BASED FEATURE EXTRACTION METHOD

DNNs suffer the curse of dimensionality. Thus, if we take each SNP as a feature, it becomes computationally expensive and encourages overfitting. This inspired us to use mutation information, i.e., the number of mutations in a particular range, as a feature instead. To do so, we divided the SNPs into fixed partitions and calculated the number of mutations, i.e., the mutation rate per partition. However, our exploration results show it is computationally expensive if we search for the optimal partition size, where we fall back to the same computational complexity problem that we are trying to overcome.

While exploring other approaches, we found that using number theory sequences, particularly prime number sequences as partition sizes, works sufficiently well. Thus, each prime number indicates the mutation rate per partition and is used as a feature. Mutation rates are computed in forward and reverse directions of the SNPs data, as shown in Figure 4. This results in two prime-number-based feature sets: forward and reverse. For example, the first feature in the reverse feature set has a mutation rate of "2" as there are two mutations in the first partition. On the other hand, the first feature in the forward feature set has a mutation rate of "0", as there are no mutations in the first partition.

The pseudocode CPF of the algorithm that computes the forward feature set is presented as Algorithm 4. CPF begins by making a left-to-right scan over the SNPs data to calculate the cumulative sum. A cumulative sum is a sequence of partial sums of a given sequence, and in our case, the sum of

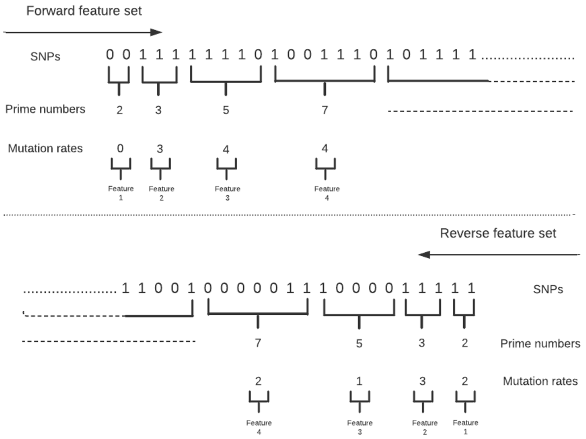


FIGURE 4. **Forward and reverse feature sets of prime-number-based features.**

mutations of the SNPs data. This aids in efficient processing as positions closer to the current partition are used to obtain the mutation rate instead of having to recompute them every time we need to partition the data. A sequence of prime numbers forms the boundaries of the partitions, where each subsequent pair denotes the lower and upper boundary of the partition, respectively. By contrast, the reverse feature set is computed by modifying pseudocode CPF to make a right-to-left scan at step 2.

Algorithm 4 Compute Forward Feature Set of Prime-Number-Based Features

```
1 pseudocode CPF(SNPs,$u$);
  Input  : SNPs data and upper limit of prime numbers $u$.
  Output: Forward feature set of prime-number-based
          features, $F$.
2 Make a left-to-right scan over SNPs to compute the
  cumulative sum and store it in $C$.
3 Compute a list of prime numbers based on $u$ and store it
  in $P$.
4 while not end of list $P$ do
5     Scan $C$ for the positions indicated by the first
      subsequent pair of prime numbers.
6     Compute the mutation rate by taking the difference
      of the number of mutations between those two
      positions.
7     Store the mutation rate in $F$.
8     Repeat steps 5-7 for the next subsequent pair of
      prime numbers.
9 end
```

3) CLASSIFICATION

For the first experiment, we used forward and reverse prime-number-based feature sets on the DNNs classifier described in Section III. The purpose of this experiment was twofold. First, it assessed if our model can accurately classify haplogroups and subgroups using prime-number-based features, and which feature set gave the best results. Second, it elucidated the complexity of the model with the new feature sets. Thus, for baseline comparisons, we used each SNP as a feature on the same DNNs classifier. Hereafter, we refer to this feature set as baseline.

To further gauge the performance of the prime-number-based feature set, and whether our model can classify accurately using fewer SNPs than the callable SNPs, variations in the number of features were also assessed.

