Journal of Intelligent & Fuzzy Systems xx (20xx) x–xx
DOI:10.3233/JIFS-220990
IOS Press

On intelligent *Prakriti* assessment in Ayurveda: a comparative study

Saibal Majumder[a], Rintu Kutum[b], Debnarayan Khatua[c], Arif Ahmed Sekh[d], Samarjit Kar[e,f,*], Mitali Mukerji[g] and Bhavana Prasher[h]
[a]*Department of Computer Science and Engineering (Data Science), Dr. B. C. Roy Engineering College, Durgapur*
[b]*Department of Computer Science, Ashoka University, Haryana*
[c]*Department of Mathematics, Vignan's Foundation for Science, Technology & Research, Andhra Pradesh*
[d]*School of Computer Science, XIM University, Bhubaneshwar*
[e]*Department of Mathematics, National Institute of Technology Durgapur*
[f]*Department of Graphical Systems, Vilnius Gediminas Technical University, Lithuania*
[g]*Department of Bioscience and Bioengineering, Indian Institute of Technology Jodhpur*
[h]*Ayurgenomics Unit-TRISUTRA, CSIR-Institute of Genomics and Integrative Biology, New Delhi*

Abstract. Predictive medicine for a holistic and proactive approach to health management is steadily replacing the reactive healthcare model as the dominant paradigm in the twenty-first century. The Ayurvedic medical system, which incorporates all parts of predictive medicine, divides people into seven constitution types, or Prakriti, to help practitioners determine their initial homeostatic conditions. This article uses data on the phenotypic characteristics of 217 healthy people who fall into three extreme Prakriti types to conduct a study for predicting Prakriti classes. Those who fit the Prakriti type are drawn from two genetically different northern and western India cohorts. In order to dichotomize inter-individual variability in various individuals, eight machine learning (ML) classifiers are used. The prediction skills of the ML algorithms are evaluated here using ten pairs of predefined training and testing datasets for each cohort. Lastly, a performance comparison of various ML algorithms is carried out using six crucial performance criteria.

The study aims to investigate and appraise using artificial intelligence (AI) to evaluate Prakriti in Ayurveda. The use of AI in Prakriti assessment may have several advantages, including enhancing the consistency and accuracy of assessments and minimizing reliance on subjective judgements. This study aims to further our knowledge of how technology can be applied to enhance the practice of Ayurveda and possibly improve patient outcomes.

Keywords: Ayurveda, Ayurgenomics, classification, performance metrics

1. Introduction

According to Panday et al. [1], Ayurveda is a comprehensive, all-natural system of medicine that has its roots in ancient India's Vedic era. The words "Ayurveda" and "Veda," which denote science and knowledge, respectively, come from the Sanskrit language. Ayurveda, when taken as a whole, is known as "the science of life" or "the science of lifespan." Regrettably, many Ayurveda and Vedic notions lack adequate definitions considering current understanding, which leads to divergent interpretations in contemporary discourse [2].

Ayurveda, the oldest holistic medical system in India that has been documented and used since 1500 B.C., uses a tailored approach to care for patients along with a focus on health promotion [3–6]. Individuals are categorized in this medical system according to their Prakriti constitution kinds. Every

*Corresponding author. Samarjit Kar, E-mail: samarjit.kar@maths.nitdgp.ac.in.

ISSN 1064-1246/$35.00

individual has a unique genetic makeup, or Prakriti [7]. These constitution types are divided into seven groups based on how susceptible they are to certain diseases and environmental factors. This approach effectively shows promise for predicting a person's trajectory. The three groups Vata (V), Pitta (P), and Kapha (K) are at the extremes of the phenotypic spectrum among the seven constitution types. A person can be classified as belonging to a certain Dosha type depending on the attributes that each Dosha bestows upon them [8–10]. They are said to be more susceptible to many diseases [11]. A Prakriti classification like this enables practitioners to identify the causes of patients' homeostatic states, evaluate disturbances caused by illness states (Vikriti), and suggest individualised therapy for reestablishing balance [4,5]. For optimum health, these three Doshas must be in harmony [12]. In this context, the Prakriti types V, P, and K are referred to as extreme (distinct) Prakriti, whereas VP, PK, VK, and VPK are non-extreme Prakriti. This is because research [13, 14, 2] has shown that the extreme Prakriti kinds differ molecularly from one another. Prakriti and tridoshas include the fundamentals of unique ayurvedic principles that can be applied in prognosis treatments, but it is essential to develop their molecular underpinnings [15]. As a result, Prakriti's phenotypic classification is based on anatomical characteristics such as body build, physiology, and physical stamina as well as size and symmetry of body components [16,17]. Researchers from the Ministry of Science and Technology's CSIR-Institute of Genomics and Integrative Biology (CSIR-IGIB) in New Delhi believe there is a chance to examine whether the Prakriti-based classification of people has a genetic basis [18].

One of the key features of the Ayurvedic medical system is its ability to divide people into groups depending on their prevailing Prakriti. This aids in promoting health, preventing disease, and treating illness by assisting in understanding a person's susceptibility to diseases in addition to their mental and physical makeup. It should be noted that the Ayurveda system works to detect the imbalance of the Tridoshas to cure the disease's root cause rather than just its symptoms [19]. The present classification of human phenotypes, which considers individual system qualities such as somatotypes for anthropometric attributes, phototypes for skin phenotypes, and chronotypes for early and late risers, is considerably different from this method. We have created a new framework called Ayurgenomics that combines Ayurvedic philosophy with genomics. To determine the common correlates of Prakriti, ayurgenomics-based phenotyping has been carried out in numerous ethnic populations along with other objective measurements. The development of analytical techniques for the impartial evaluation of Prakriti is one of the goals. This might make it possible to apply it in contexts with varied populations both domestically and abroad. Our study uses a questionnaire based on accurate textual descriptions to assess Prakriti formally. The questionnaire offers numerous possibilities for each aspect, each mapped to V, P, or K based on textual descriptions. In this line, Datar and Murthy [20] developed a Prakriti-issuing questionnaire and presented it as a reliable validity instrument for Prakriti prediction.

Today, ML has gotten much traction thanks to improvements in processing power and the accessibility of an unprecedented amount of data in the public domain. As a result, ML approaches have been used in various fields, including science, engineering, medicine, finance, and academia. ML is mainly used in medicine to assist doctors in identifying and diagnosing diseases and developing individualized remedies [21,22]. Nevertheless, Tiwari et al. [23] conducted a study in two genetically homogenous cohorts from northern and western India that focused on most Prakriti individuals. Powerful, dense neural network deep learning methods have recently been utilized for the first time by Khatua et al. [24] to predict Prakriti courses. Here, the authors have classified the individuals using K, V, or P for the first time using three ML models. The regression framework of the LASSO model is utilized for extreme Prakriti modelling in their study. In their study, the scientists also utilized an elastic net method for keeping correlated data and predicting non-redundant factors that might not be able to distinguish severe Prakriti persons from non-extreme ones on their own. To the best of our knowledge, there has yet to be any research comparing the effectiveness of various ML classifiers for identifying an individual's dominant Prakriti. As a result, we compare eight supervised machine learning methods on two genetically homogenous cohorts from northern and western India in this research. Figure 1 shows a pipeline diagram of the whole study. We then briefly discuss our study's primary contributions, which are listed below.