4) EVALUATION METRICS

We evaluated the described experiments using four machine learning evaluation metrics: categorical cross-entropy loss indicated as loss, accuracy, prediction and recall. All values were between 0 and 1, and the standard deviations are shown in brackets. Cross-entropy loss increases as the predicted labels continue to differ from the predicted labels. By contrast, accuracy is a metric tied to precision and recall. High precision and recall scores show that the classifier is giving accurate results (high precision), and the majority of the results are positive (recall).

C. RESULTS AND DISCUSSION

For the training dataset (consisting of 4 classes), both the forward and reverse feature sets (311 prime-number-based features each) used about the same running time (∼6.4 seconds in contrast to the 2,041 baseline features, which used ∼8.6 seconds). However, the forward feature set achieved the lowest loss value (cf. 0.001 with 0.093 on the reverse feature set and 0.007 on the baseline feature set).

Table 4 summarizes the results on the test dataset consisting of 27 classes when all of the 5,943 prime-number-based features are used in contrast to the 58,732 baseline features. Once again, both the forward and reverse feature sets use about the same running time (∼ 200 seconds in contrast to the baseline, which used ∼ 32 minutes). The forward feature set achieved the lowest loss value (cf. 0.039 with 0.072 on the reverse feature set and 0.056 on the baseline feature set). On the other hand, Tables 5 and 6 show the variation in results for the forward and reverse feature sets. We can conclude that the reverse feature set gave better prediction results. This is because when 512 features were used in both feature sets, the reverse feature set gave the lowest loss value (cf. 0.038 with 0.092 on the forward feature set). Of note, we ran the model on the same settings for the subgroups

TABLE 4. **Comparison between baseline feature set and prime-number-based feature sets on test dataset (27 classes). Standard deviation given in brackets.**

Metrics	Baseline feature set	Forward feature set	Reverse feature set
Time	1966.64s	201.84s	200.62s
Loss	0.056 (0.046)	0.039 (0.015)	0.072 (0.062)
Accuracy	0.982 (0.021)	0.995 (0.002)	0.977 (0.028)
Precision	0.988 (0.012)	0.995 (0.002)	0.977 (0.028)
Recall	0.976 (0.029)	0.995 (0.002)	0.977 (0.028)

TABLE 5. Effect of variation in number of features on forward feature set of test dataset (27 classes). Standard deviation given in brackets.

Features	Loss	Accuracy	Precision	Recall
512	0.092 (0.089)	0.976 (0.027)	0.975 (0.028)	0.972 (0.032)
1024	0.083 (0.082)	0.975 (0.030)	0.977 (0.027)	0.966 (0.042)
2048	0.069 (0.059)	0.977 (0.028)	0.977 (0.028)	0.977 (0.028)
4096	0.052 (0.037)	0.980 (0.023)	0.982 (0.020)	0.980 (0.023)

TABLE 6. Effect of variation in number of features on reverse feature set of test dataset (27 classes). Standard deviation given in brackets.

Features	Loss	Accuracy	Precision	Recall
512	0.038 (0.014)	0.990 (0.004)	0.991 (0.003)	0.990 (0.004)
1024	0.082 (0.076)	0.976 (0.027)	0.976 (0.028)	0.973 (0.031)
2048	0.055 (0.044)	0.977 (0.027)	0.978 (0.026)	0.977 (0.028)
4096	0.065 (0.054)	0.977 (0.028)	0.977 (0.028)	0.977 (0.028)

TABLE 7. Comparison between baseline feature set and prime-number-based feature set on test dataset (76 classes). Standard deviation given in brackets.

Metrics	Baseline feature set	Forward feature set	Reverse feature set
Time	1615.67s	162.31s	160.35s
Loss	0.320 (0.039)	0.370 (0.049)	0.331 (0.050)
Accuracy	0.921 (0.015)	0.897 (0.012)	0.914 (0.017)
Precision	0.931 (0.014)	0.918 (0.009)	0.928 (0.010)
Recall	0.902 (0.017)	0.876 (0.014)	0.899 (0.023)

TABLE 8. Effect of variation in number of features on forward feature set of test dataset (76 classes). Standard deviation given in brackets.

Features	Loss	Accuracy	Precision	Recall
512	0.585 (0.044)	0.811 (0.019)	0.850 (0.023)	0.763 (0.013)
1024	0.588 (0.055)	0.815 (0.019)	0.860 (0.012)	0.801 (0.014)
2048	0.495 (0.050)	0.858 (0.016)	0.897 (0.004)	0.841 (0.015)
4096	0.397 (0.067)	0.883 (0.013)	0.901 (0.007)	0.863 (0.012)

consisting of 76 classes but obtained high cross-entropy loss due to the small sample size in some of the subgroups. The results are presented in Tables 7, 8 and 9 in Appendix.

V. CONCLUDING REMARKS

We conclude from the above results that our novel DNNs model can be used to predict haplogroups accurately, and that our novel feature extraction method reduced the model's running time without degrading the prediction performance.

TABLE 9. Effect of variation in number of features on reverse feature set of test dataset (76 classes). Standard deviation given in brackets.

Features	Loss	Accuracy	Precision	Recall
512	0.705 (0.064)	0.739 (0.029)	0.830 (0.024)	0.682 (0.029)
1024	0.604 (0.049)	0.812 (0.021)	0.853 (0.008)	0.779 (0.021)
2048	0.389 (0.053)	0.891 (0.017)	0.917 (0.015)	0.866 (0.023)
4096	0.321 (0.055)	0.921 (0.011)	0.934 (0.010)	0.899 (0.018)

Second, the prime-number-based feature sets can be used to achieve practical performance results where the results are similar. This is because the forward feature set gave better results for the training dataset, and the reverse feature set gave better results for the test dataset. We believe this can be attributed to the position of SNPs that indicate whether a mutation has occurred. As a result, if the same positions were chosen, a mutation pattern might not be seen as the SNPs are currently being represented as a sequence of 0s and 1s. Third, we showed that the accuracy improves as more features are used. This further indicates that fewer SNPs than the callable SNPs described in [56] may be used to differentiate haplogroups.

In this paper, we have 1) proposed a novel SNP-based DNNs model that learns patterns between haplogroups and subgroups; 2) proposed a novel feature extraction method based on prime numbers that reduces the computational complexity of DNNs; and 3) provided a comprehensive analysis of the experimental results that show patterns learned in either direction (forward or reverse) of the SNPs data can be used for haplogroup and sub-groups predictions.

APPENDIX
EXPERIMENTAL RESULTS ON THE 76 SUBGROUPS OF THE 1000 GENOMES DATA

See Table 7–9.

REFERENCES

[1] J. Schlecht, M. E. Kaplan, K. Barnard, T. Karafet, M. F. Hammer, and N. C. Merchant, "Machine-learning approaches for classifying haplogroup from Y chromosome STR data," *PLoS Comput. Biol.*, vol. 4, no. 6, Jun. 2008, Art. no. e1000093.

[2] B. Emmerova, E. Ehler, D. Comas, J. Votrubova, and D. Vanek, "Comparison of Y-chromosomal haplogroup predictors," *Forensic Sci. Int., Genet. Suppl. Ser.*, vol. 6, pp. e145–e147, Dec. 2017.

[3] H. Xu, C.-C. Wang, R. Shrestha, L.-X. Wang, M. Zhang, Y. He, J. R. Kid, K. K. Kidd, L. Jin, and H. Li, "Inferring population structure and demographic history using Y-STR data from worldwide populations," *Mol. Genet. Genomics*, vol. 290, pp. 141–150, Aug. 2015.

[4] T. W. Athey, "Haplogroup prediction from Y-STR values using an allele-frequency approach," *J. Genetic Geneal.*, vol. 1, pp. 1–7, Jan. 2005.

[5] J. Jannuzzi, J. Ribeiro, C. Alho, G. de Oliveira Lázaro e Arão, R. Cicarelli, H. Simões Dutra Corrêa, S. Ferreira, C. Fridman, V. Gomes, S. Loiola, M. F. da Mota, Â. Ribeiro-dos-Santos, C. A. de Souza, R. S. de Sousa Azulay, E. F. Carvalho, and L. Gusmão, "Male lineages in Brazilian populations and performance of haplogroup prediction tools," *Forensic Sci. Int., Genet.*, vol. 44, Jan. 2020, Art. no. 102163.