(i) Eight ML classifiers, including the ridge classifier (RC), multinomial nave Bayes classifier (MNBC), random forest classifier (RFC), extra tree classifier (ETC), v-support vector classi-

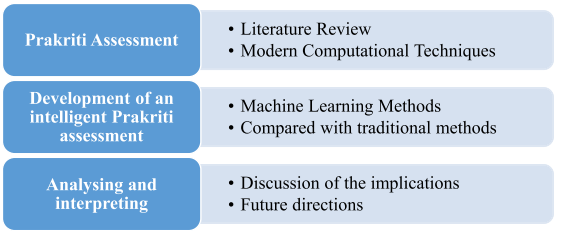

Fig. 1. A pipeline diagram of the study.

fier (v-SVC), the passive-aggressive classifier (PAC), stochastic gradient descent classifier (SGDC), and logistic regression with cross-validation classifier (LRCVC), are used on the North and West Indian cohorts of people for the comparative study.

(ii) These ML classifiers are validated with a cross-dataset setup.

(iii) The performance of these classifiers is analyzed concerning six performance metrics, including accuracy, precision, f1-score, area under the receiver operating characteristic curve score (AUROCCS) score, Matthews correlation coefficient (MCC) and hamming loss (HL).

The rest of the article is organized as follows. In section 2, discussed about the aims and motivation of our work. In section 3, we briefly discuss eight different machine learning models. Consequently, in section 4, the performance metrics used in our study are discussed concisely. A detailed discussion of the results and their analysis are provided in sections 5 and 6. Finally, the culmination of the study is presented in section 7.

2. Aims and motivations

The study project aims to examine the Ayurvedic notion of Prakriti and compare conventional methods of Prakriti assessment with contemporary computational methodologies. A person's individual set of physical, mental, and emotional traits are referred to as their Prakriti, and it is thought that both their genetic make-up and their environment have a role in this.

The research aims to close the knowledge gap between conventional Ayurveda wisdom and contemporary scientific evaluation techniques. The goal of the project is to create an intelligent Prakriti evaluation system that blends modern computational methods like machine learning and artificial intelligence with knowledge of Ayurveda. This would make it possible to diagnose and treat patients with greater precision and individualization depending on their Prakriti.

The goal of this research project is to increase the efficacy of Ayurveda treatment by giving its traditional practices a scientific foundation and making them more approachable to a larger audience. It also seeks to advance customized medicine by investigating the application of ancient knowledge systems in concert with contemporary scientific methods.

3. Machine learning models

With the progressive improvement of computational capability of processing units and availability of an unprecedented amount of data in the public domain, machine learning has gained colossal attention in applications across diverse fields. One of the essential concepts of ML is supervised learning. In supervised learning, we use an ML algorithm to learn the mapping function between the input (X) and output (Y) variables such that $Y = f(X)$. Here, the goal is to approximate the mapping function so that the algorithm can predict the out variables (Y) for a new input data (X).

In this article, we have compared the performance of eight supervised machine learning models on the datasets of North and West Indian cohorts. We have used these following models:

Ridge classifier (RC) uses the concept of ridge regression [25] which eventually opened the door of penalty estimators based on Tikhonov [26] regularization. This method solves a regression model, where the minimization of least squares is done subject to l_2 penalty.

Multinomial naïve Bayes classifier (MNBC) [27, 28] is one of the variants of Naïve Bayes classifier [29]. This classifier models the data by assuming the underline distribution of the data follows the multinomial distribution.

A random forest classifier (RFC) [30] is an ensemble of decision trees where a prediction is made collectively by several decision trees. Here, each tree in the ensemble is formed from a sample of the training set, which is drawn with replacement.

Extremely randomized trees or extra trees classifier (ETC) [31] is an ensemble of decision trees like random forests which essentially creates many unpruned decision trees from the training data and performs predictions employing a majority vote of decision

trees. The extra trees algorithm fits each decision tree on the whole training data, which is very much different from the random forest, where each decision tree is created from a bootstrap sample of training data.

nu-Support Vector Classifier(ν-SVC) is one of the variants of a support vector classifier, which is introduced by Scholkopf et al. [32]. This variant of the support vector machine (SVM) algorithm is essentially used to govern the maximum separation between the subsets of the convex hulls of the data, which are usually known as soft convex hulls. These soft convex hulls are generally controlled by the value of the parameter.

Passive aggressive algorithms [33] are a family of algorithms that perform online learning of massive streams of data. In online machine learning algorithms like Passive Aggressive Classifier (PAC), the ML model is updated in a step-by-step fashion with respect to the sequential arrival of the input streams of data.

Stochastic gradient descent classifier (SGDC) [34] algorithm is an optimization technique which is used to train an ML model and essentially does not correspond to a specific family of ML models. In situations where there is a large amount of data in-hand, the stochastic gradient descent (SGD) algorithm is generally used.

Logistic Regression with Cross Validation Classifier (LRCVC) [35] is considered as the linear model of classification, which help us to explore the relationships between dependent and independent variables.

In Fig. 2, details our overall study methodology for the paper.

4. Performance metrics for classification

In this section, we briefly discuss six performance metrics which are used to measure the performance of the classifiers.

Accuracy: Accuracy measures the ability of a classifier to classify all instances correctly. It is calculated as the ratio of the number of correct predictions and the total predictions made by a classifier. Accuracy may not be useful for large class imbalance problems, where a classifier can achieve higher accuracy by predicting the majority class values for all the predictions. The mathematical expression of accuracy as

$$accuracy\,(c, \hat{c}) = \frac{1}{n}\sum_{k=0}^{n-1} 1\,(\hat{c}_k = c_k)\,, \qquad (2)$$

where n is the total number of samples predicted, c_k and $\hat{c}_k$ are the actual and predicted values of the

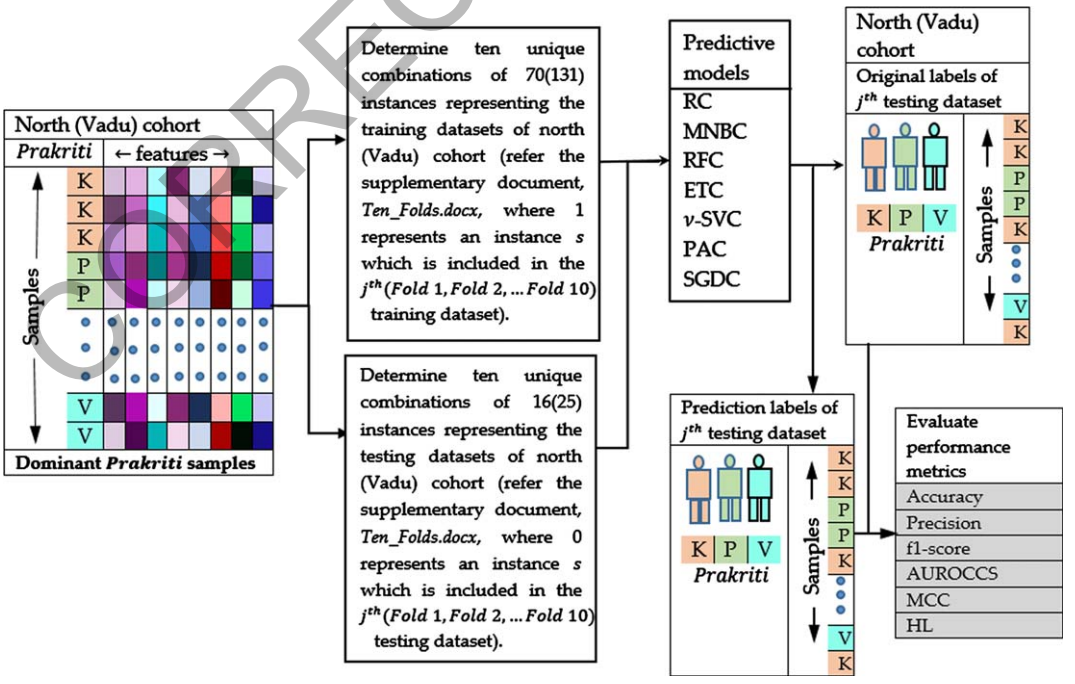

Fig. 2. Study method diagram of the modelling strategies of the eight ML estimators.

k^{th} sample, and 1 $(\cdot)$ is the indicator function. Higher value of accuracy is desirable.