[6] J. Cullen. *World Haplogroup & Haplo-I Subclade Predictor*. Accessed: Jul. 12, 2020. [Online]. Available: https://members.bex.net/jtcullen515/haplotest.htm

[7] Nevgen. *Y-DNA Haplogroup Predictor—Nevgen*. Accessed: Jul. 12, 2020. [Online]. Available: https://www.nevgen.org

[8] G. D. Poznik, "Identifying Y-chromosome haplogroups in arbitrarily large samples of sequenced or genotyped men," *BioarXiv*, Nov. 2016, Art. no. 088716.

[9] P. Kulkarni and P. Joshi, *Artificial Intelligence Building Intelligent System*. New Delhi, India: PHI Learning Private Limited, 2015.

[10] P. A. Jacobs, M. Brunton, W. M. C. Brown, R. Doll, and H. Goldstein, "Change of human chromosome count distribution with age: Evidence for a sex differences," *Nature*, vol. 197, p. 1080, Mar. 1963.

[11] A. K. Wong, B. Fang, L. Zhang, X. Guo, S. Lee, and R. Schreck, "Loss of the Y chromosome: An age-related or clonal phenomenon in acute myelogenous leukemia/myelodysplastic syndrome?" *Arch. Pathol. Lab. Med.*, vol. 132, no. 8, pp. 1329–1332, 2008.

[12] UKCCG, "Loss of the Y chromosome from normal and neoplastic bone marrows. United Kingdom cancer cytogenetics group (UKCCG)," *Genes, Chromosome Cancer*, vol. 5, no. 1, pp. 83–88, 1992.

[13] L. Forsberg, C. Rasi, N. Malmqvist, H. Davies, S. Pasupulati, G. Pakalapati, J. Sandgren, T. D. de Ståhl, A. Zaghlool, V. Giedraitis, L. Lannfelt, J. Score, N. C. P. Cross, D. Absher, E. T. Janson, C. M. Lindgren, A. P. Morris, E. Ingelsson, L. Lind, and J. Dumanski, "Mosaic loss of chromosome Y in peripheral blood is associated with shorter survival and higher risk of cancer," *Nature Genet.*, vol. 46, pp. 624–628, Apr. 2014.

[14] N. O. Bianchi, "Y chromosome structural and functional changes in human malignant diseases," *Mutation Res./Rev. Mutation Res.*, vol. 682, no. 1, pp. 21–27, 2009.

[15] S. Hunter, T. Gramlich, K. Abbott, and V. Varma, "Y chromosome loss in esophageal carcinoma: An *in situ* hybridization study," *Genes Chromosomes Cancer*, vol. 8, no. 3, pp. 172–177, 1993.

[16] S.-J. Park, S.-Y. Jeong, and H. J. Kim, "Y chromosome loss and other genomic alterations in hepatocellular carcinoma cell lines analyzed by CGH and CGH array," *Cancer Genet. Cytogenetics*, vol. 166, no. 1, pp. 56–64, Apr. 2006.

[17] P. H. Duijf, N. Schultz, and R. Benezra, "Cancer cells preferentially lose small chromosomes," *Int. J. Cancer*, vol. 132, no. 10, pp. 2316–2326, 2012.

[18] D. L. Bichile, A. R. Kharkar, P. Menon, M. Potnis-Lele, M. Bankar, and G. A. Shroff, "Y chromosome: Structure and biological functions," *Indian J. Basic Appl. Med. Res.*, vol. 3, no. 3, pp. 152–160, 2014.

[19] M. Jobling and C. Tyler-Smith, "Human Y-chromosome variation in the genome-sequencing era," *Nature Rev. Genet.*, vol. 18, no. 8, pp. 485–497, 2017.

[20] M. A. Jobling, A. Pandya, and C. Tyler-Smith, "The Y chromosome in forensic analysis and paternity testing," *Int. J. Legal Med.*, vol. 110, pp. 118–124, Jun. 1997.

[21] M. A. Jobling and C. Tyler-Smith, "The human Y chromosome: An evolutionary marker comes of age," *Nature Rev. Genet.*, vol. 4, no. 8, pp. 598–612, Aug. 2003.

[22] T. Lappalainen, S. Koivumäki, E. Salmela, K. Huoponen, P. Sistonen, M.-L. Savontaus, and P. Lahermo, "Regional differences among the Finns: A Y-chromosomal perspective," *Gene*, vol. 376, no. 2, pp. 207–215, 2006.

[23] J. Di Cristofaro, S. Mazières, A. Tous, C. Di Gaetano, A. A. Lin, P. Nebbia, A. Piazza, R. J. King, P. Underhill, and J. Chiaroni, "Prehistoric migrations through the mediterranean basin shaped corsican Y-chromosome diversity," *PLoS ONE*, vol. 13, no. 8, Aug. 2018, Art. no. e0200641.

[24] D. Primorac and M. Schanfield, "Forensic DNA applications: An interdisciplinary perspective–a new book in forensic science," *Croatian Med. J.*, vol. 55, no. 4, pp. 434–436, 2014.

[25] Y. Sato, T. Shinka, T. Iwamoto, A. Yamauchi, and Y. Nakahori, "Y chromosome haplogroup D2* lineage is associated with Azoospermia in Japanese Males1," *Biol. Reproduction*, vol. 88, no. 4, pp. 1–5, Apr. 2013.

[26] J. Ran, T. T. Han, X. P. Ding, X. Wei, L. Y. Zhang, Y. P. Zhang, T. J. Li, S. S. Nie, and L. Chen, "Association study between Y-chromosome haplogroups and susceptibility to spermatogenic impairment in Han people from Southwest China," *Genet. Mol. Res.*, vol. 12, no. 1, pp. 59–66, 2013.

[27] A. Puzuka, N. Pronina, I. Grinfelde, J. Erenpreiss, V. Lejing, J. Bars, L. Pliss, I. Pelnena, V. Baumanis, and A. Krumina, "Y chromosome—A tool in infertility studies of Latvian population," *Genetika*, vol. 47, pp. 394–400, Mar. 2011.

[28] Y. Yang, M. Ma, L. Li, W. Zhang, C. Xiao, S. Li, Y. Ma, D. Tao, Y. Liu, L. Lin, and S. Zhang, "Evidence for the association of Y-chromosome haplogroups with susceptibility to spermatogenic failure in a Chinese Han population," *J. Med. Genet.*, vol. 45, no. 4, pp. 210–215, 2008.

[29] Y. Sato, T. Iwamoto, T. Shinka, S. Nozawa, M. Yoshiike, E. Koh, J. Kanaya, M. Namiki, K. Matsumiya, A. Tsujimura, K. Komatsu, N. Itoh, J. Eguchi, A. Yamauchi, and Y. Nakahori, "Y chromosome gr/gr subdeletion is associated with lower semen quality in young men from the general Japanese population but not in fertile Japanese Men1," *Biol. Reproduction*, vol. 90, no. 6, pp. 1–8, Jun. 2014.

[30] L. Bloomer, C. Nelson, J. Eales, M. Denniff, P. Christofidou, R. Debiec, J. Moore, C. Consortium, E. Zukowska-Szczechowska, A. H. Goodall, J. Thompson, N. J. Samani, F. J. Charchar, and M. Tomaszewski, "Male-specific region of the Y chromosome and cardiovascular risk: Phylogenetic analysis and gene expression studies," *Arteriosclerosis, Thrombosis, Vascular Biol.*, vol. 33, no. 7, pp. 1722–1727, 2013.

[31] G. Kostrzewa, G. Broda, M. Konarzewska, P. Krajewki, and R. Płoski, "Genetic polymorphism of human Y chromosome and risk factors for cardiovascular diseases: A study in WOBASZ cohort," *PLoS ONE*, vol. 8, no. 7, Jul. 2013, Art. no. e68155.

[32] F. Charchar *et al.*, "Inheritance of coronary artery disease in men: An analysis of the role of the Y chromosome," *Lancet*, vol. 379, no. 9819, pp. 915–922, 2012.