Precision: Precision measures the ability of a classifier to correctly classify the positive labels among all the instances predicted as positive. In other words, precision can be expressed as the accuracy of the predictions of positive levels. It is expressed as

$$Precision = \frac{TP}{TP + FP}, \tag{3}$$

where TP and FP are the number of true positive instances and number of false positive instances, respectively. A classifier with higher precision value is always preferred.

f1-score: f1-score is the measure which combines precision and recall. It is to be mentioned that recall is also known as true positive rate, i.e., the ratio of the correctly predicted positive instances, and the sum of falsely predicted negative instances and the correctly predicted positive instances, i.e.,

$$Recall = \frac{TP}{FN + TP}, \tag{4}$$

where FN is the number of false negative instances.

Accordingly, the f1-score is determined by calculating the harmonic mean of precision and recall. Since the harmonic mean gives more weightage to minority class values, therefore a classifier will achieve a higher value only when both the precision and recall of the classifier are high. A higher value of f1-score is preferable. The f1-score is expressed as follows.

$$f1 - score = \frac{TP}{TP + \frac{FN+FP}{2}}. \tag{5}$$

Area under the Receiver Operating Characteristic Curve score (AUROCCS): The receiver operating characteristic (ROC) curve measures a classifier's performance by plotting the true positive rate (TPR) corresponding to false positive rate (FPR) by varying the discrimination threshold. Subsequently, the area under the ROC curve is computed by the AUROCCS in the form of a numeric value. The AUROCCS varies between 0 and 1. Here, a value close to 1 is always preferable, which implies a near perfect prediction of the classifier.

Matthews Correlation Coefficient (MCC): The Matthews correlation coefficient considers all the true and false positives and negatives to measure the prediction quality of a classifier. This metric works well even if the classification classes have indifferent sizes. Being a correlation coefficient, the value of MCC lies within the interval $[-1, +1]$. For a perfect prediction, the MCC takes the value+1. An average random prediction is implied if the value of MCC is 0. Whereas, for an inverse prediction value of MCC becomes –1. MCC can be expressed as

$$MCC = \frac{((TP \times TN) - (FP \times FN))}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}. \tag{6}$$

Hamming Loss (HL): Hamming loss determines the fraction of labels that are incorrectly predicted. The hamming loss is expressed as,

$$HL(c, \hat{c}) = \frac{1}{n} \sum_{k=0}^{n-1} 1(\hat{c}_k \neq c_k). \tag{7}$$

A smaller value of HL is always desirable.

5. Results

In this section, we conduct a comparative study of eight ML classifiers on ten predetermined training and testing datasets of two cohorts. Consequently,

Table 1
List of highly correlated features eliminated from the j^{th} training and its testing datasets of North and Vadu cohorts

# Training and its corresponding testing dataset	Features eliminated from the dataset of North Indian cohort	Features eliminated from the dataset of Vadu cohort
*Fold*1	F18, F31, F77, F99, F117, F118	F5, F27, F101
*Fold*2	F18, F31, F77, F99, F117	
*Fold*3	F13, F18, F31, F77, F99, F117, F118	
*Fold*4	F18, F31, F77, F99, F117, F118	
*Fold*5	F18, F31, F99, F117	
*Fold*6	F18, F31, F77, F99, F117, F118	
*Fold*7	F13, F18, F31, F77, F99, F117, F118	
*Fold*8	F18, F31, F77, F99, F117, F118	
*Fold*9	F13, F18, F31, F77, F99, F117	
*Fold*10	F18, F31, F77, F99, F117, F118	

the performance of the classifiers is analysed with respect to six performance metrics, as discussed in section 3. A detailed discussion related to this comparative study is provided in the subsequent subsections.

5.1. Dataset Creation

The datasets considered in this study are developed from the predominant Prakriti of the individuals belonging to two genetically homogeneous rural cohorts of northern and western India. Here, the cohort representing the Prakriti of the individuals from western India is referred to as Vadu cohort. The details about the formation of the North and Vadu cohorts can be found in the respective studies of Prasher et al. [13] and Tiwari et al. [23]. The original datasets of North Indian and Vadu cohorts are provided as supplementary documents, *Northern_India.csv* and *Westerrn_India.csv*, respectively. Here, the North Indian cohort consists of one hundred and five features and eighty-six individuals. Whereas the Vadu cohort comprises one hundred and thirty-one individuals, each having one hundred and thirty-three features. The feature set of the North cohorts is the proper subset of the feature set of the Vadu cohort. Apart from the feature GENDER, which is there in the west cohort, a feature of both the North and the Vadu cohorts is expressed as $F_{i,}$ which corresponds to the i^{th} feature of an individual belonging to a cohort. The mapping of each F_i with that of an actual feature name is reported in the supplementary document, *Supplementary IV.docx*. Subsequently, from each dataset of North and Vadu cohorts, ten different combinations of predetermined training and testing datasets are considered for the prediction purpose of the ML models. Each of these training and testing pairs of datasets of the North Indian cohort consists of seventy-six instances and sixteen instances of the individuals, respectively. Moreover, for the Vadu cohort, each of the training and testing datasets correspondingly contains one hundred and six instances and twenty-five instances. This information can be well observed from the supplementary document, *Supplementary I.docx*. For each of the Table's SI-1 and SI-2 provided in this file, the first column gives information about the instances number of a particular dataset. Whereas the data pre-

Table 2
Number of optimal features selected from the j^{th} training and its testing dataset of *north_train_test* for all the eight classifiers. **Red** represents highest and **blue** represents lowest number of features

# Training and its corresponding testing dataset	Optimal number of features							
	RC	MNBC	RFC	ETC	ν-SVC	PAC	SGDC	LRCVC
*Fold*1	10	91	42	50	18	15	08	13
*Fold*2	19	97	03	86	18	16	59	14
*Fold*3	07	75	06	62	18	23	20	21
*Fold*4	66	95	03	29	21	43	39	12
*Fold*5	36	93	14	06	16	18	13	14
*Fold*6	30	94	03	49	25	10	89	11
*Fold*7	57	96	03	51	29	05	08	04
*Fold*8	42	96	18	11	19	04	51	57
*Fold*9	83	94	06	85	62	09	27	48
*Fold*10	13	91	06	31	22	21	17	14

Table 3
List of optimal features selected from the j^{th} training and its testing dataset of *vadu_train_test* for all the eight classifiers. **Red** represents highest and **blue** represents lowest number of features

# Training and its corresponding testing dataset	Optimal number of features							
	RC	MNBC	RFC	ETC	ν-SVC	PAC	SGDC	LRCVC
*Fold*1	25	109	28	23	77	55	17	45
*Fold*2	34	129	23	12	74	52	45	25
*Fold*3	14	128	15	28	31	11	38	18
*Fold*4	22	77	19	41	50	12	15	118
*Fold*5	15	126	07	101	97	03	129	76
*Fold*6	40	113	14	117	32	73	85	47
*Fold*7	43	130	15	71	15	64	82	28
*Fold*8	07	127	06	47	20	101	13	70
*Fold*9	11	103	14	46	46	31	03	114
*Fold*10	05	96	34	22	18	104	113	39

Table 4
List of optimal features selected from the j^{th} training and its testing dataset of *north_vadu_train_test* for all the eight classifiers. **Red** represents highest and **blue** represents lowest number of features