[33] F. Charchar, M. Tomaszewski, S. Padmanabhan, B. Lacka, M. Upton, G. Inglis, N. H. Anderson, A. McConnachie, E. Zukowska-Szczechowska, W. Grzeszczak, J. M. C. Connell, G. C. M. Watt, and A. Dominiczak, "The Y chromosome effect on blood pressure in two European populations," *Hypertension*, vol. 39, no. 2, pp. 353–356, 2002.

[34] S. Lindstrom, H.-O. Adami, J. Adolfsson, and F. Wiklund, "Y chromosome haplotypes and prostate cancer in Sweden," *Clin. Cancer Res.*, vol. 14, no. 20, pp. 6712–6716, 2008.

[35] Z. Wang *et al.*, "Y chromosome haplogroups and prostate cancer in populations of European and Ashkenazi Jewish ancestry," *Hum. Genet.*, vol. 131, no. 7, pp. 1173–1185, 2012.

[36] C.-C. Wang, L.-X. Wang, R. Shrestha, S. Wen, M. Zhang, X. Tong, L. Jin, and H. Li, "Convergence of Y chromosome STR haplotypes from different SNP haplogroups compromises accuracy of haplogroup prediction," *J. Genet. Genomics*, vol. 42, no. 7, pp. 403–407, 2015.

[37] E. Petrejcíková, J. Čarnogurská, D. Hronská, J. Bernasovská, I. Boronova, D. Gabriková, A. Bozikova, and S. Mačeková, "Y-SNP analysis versus Y-haplogroup predictor in the Slovak population," *Anthropologischer Anzeiger*, vol. 71, no. 3, pp. 275–285, 2014.

[38] C. Núñez, M. Geppert, M. Baeta, L. Roewer, and B. Martínez-Jarreta, "Y chromosome haplogroup diversity in a Mestizo population of Nicaragua," *Forensic Sci. Int., Genet.*, vol. 6, no. 6, pp. e192–e195, Dec. 2012.

[39] M. Muzzio, V. Ramallo, J. Motti, M. Santos, J. S. L. Camelo, and G. Bailliet, "Software for Y-haplogroup predictions: A word of caution," *Int. J. Legal Med.*, vol. 125, no. 1, pp. 143–147, 2011.

[40] M. Singh, A. Sarkar, and M. R. Nandineni, "A comprehensive portrait of Y-STR diversity of Indian populations and comparison with 129 worldwide populations," *Sci. Rep.*, vol. 8, no. 1, pp. 1–7, Dec. 2018.

[41] T. I. Huszar, M. A. Jobling, and J. H. Wetton, "A phylogenetic framework facilitates Y-STR variant discovery and classification via massively parallel sequencing," *Forensic Sci. Int., Genet.*, vol. 35, pp. 97–106, Jul. 2018.

[42] M. Song, C. Zhao, Z. Wang, and Y. Hou, "Applying machine learning algorithms to a real forensic case to predict Y-SNP haplogroup based on Y-STR haplotype," *Forensic Sci. Int., Genet. Suppl. Ser.*, vol. 7, no. 1, pp. 637–638, 12 2019.

[43] S. Zhao, W. Jing, D. C. Samuels, Q. Sheng, Y. Shyr, and Y. Guo, "Strategies for processing and quality control of illumina genotyping arrays," *Briefings Bioinf.*, vol. 19, no. 5, pp. 765–775, 2018.

[44] S. Purcell, B. Neale, K. Todd-Brown, L. Thomas, M. A. R. Ferreira, D. Bender, J. Maller, P. I. W. de Bakker, M. J. Daly, and P. C. Sham, "PLINK: A tool set for whole-genome association and population-based linkage analyses," *Amer. J. Hum. Genet.*, vol. 81, no. 3, pp. 559–575, 2007.

[45] Illumina. (2020). *Simple Guidelines for Identifying Top/Bottom (Top/Bot) Strand and A/B Allele*. Accessed: Jun. 27, 2020. [Online]. Available: https://emea.support.illumina.com/bulletins/2016/06/simple-guidelines-for-identifying-topbottom-topbot-strand-and-ab-allele.html

[46] I. S. of Genetic Genealogy. (2020). *Y-DNA Haplogroup Tree 2019-2020*. Accessed: Jul. 24, 2020. [Online]. Available: http://www.isogg.org/tree/

[47] M. van Oven, A. Van Geystelen, M. Kayser, R. Decorte, and M. Larmuseau, "Seeing the wood for the trees: A minimal reference phylogeny for the human Y chromosome," *Hum. Mutation*, vol. 35, no. 2, pp. 187–191, 2014.