# Training and its corresponding testing dataset	Optimal number of features							
	RC	MNBC	RFC	ETC	ν-SVC	PAC	SGDC	LRCVC
*Fold*1	24	91	42	50	51	49	36	13
*Fold*2	14	97	03	86	17	40	62	14
*Fold*3	12	76	06	62	36	23	21	06
*Fold*4	09	95	03	29	21	55	39	27
*Fold*5	38	89	14	81	32	18	13	17
*Fold*6	23	94	03	49	80	19	89	32
*Fold*7	12	96	03	56	57	06	33	19
*Fold*8	49	96	03	41	64	28	50	57
*Fold*9	78	94	69	44	62	47	27	48
*Fold*10	11	91	06	40	60	29	17	44

Table 5
List of common features selected by considering all the ten folds of training and testing datasets of *north_train_test*. Purple (F74) is the most important feature

Classifier	Selected features common to all the ten pairs of training and testing datasets
RC	F74
MNBC	F1, F3, F4, F6, F7, F9, F10, F11, F14, F15, F16, F17, F19, F20, F22, F23, F24, F25, F26, F28, F29, F30, F32, F33, F34, F35, F36, F37, F41, F42, F43, F44, F45, F46, F47, F48, F50, F51, F53, F54, F55, F59, F68, F69, F71, F72, F73, F75, F76, F78, F79, F80, F81, F82, F83, F84, F87, F88, F89, F90, F91, F92, F93, F100, F102, F104, F105, F106, F110, F113, F114, F115, F127, F131, F132
RFC	F5, F59, F74
ETC	F5, F34, F74, F85
ν-SVC	F5, F25, F33, F34, F37, F44, F74, F82, F85, F86, F101
PAC	F37, F44, F74, F86
SGDC	F5, F37, F44, F59, F74
LRCVC	F5, F37, F59, F74

Table 6
List of common features selected by considering all the ten folds of training and testing datasets of *vadu_train_test*. Purple (F2) is the most important feature

Classifier	Selected features common to all the ten pairs of training and testing datasets
RC	F2
MNBC	GENDER, F1, F3, F6, F7, F9, F10, F11, F12, F13, F15, F17, F18, F19, F21, F22, F23, F24, F25, F28, F29, F31, F35, F37, F42, F43, F46, F47, F49, F54, F58, F59, F60, F63, F64, F67, F68, F69, F70, F72, F73, F75, F76, F78, F79, F80, F81, F83, F84, F87, F88, F89, F90, F91, F92, F93, F94, F95, F96, F97, F98, F100, F102, F103, F104, F105, F107, F110, F115, F117, F118, F120, F131, F132
RFC	F2, F29, F59, F74, F77
ETC	F30, F37, F59, F77, F126
ν-SVC	F2, F22, F25, F31, F37, F59, F68, F74, F81, F96, F126
PAC	F2, F126
SGDC	F2, F81
LRCVC	F2, F22, F29, F30, F31, F34, F37, F57, F59, F68, F74, F80, F81, F96, F126

sented in the second column through the eleventh column provide information about which instance is to be considered as the training or the testing instance. Particularly, if a dataset instance d_s from the first column has the value 1 in the corresponding j^{th} column, then the instance d_s of the North Indian (Vadu) cohort will be included in the training dataset of the j^{th} pair of predetermined training and testing dataset, where $j \in \{Fold1, Fold2, \ldots, Fold10\}$ and each $Foldk, k = 1, 2, \ldots 10$ is considered as one of the ten predetermined pairs of training and testing dataset of a particular cohort. Similarly, if d_s has the value 0 in the corresponding j^{th} column, then d_s will be included in the j^{th} testing dataset.

5.2. Data Preprocessing and Feature Selection

In this section, we discuss the preprocessing techniques which are applied to our considered datasets.

Table 7
List of common features selected by considering all the ten folds of training and testing datasets of *north_vadu_train_test*. Purple (F59, F74) are the most important features

Classifier	Selected features common to all the ten pairs of training and testing datasets
RC	F59, F74
MNBC	F1, F3, F4, F6, F7, F9, F10, F11, F14, F15, F16, F17, F19, F20, F22, F23, F24, F25, F26, F28, F29, F30, F32, F33, F34, F35, F36, F37, F41, F42, F43, F44, F45, F46, F47, F48, F50, F51, F53, F54, F55, F59, F68, F69, F71, F72, F73, F75, F76, F78, F79, F80, F81, F82, F83, F84, F87, F88, F89, F90, F91, F92, F93, F100, F101, F102, F104, F105, F106, F110, F113, F114, F115, F127, F131, F132
RFC	F5, F59, F74
ETC	F5, F10, F45, F68, F74, F85, F86, F101
ν-SVC	F5, F25, F28, F33, F34, F37, F44, F59, F68, F74, F82, F85, F86, F101
PAC	F37, F44, F59, F74, F86, F106
SGDC	F5, F25, F37, F44, F59, F74, F82, F101, F106
LRCVC	F37, F59, F74, F86

Table 8
Accuracy, precision and f1-score generated by RC, MNBC, RFC, ETC, ν-SVC, PAC, SGDC and LRCVC for ten pairs of training and testing datasets for *north_train_test*. Highest mean accuracy achieved by using MNBC in terms of accuracy, precision, and f1-score (highlighted in red color)

Classifier	1st T&P[1] set	2nd T&P set	3rd T&P set	4th T&P set	5th T&P set	6th T&P set	7th T&P set	8th T&P set	9th T&P set	10th T&P set	Mean
RC											
Accuracy	0.938	0.875	0.813	0.938	0.875	0.813	0.938	0.813	1.000	0.813	0.881
Precision	0.944	0.905	0.833	0.944	0.878	0.833	0.952	0.889	1.000	0.875	0.905
f1-score	0.939	0.878	0.816	0.933	0.878	0.816	0.937	0.79	1.000	0.78	0.877
MNBC											
Accuracy	**1.000**	**1.000**	0.875	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**0.988**
Precision	**1.000**	**1.000**	0.878	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**0.988**
f1-score	**1.000**	**1.000**	0.878	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**0.988**
RFC											
Accuracy	**1.000**	**1.000**	0.937	0.937	0.937	0.937	0.937	0.937	**1.000**	0.875	0.949
Precision	**1.000**	**1.000**	0.944	0.944	0.944	0.944	0.944	0.944	**1.000**	0.867	0.953
f1-score	**1.000**	**1.000**	0.932	0.932	0.932	0.932	0.932	0.932	**1.000**	0.868	0.946
ETC											
Accuracy	0.938	**1.000**	**0.937**	**1.000**	0.937	0.937	0.937	1.000	**1.000**	0.812	0.949
Precision	0.944	**1.000**	**0.952**	**1.000**	0.944	0.952	0.944	1.000	**1.000**	0.875	0.961
f1-score	0.939	**1.000**	**0.937**	**1.000**	0.932	0.937	0.932	1.000	**1.000**	0.780	0.945
ν-SVC											
Accuracy	**1.000**	**1.000**	0.937	**1.000**	0.937	0.875	0.937	0.937	**1.000**	0.937	0.956
Precision	**1.000**	**1.000**	0.945	**1.000**	0.945	0.905	0.945	0.945	**1.000**	0.945	0.963
f1-score	**1.000**	**1.000**	0.933	**1.000**	0.939	0.877	0.933	0.933	**1.000**	0.933	0.955
PAC											
Accuracy	0.937	0.937	0.875	0.937	0.937	0.937	0.875	0.813	0.937	0.937	0.913
Precision	0.945	0.952	0.905	0.945	0.945	0.945	0.905	0.833	0.952	0.945	0.927
f1-score	0.939	0.937	0.877	0.933	0.933	0.933	0.877	0.817	0.938	0.933	0.912
SGDC											
Accuracy	**1.000**	0.938	0.813	0.938	0.938	**1.000**	0.938	0.938	0.938	0.938	0.937
Precision	**1.000**	0.945	0.875	0.945	0.952	**1.000**	0.945	0.945	0.945	0.945	0.950
f1-score	**1.000**	0.933	0.809	0.933	0.937	**1.000**	0.933	0.933	0.933	0.933	0.934
LRCVC											
Accuracy	0.938	0.938	0.875	0.813	0.875	0.875	0.875	0.875	0.938	0.875	0.887
Precision	0.945	0.945	0.905	0.821	0.879	0.905	0.905	0.886	0.945	0.867	0.899
f1-score	0.939	0.939	0.877	0.803	0.873	0.877	0.877	0.871	0.933	0.868	0.885

T&P: jth testing and prediction set.

Furthermore, in subsection 4.1, we also discuss the recursive feature elimination with cross validation (RFECV) which is used on our datasets to reduce their corresponding dimensions.

The datasets of North Indian and Vadu cohorts considered in this study contain all categorical data. Among these, the Vadu dataset contains some values that need to be added. Here, in Vadu dataset, if

Table 9
Accuracy, precision and f1-score generated by RC, MNBC, RFC, ETC, ν-SVC, PAC, SGDC and LRCVC for ten pairs of training and testing datasets for *vadu_train_test*. Highest mean accuracy achieved by using ν-SVC in terms of accuracy, precision, and f1-score (highlighted in red color)

Classifier	1st T&P set	2nd T&P set	3rd T&P set	4th T&P set	5th T&P set	6th T&P set	7th T&P set	8th T&P set	9th T&P set	10th T&P set	Mean
RC											
Accuracy	0.880	0.880	0.880	0.920	0.840	0.800	0.920	0.960	0.840	0.880	0.880
Precision	0.889	0.912	0.893	0.935	0.831	0.815	0.914	0.952	0.800	0.889	0.883
f1-score	0.879	0.842	0.869	0.899	0.827	0.800	0.914	0.958	0.798	0.870	0.866
MNBC											
Accuracy	0.880	**0.920**	**0.960**	0.960	**1.000**	0.840	**0.960**	**1.000**	**0.960**	0.920	0.940
Precision	0.867	**0.935**	**0.952**	0.952	**1.000**	0.889	**0.972**	**1.000**	**0.944**	0.933	0.944
f1-score	0.856	**0.899**	**0.952**	0.952	**1.000**	0.806	**0.955**	**1.000**	**0.950**	0.896	0.927
RFC											
Accuracy	**1.000**	**0.920**	**0.960**	0.920	0.920	**0.960**	**0.960**	0.880	0.880	0.920	0.932
Precision	**1.000**	**0.935**	**0.952**	0.917	0.917	**0.952**	**0.972**	0.874	0.852	0.907	0.928
f1-score	**1.000**	**0.899**	**0.952**	0.908	0.914	**0.952**	**0.955**	0.871	0.856	0.909	0.922
ETC											
Accuracy	0.960	0.880	0.920	0.960	0.920	0.880	0.960	0.920	0.920	**0.960**	0.928
Precision	0.963	0.882	0.917	0.972	0.949	0.866	0.972	0.933	0.900	0.952	0.931
f1-score	0.950	0.859.	0.908	0.955	0.920	0.859	0.955	0.917	0.900	**0.958**	0.918
ν-SVC											
Accuracy	0.960.	**0.920**	0.920	0.960	**1.000**	0.920	**0.960**	0.960	**0.960**	**0.960**	**0.952**
Precision	0.972	**0.935**	0.917	0.972	**1.000**	0.917	**0.972**	0.952	**0.944**	**0.963**	**0.954**
f1-score	0.955	**0.899**	0.908	0.955	**1.000**	0.908	**0.955**	0.952	**0.950**	0.950	**0.943**
PAC											
Accuracy	0.880	0.880	0.840	0.880	0.840	0.880	0.920	**1.000**	0.920	0.880	0.892
Precision	0.899	0.912	0.817	0.889	0.853	0.866	0.935	**1.000**	0.905	0.878	0.894
f1-score	0.879	0.842	0.817	0.870	0.818	0.859	0.899	**1.000**	0.911	0.874	0.877
SGDC											
Accuracy	0.920	0.800	0.920	0.880	**1.000**	0.880	**0.960**	0.960	0.920	0.880	0.912
Precision	0.914	0.769	0.949	0.889	**1.000**	0.906	**0.972**	0.972	0.939	0.878	0.919
f1-score	0.914	0.763	0.920	0.878	**1.000**	0.837	**0.955**	0.955	0.885	0.874	0.898
LRCVC											
Accuracy	0.920	0.840	0.880	**1.000**	0.960	0.880	0.920	**1.000**	**0.960**	0.880	0.924
Precision	0.914	0.824	0.893	**1.000**	0.963	0.906	0.949	**1.000**	**0.944**	0.867	0.926
f1-score	0.914	0.797	0.869	**1.000**	0.950	0.837	0.920	**1.000**	**0.954**	0.856	0.910

a particular feature f contains a missing value, it is replaced with the modal value of f.

From the perspective of machine learning, the preprocessing of data is very important as far as the overfitting problem of the ML estimators is concerned. Here, it is to be mentioned that one of the causes of the overfitting problem is due to the presence of highly correlated features in a dataset. Accordingly, in our work, we have eliminated the highly correlated features from each pair of the training and the testing datasets. In this perspective, we have calculated the *Pearson* correlation coefficient between a pair of features in the j^{th} training dataset, and essentially eliminated the features having a correlation coefficient more than $|c|$ from the training dataset and its corresponding testing dataset, where c takes the value of either -0.8 or $+0.8$. The highly correlated features which are eliminated from the datasets of North Indian and Vadu cohorts are reported in Table 1.

Moreover, to facilitate the training process of the ML models, the categorical data of our dataset is transformed into numerical data and is rescaled using the in-build scikit-learn's *LabelEncoder* and *MinMaxScaler* classes, respectively.

5.2.1. *Recursive Feature Elimination with Cross Validation (RFECV)*

To reduce the dimension of the feature matrices, in this study, we have considered scikit-learn's in-built class *RFECV*, which is popularly used to select those features from a training dataset that are most pertinent as far as a prediction of the target variable is concerned. Usually, some ML algorithms can be deceived by irrelevant features, resulting in shoddier performance in their predictive capabilities.

Table 10
Accuracy, precision and f1-score generated by RC, MNBC, RFC, ETC, ν-SVC, PAC, SGDC and LRCVC for ten pairs of training and testing datasets for *north_vadu_train_test*. Highest mean accuracy achieved by using ETC in terms of accuracy, precision, and f1-score (highlighted in red color)