[48] B. Tang, Z. Pan, K. Yin, and A. Khateeb, "Recent advances of deep learning in bioinformatics and computational biology," *Frontiers Genet.*, vol. 10, p. 214, Mar. 2019.

[49] L. Bottou, O. Chapelle, D. DeCoste, and J. Weston, "Scaling learning algorithms toward AI," in *Large-Scale Kernel Machines*. 2007, pp. 321–359.

[50] S. Zhang, J. Zhou, H. Hu, H. Gong, L. Chen, C. Cheng, and J. Zeng, "A deep learning framework for modeling structural features of RNA-binding protein targets," *Nucleic Acids Res.*, vol. 44, no. 4, p. e32, Feb. 2016.

[51] B. Alipanahi, A. Delong, M. Weirauch, and B. Frey, "Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning," *Nature Biotechnol.*, vol. 33, pp. 831–838, Jul. 2015.

[52] M. Libbrecht and W. Noble, "Machine learning applications in genetics and genomics," *Nature Rev. Genet.*, vol. 16, no. 6, pp. 321–332, 2015.

[53] A. Esteva, B. Kuprel, R. Novoa, J. Ko, S. Swetter, H. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, pp. 115–118, Jan. 2017.

[54] T. Ching *et al.*, "Opportunities and obstacles for deep learning in biology and medicine," *J. Roy. Soc. Interface*, vol. 15, no. 141, p. 47, 2018.

[55] L. Quintana-Murci and M. Fellous, "The human Y chromosome: The biological role of a 'functional wasteland,'" *J. Biomed. Biotechnol.*, vol. 1, no. 1, pp. 18–24, 2001.

[56] G. D. Poznik, B. M. Henn, M.-C. Yee, E. Sliwerska, G. M. Euskirchen, A. A. Lin, M. Snyder, L. Quintana-Murci, J. M. Kidd, P. A. Underhill, and C. D. Bustamante, "Sequencing Y chromosomes resolves discrepancy in time to common ancestor of males versus females," *Science*, vol. 341, no. 6145, pp. 562–565, 2013.

[57] M. Jobling, E. Hollox, M. Hurles, T. Kivisild, and C. Tyler-Smith, *Human Evolutionary Genetics*, 2nd ed. New York, NY, USA: Taylor & Francis Group, LLC, 2014.

[58] Y. Chromosome Consortium, "A nomenclature system for the tree of human Y-chromosomal binary haplogroups," *Genome Res.*, vol. 12, no. 2, pp. 339–348, 2002.

[59] T. M. Karafet, F. L. Mendez, M. B. Meilerman, P. A. Underhill, S. L. Zegura, and M. F. Hammer, "New binary polymorphisms reshape and increase resolution of the human Y chromosomal haplogroup tree," *Genome Res.*, vol. 18, no. 5, pp. 830–838, 2008.

[60] A. Auton, "A global reference for human genetic variation," *Nature*, vol. 526, pp. 68–74, Sep. 2015.

[61] M.-S. Peng, J.-D. He, L. Fan, J. Liu, A. Adeola, S.-F. Wu, R. W. Murphy, and Y.-P. Zhang, "Retrieving Y chromosomal haplogroup trees using GWAS data," *Eur. J. Hum. Genet.*, vol. 22, no. 8, pp. 1046–1050, 2014.

[62] R. Gibbs *et al.*, "The international HapMap project," *Nature*, vol. 426, pp. 789–796, 2003.

[63] A. Pasini, "Artificial neural networks for small dataset analysis," *J. Thoracic Disease*, vol. 7, no. 5, pp. 953–960, 2015.