Classifier	1st T&P set	2nd T&P set	3rd T&P set	4th T&P set	5th T&P set	6th T&P set	7th T&P set	8th T&P set	9th T&P set	10th T&P set	Mean
RC											
Accuracy	0.880	**0.960**	0.920	0.880	0.760	0.800	0.800	0.800	0.840	0.880	0.852
Precision	0.889	**0.972**	0.911	0.878	0.833	0.783	0.810	0.848	0.852	0.867	0.864
f1-score	0.856	**0.955**	0.917	0.866	0.769	0.780	0.801	0.802	0.830	0.856	0.843
MNBC											
Accuracy	**0.960**	0.920	0.800	0.920	0.880	0.840	0.840	0.880	0.920	0.840	0.880
Precision	**0.972**	0.914	0.836	0.917	0.889	0.833	0.844	0.889	0.905	0.833	0.883
f1-score	0.963	0.914	0.775	0.919	0.878	0.830	0.840	0.881	0.903	0.830	0.873
RFC											
Accuracy	0.920	0.840	**0.960**	**0.960**	**0.960**	0.840	0.800	0.800	0.880	**0.920**	0.888
Precision	0.917	0.827	0.963	**0.972**	**0.952**	0.856	0.769	0.848	0.875	**0.935**	0.891
f1-score	0.914	0.822	0.965	0.955	**0.952**	0.838	0.768	0.792	0.866	**0.899**	0.877
ETC											
Accuracy	**0.960**	0.920	**0.960**	0.920	0.920	**0.880**	**0.920**	0.880	0.880	0.880	**0.912**
Precision	**0.972**	0.949	**0.972**	0.911	0.949	**0.889**	**0.914**	0.889	0.875	0.868	**0.919**
f1-score	**0.955**	0.905	**0.963**	0.917	0.920	**0.869**	**0.914**	0.878	0.865	0.870	**0.906**
ν-SVC											
Accuracy	**0.960**	0.880	0.880	0.800	0.840	0.840	0.880	**0.920**	**0.920**	0.880	0.880
Precision	0.963	0.863	0.889	0.776	0.833	0.833	0.874	**0.907**	**0.905**	0.863	0.871
f1-score	0.950	0.866	0.870	0.763	0.830	0.830	0.871	**0.909**	**0.909**	0.866	0.866
PAC											
Accuracy	0.920	0.880	0.840	0.920	0.760	0.840	0.840	0.840	0.880	0.840	0.856
Precision	0.931	0.867	0.842	0.907	0.726	0.829	0.836	0.833	0.856	0.815	0.844
f1-score	0.914	0.856	0.823	0.909	0.727	0.820	0.824	0.819	0.840	0.793	0.832
SGDC											
Accuracy	0.880	0.920	0.920	**0.960**	0.840	0.840	0.880	**0.920**	0.920	0.880	0.896
Precision	0.869	0.907	0.933	0.952	0.813	0.838	0.866	0.935	0.900	0.906	0.892
f1-score	0.863	0.909	0.896	**0.958**	0.808	0.833	0.859	0.899	0.900	0.837	0.876
LRCVC											
Accuracy	0.840	0.880	0.760	**0.960**	0.840	0.840	0.800	**0.920**	0.880	0.880	0.860
Precision	0.867	0.906	0.733	0.952	0.838	0.817	0.819	0.903	0.852	0.909	0.860
f1-score	0.836	0.837	0.727	**0.958**	0.833	0.817	0.794	0.903	0.856	0.836	0.840

In such contexts, feature selection becomes useful which selects only a subset of features to enhance the effectiveness and efficiency of the ML algorithms. RFECV uses recursive feature selection (RFE) with cross validation loop to find the optimal features.

Using RFECV, we determine the number of optimal features for all the ten combinations of training and testing datasets of the North Indian and Vadu cohorts. For the sake of convenience, hereafter, we categorize ten pairs of predetermined training and testing datasets into three groups and address each of these groups of ten pairs of datasets as follows.

(i) Training and testing dataset of North Indian cohort as *north_train_test*.
(ii) Training and testing dataset of Vadu cohort as *vadu_train_test*.
(iii) Training and testing datasets of North Indian and Vadu cohorts, respectively as *north_vadu_train_test*.

The total number of optimal features obtained by applying RFECV along with each of the eight classifiers on the ten predetermined training and testing datasets of *north_train_test*, *vadu_train_test* and *north_vadu_train_test* are respectively reported in Tables 2, 3 and 4. Furthermore, for each of the ten pairs of training and testing datasets, all the optimal features for *north_train_test*, *vadu_train_test* and *north_vadu_train_test* are presented in the supplementary document *Supplementary II.docx*. Subsequently, based on those selected features, we have also identified the list of optimal features which are common to all the ten pairs of training and testing datasets as far as the feature selection

Table 11
AUROCCS, MCC and HL generated by RC, MNBC, RFC, ETC, ν-SVC, PAC, SGDC and LRCVC for ten pairs of training and testing datasets for *north_train_test*. Highest mean value achieved by using MNBC in terms of AUROCCS, MCC, and HL (highlighted in red color)

Classifier	1st T&P set	2nd T&P set	3rd T&P set	4th T&P set	5th T&P set	6th T&P set	7th T&P set	8th T&P set	9th T&P set	10th T&P set	Mean
RC											
AUROCCS	0.957	0.914	0.859	0.952	0.907	0.859	0.950	0.850	1.000	0.855	0.910
MCC	0.912	0.833	0.722	0.911	0.812	0.722	0.911	0.755	1.000	0.759	0.834
HL	0.063	0.125	0.188	0.063	0.125	0.188	0.063	0.188	0.000	0.188	0.119
MNBC											
AUROCCS	**1.000**	**1.000**	0.907	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**0.991**
MCC	**1.000**	**1.000**	0.812	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**0.981**
HL	**0.000**	**0.000**	0.125	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.013**
RFC											
AUROCCS	1.000	1.000	**0.952**	0.952	0.952	0.952	0.952	0.952	1.000	0.903	0.961
MCC	1.000	1.000	**0.911**	0.911	0.911	0.911	0.911	0.911	1.000	0.812	0.928
HL	0.000	0.000	0.063	0.063	0.063	0.063	0.063	0.063	0.000	0.125	0.050
ETC											
AUROCCS	0.957	1.000	0.950	1.000	0.952	0.950	0.952	1.000	1.000	0.855	0.961
MCC	0.912	1.000	**0.911**	1.000	0.911	0.911	0.911	1.000	1.000	0.759	0.931
HL	0.063	0.000	**0.063**	0.000	0.063	0.063	0.063	0.000	0.000	0.188	0.050
ν-SVC											
AUROCCS	1.000	1.000	**0.952**	1.000	0.957	0.909	0.952	0.952	1.000	0.952	0.967
MCC	1.000	1.000	**0.911**	1.000	0.912	0.827	0.911	0.911	1.000	0.911	0.938
HL	0.000	0.000	**0.063**	0.000	0.063	0.125	0.063	0.063	0.000	0.063	0.044
PAC											
AUROCCS	0.957	0.950	0.909	0.952	0.952	0.952	0.909	0.859	0.950	0.952	0.934
MCC	0.912	0.911	0.827	0.911	0.911	0.911	0.827	0.722	0.911	0.911	0.876
HL	0.063	0.063	0.125	0.063	0.063	0.063	0.125	0.188	0.063	0.063	0.088
SGDC											
AUROCCS	1.000	0.952	0.860	0.952	0.950	1.000	0.952	0.952	0.952	0.952	0.952
MCC	1.000	0.911	0.751	0.911	0.911	1.000	0.911	0.911	0.911	0.911	0.913
HL	0.000	0.063	0.188	0.063	0.063	0.000	0.063	0.063	0.063	0.063	0.063
LRCVC											
AUROCCS	0.957	0.957	0.909	0.860	0.909	0.909	0.909	0.902	0.952	0.903	0.916
MCC	0.912	0.912	0.827	0.732	0.818	0.827	0.827	0.816	0.911	0.812	0.839
HL	0.063	0.063	0.125	0.188	0.125	0.125	0.125	0.125	0.063	0.125	0.113

of RFECV for a particular classifier is concerned. Considering *north_train_test*, *vadu_train_test* and *north_vadu_train_test*, it is observed that maximum and minimum number of optimal features, which are common in all the ten training and testing datasets are selected respectively by MNBC and RC (cf. Tables 5, 6 and 7). Furthermore, from Table 5, we observe that F74 is selected by seven out of eight classifiers, including RC, RFC, ETC, ν-SVC, PAC, SGDC and LRCVC when *north_train_test* is considered. Similarly, from Table 6, the feature F2 of *vadu_train_test* is selected by six classifiers, namely RC, RFC, ν-SVC, PAC, SGDC and LRCVC. Moreover, as far as Table 7 is concerned, F59 and F74 of *north_vadu_train_test* are selected by seven classifiers. Here, F59 is selected by RC, MNBC, RFC, ν-SVC, PAC, SGDC and LRCVC, and F74 is selected by RC, RFC, ETC, ν-SVC, PAC, SGDC and LRCVC.