[64] P. Liashchynskyi and P. Liashchynskyi, "Grid search, random search, genetic algorithm: A big comparison for NAS," 2019, *arXiv:1912.06059*. [Online]. Available: https://arxiv.org/abs/1912.06059

[65] Y. Bengio, *Practical Recommendations for Gradient-Based Training Deep Architectures*. Berlin, Germany: Springer, 2012, pp. 437–478.

[66] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org

[67] D. Masters and C. Luschi, "Revisiting small batch training for deep neural networks," 2018, *arXiv:1804.07612*. [Online]. Available: https://arxiv.org/abs/1804.07612

[68] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation functions: Comparison of trends in practice and research for deep learning," 2018, *arXiv:1811.03378*. [Online]. Available: https://arxiv.org/abs/1811.03378

[69] S. J. Rogers, J. Fang, C. Karr, and D. Stanley, "Determination of lithology from well logs using a neural network," *Amer. Assoc. Petroleum Geologists Bull.*, vol. 76, no. 5, pp. 731–739, 1992.

[70] P. Wang, H. Ni, and R. Wang, "A novel vibration drilling tool used for reducing friction and improve the penetration rate of petroleum drilling," *J. Petroleum Sci. Eng.*, vol. 165, pp. 436–443, Jun. 2018.

[71] R. A. Azim, "Application of artificial neural network in optimizing the drilling rate of penetration of western desert Egyptian wells," *Social Netw. Appl. Sci.*, vol. 2, no. 7, pp. 1–13, Jul. 2020.

[72] K. G. Sheela and S. N. Deepa, "Review on methods to fix number of hidden neurons in neural networks," *Math. Problems Eng.*, vol. 2013, pp. 1–11, Jun. 2013.

[73] G. Aksu, C. O. Güzeller, and M. T. Eser, "The effect of the normalization method used in different sample sizes on the success of artificial neural network model," *Int. J. Assessment Tools Edu.*, vol. 6, no. 2, pp. 170–192, 2019.

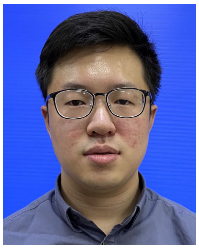
[74] B. Karlik and A. Vehbi, "Performance analysis of various activation functions in generalized MLP architectures of neural networks," *Int. J. Artif. Intell. Expert Syst.*, vol. 1, no. 4, pp. 111–122, 2011.

[75] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[76] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd ICML*, vol. 37, F. R. Bach and D. M. Blei, Eds. Brookline, MA, USA, JMLR, Inc. and Microtome Publishing, 2015, pp. 448–456.

[77] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 56, pp. 1929–1958, 2014.

JASBIR DHALIWAL received the B.Sc. degree (Hons.) in computer science from the University of Malaya, in 2004, and the M.Sc. degree in applied science and Ph.D. degree from the Royal Melbourne Institute of Technology (RMIT) University, Australia, in 2008 and 2013, respectively. She worked as a Software Engineer with Motorola, Malaysia. She worked as a Postdoctoral Researcher with IBM Research, Australia. She worked as a Data Scientist with FTI Consulting, Australia. She is currently a Lecturer with Monash University Malaysia. Her research interests include bioinformatics, computer vision, machine learning, deep learning, and string related algorithms.



KEONG JIN received the B.Sc. degree (Hons.) in computer science from the University of Monash Malaysia, in 2020. He was a Research Assistant attached to the project. His research interest includes bioinformatics.

ZHE JIN (Member, IEEE) received the B.I.T. degree (Hons.) in software engineering and the M.Sc. (IT) degree from Multimedia University, Malaysia, in 2007 and 2011, respectively, and the Ph.D. degree in engineering from University Tunku Abdul Rahman Malaysia, in 2016. He visited the University of Salzburg, Austria, and the University of Sassari, Italy, respectively, as a Visiting Scholar, under the EU Project IDENTITY. He is currently a Senior Lecturer with the School of Information Technology, Monash University Malaysia. He has published more than 40 refereed journals, conference papers, including the IEEE Transactions on Information Forensics and Security, SMC-S, DSC, and *Pattern Recognition*. His research interests include biometric security, computer vision, and machine learning. He received the Marie Skłodowska-Curie Research Exchange Fellowship.