5.3. Performance analysis of the Machine Learning Classifiers

In this section, we analyze the performance of the eight ML classifiers on the datasets corresponding to the North Indian and Vadu cohorts. For training the ML algorithms efficiently, some important hyperparameters of the classifiers are tuned. Accordingly, we have used the in-built class of scikit-learn library, *RandomizedSearchCV* to determine the optimal values of the hyperparameters of the classifiers for each of the ten predetermined training datasets of the two cohorts. The optimal values of the associated hyperparameter of all the eight classifiers corresponding to each of the ten training datasets are reported in the supplementary document *Supplementary III.docx*.

In this study, the *north_vadu_train_test* implies a cross-dataset setup, where the prediction capabilities

Table 12
AUROCCS, MCC and HL generated by RC, MNBC, RFC, ETC, ν-SVC, PAC, SGDC and LRCVC for ten pairs of training and testing datasets for *vadu_train_test*. Highest mean value achieved by using ν-SVC in terms of AUROCCS, MCC, and HL (highlighted in red color)

Classifier	1st T&P set	2nd T&P set	3rd T&P set	4th T&P set	5th T&P set	6th T&P set	7th T&P set	8th T&P set	9th T&P set	10th T&P set	Mean
RC											
AUROCCS	0.904	0.883	0.898	0.923	0.883	0.859	0.936	0.976	0.862	0.917	0.904
MCC	0.815	0.824	0.82	0.881	0.762	0.712	0.876	0.941	0.752	0.833	0.822
HL	0.120	0.120	0.120	0.080	0.160	0.200	0.080	0.040	0.160	0.120	0.120
MNBC											
AUROCCS	0.901	**0.923**	**0.970**	**0.970**	**1.000**	0.862	**0.960**	**1.000**	0.973	0.925	0.949
MCC	0.821	**0.881**	**0.941**	**0.941**	**1.000**	0.776	**0.940**	**1.000**	0.940	0.884	0.912
HL	0.120	**0.080**	**0.040**	**0.040**	**0.000**	0.160	**0.040**	**0.000**	**0.040**	0.080	0.060
RFC											
AUROCCS	1.000	0.923	**0.970**	0.931	0.946	**0.970**	**0.960**	0.907	0.906	0.938	0.945
MCC	1.000	0.881	**0.941**	0.878	0.885	**0.941**	**0.940**	0.817	0.816	0.879	0.898
HL	0.000	0.080	**0.040**	0.080	0.080	**0.040**	**0.040**	0.120	0.120	0.080	0.068
ETC											
AUROCCS	**0.962**	0.891	0.931	0.960	0.928	0.899	**0.960**	0.937	0.933	**0.976**	0.938
MCC	**0.940**	0.817	0.878	0.940	0.881	0.815	**0.940**	0.883	0.877	0.941	0.891
HL	**0.040**	0.120	0.080	**0.040**	0.080	0.120	**0.040**	0.080	0.080	**0.040**	0.072
ν-SVC											
AUROCCS	0.960	**0.923**	0.931	0.960	**1.000**	0.931	**0.960**	0.970	0.973	0.962	**0.957**
MCC	**0.940**	**0.881**	0.878	0.940	**1.000**	0.878	**0.940**	0.941	0.940	**0.940**	**0.928**
HL	**0.040**	**0.080**	0.080	**0.040**	**0.000**	0.080	**0.040**	0.040	**0.040**	**0.040**	**0.048**
PAC											
AUROCCS	0.904	0.883	0.869	0.917	0.868	0.899	0.923	**1.000**	0.953	0.914	0.913
MCC	0.815	0.824	0.752	0.833	0.764	0.815	0.881	**1.000**	0.887	0.825	0.840
HL	0.120	0.120	0.160	0.120	0.160	0.120	0.080	**0.000**	0.080	0.120	0.108
SGDC											
AUROCCS	0.936	0.830	0.928	0.923	**1.000**	0.885	**0.960**	0.960	0.911	0.914	0.925
MCC	0.876	0.689	0.881	0.834	**1.000**	0.825	**0.940**	0.940	0.878	0.825	0.869
HL	0.080	0.200	0.080	0.120	**0.000**	0.120	**0.040**	0.040	0.080	0.120	0.088
LRCVC											
AUROCCS	0.936	0.853	0.898	1.000	0.962	0.885	0.928	**1.000**	**0.977**	0.901	0.934
MCC	0.876	0.754	0.820	1.000	0.940	0.825	0.881	**1.000**	**0.941**	0.821	0.886
HL	0.080	0.160	0.120	0.000	0.040	0.120	0.080	**0.000**	**0.040**	0.120	0.076

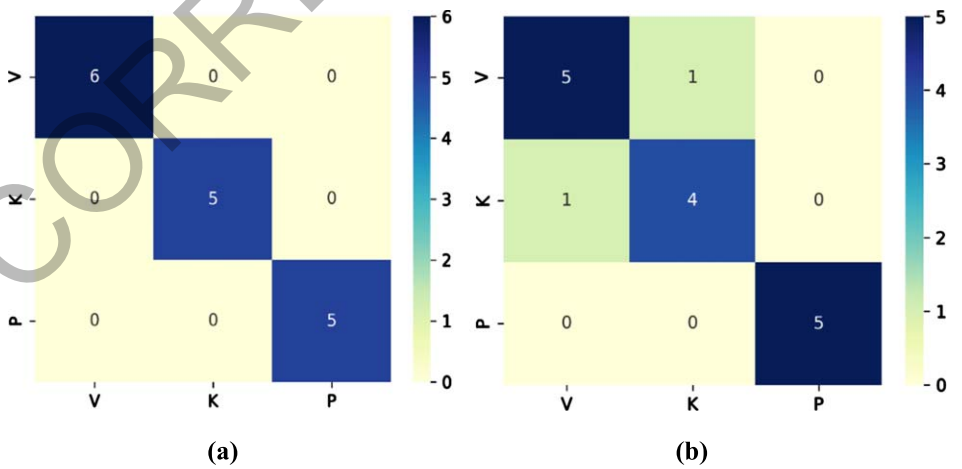

Fig. 3. The confusion matrices for *north_train_test* generated by MNBC for (a) first, second and fourth through tenth testing datasets (b) third testing dataset. From this confusion matrix we can say, (a) all results are true positive, i.e. all 6 were Vata, 5 were Kapha and 5 were Pitta but in (b) the model incorrectly predicted that 1 were Kapha when it were actually Vata (False Positive, FP), and incorrectly predicted that 1 was Vata when it was actually Kapha. (False Negative, FN).

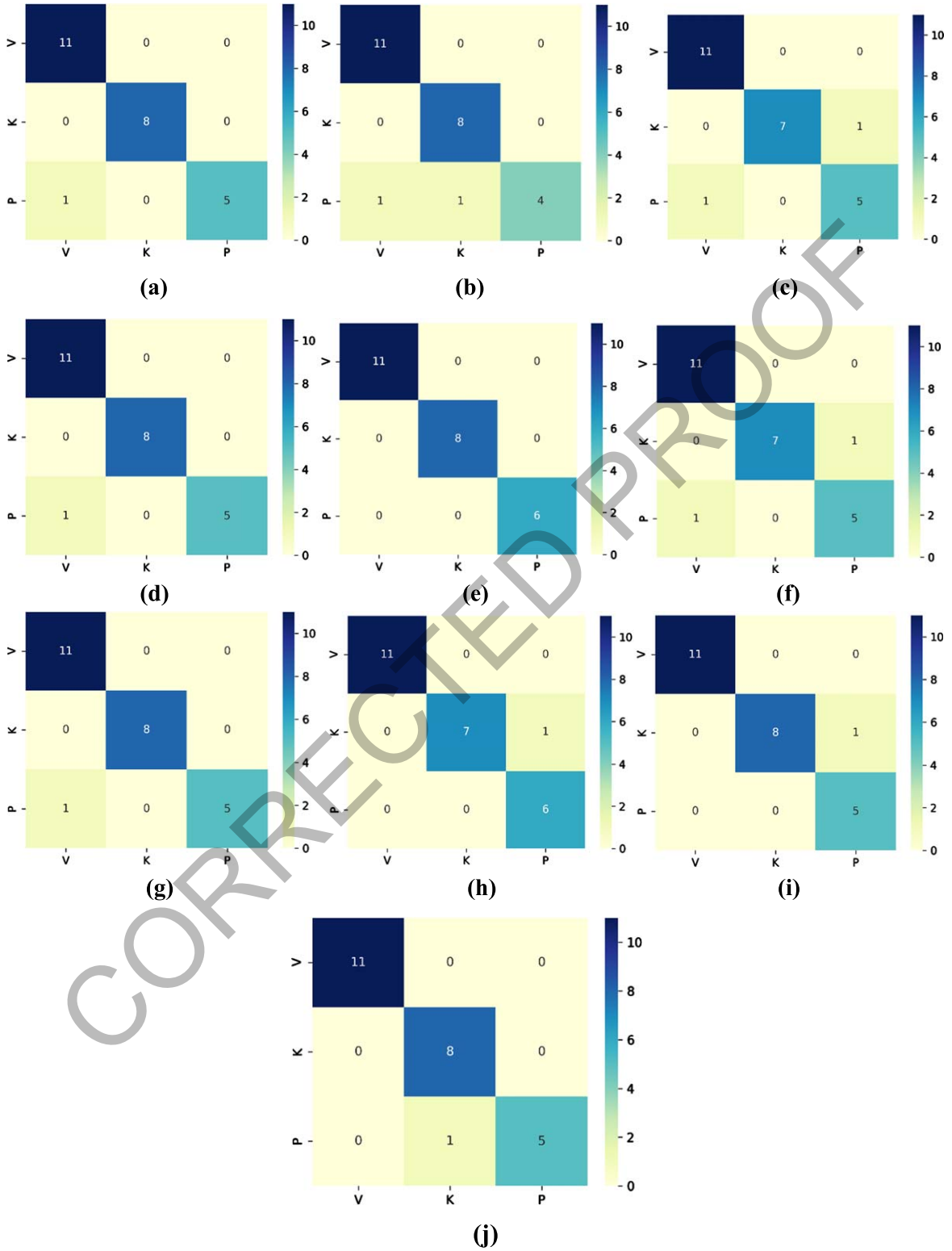

Fig. 4. The confusion matrices for *vadu_train_test* generated by ν-SVC for (a) first testing dataset, (b) second testing dataset, (c) third testing dataset, (d) forth testing dataset, (e) fifth testing dataset, (f) sixth testing dataset, (g) seventh testing dataset, (h) eighth testing dataset, (i) ninth testing dataset and (j) tenth testing dataset. From the confusion matrices we can say the percentage of false negative answers is very less.

Table 13
AUROCCS, MCC and HL generated by RC, MNBC, RFC, ETC, ν-SVC, PAC, SGDC and LRCVC for ten pairs of training and testing datasets for *north_vadu_train_test*. Highest mean value achieved by using ETC in terms of AUROCCS, MCC, and HL (highlighted in red color)

Classifier	1st T&P set	2nd T&P set	3rd T&P set	4th T&P set	5th T&P set	6th T&P set	7th T&P set	8th T&P set	9th T&P set	10th T&P set	Mean
RC											
AUROCCS	0.911	0.960	0.945	0.897	0.851	0.847	0.865	0.869	0.899	0.901	0.895
MCC	0.835	0.940	0.880	0.814	0.703	0.702	0.717	0.738	0.783	0.821	0.793
HL	0.120	**0.040**	0.080	0.120	0.240	0.200	0.200	0.200	0.160	0.120	0.148
MNBC											
AUROCCS	**0.967**	**0.936**	0.855	0.952	0.923	0.885	0.880	0.928	0.946	0.885	0.916
MCC	**0.940**	0.876	0.725	0.888	0.834	0.766	0.756	0.839	0.886	0.766	0.828
HL	**0.040**	0.080	0.200	0.080	0.120	0.160	0.160	0.120	0.080	0.160	0.120
RFC											
sAUROCCS	0.946	0.879	**0.975**	0.960	**0.970**	0.890	0.835	0.863	0.923	**0.923**	0.917
MCC	0.885	0.762	**0.941**	**0.940**	**0.941**	0.771	0.691	0.736	0.832	**0.881**	0.838
HL	0.080	0.160	**0.040**	**0.040**	**0.040**	0.160	0.200	0.200	0.120	**0.080**	0.112
ETC											
AUROCCS	0.960	0.921	0.967	0.945	0.928	**0.917**	**0.936**	0.923	0.923	0.915	**0.933**
MCC	**0.940**	**0.883**	0.940	0.880	0.881	**0.833**	0.876	0.834	0.832	0.824	**0.872**
HL	**0.040**	0.080.	**0.040**	0.080	0.080	**0.120**	**0.080**	0.120	0.120	0.120	**0.088**
ν-SVC											
AUROCCS	0.962	0.909	0.917	0.839	0.885	0.885	0.907	**0.938**	**0.950**	0.909	0.910
MCC	**0.940**	0.819	0.833	0.706	0.766	0.766	0.817	**0.879**	**0.884**	0.819	0.823
HL	**0.040**	0.120	0.120	0.200	0.160	0.160	0.120	**0.080**	**0.080**	0.120	0.120
PAC											
AUROCCS	0.930	0.901	0.876	0.938	0.808	0.868	0.866	0.870	0.889	0.856	0.880
MCC	0.877	0.821	0.768	0.879	0.633	0.753	0.751	0.759	0.817	0.754	0.781
HL	0.080	0.120	0.160	0.080	0.240	0.160	0.160	0.160	0.120	0.160	0.144
SGDC											
AUROCCS	0.900	0.938	0.925	**0.976**	0.864	0.891	0.899	0.923	0.933	0.885	0.913
MCC	0.815	0.879	0.884	0.941	0.753	0.773	0.815	0.881	0.877	0.825	0.844
HL	0.120	0.080	0.080	**0.040**	0.160	0.160	0.120	0.080	0.080	0.120	0.104
LRCVC											
AUROCCS	0.893	0.885	0.807	**0.976**	0.891	0.869	0.853	0.933	0.906	0.887	0.890
MCC	0.783	0.825	0.634	0.941	0.773	0.752	0.709	0.876	0.816	0.831	0.794
HL	0.160	0.120	0.240	0.040	0.160	0.160	0.200	0.080	0.120	0.120	0.140

of all the classifiers are studied by training them on the dataset of North Indian cohort and testing them on the dataset corresponding to Vadu cohort. For each of the three groups of datasets, we analyze the performance of the ML classifiers by studying six performance metrics which are reported in Table 8 through Table 13. Among these tables, the accuracy, precision and f1-score are presented in Tables 8, 9 and 10, respectively for *north_train_test*, *vadu_train_test* and *north_vadu_train_test*. Whereas, AUROCCS, MCC and HL are reported in Tables 11, 12 and 13 when the classifiers are executed on *north_train_test*, *vadu_train_test* and *north_vadu_train_test*, respectively. In each of these tables, the best values are highlighted in bold. Considering the accuracy, precision, and f1-score, MNBC, ν-SVC and ETC become superior in Tables 8, 9 and 10, respectively. Among all the ten pairs of training and testing datasets, in Table 8, MNBC generates better results for nine training and testing combinations of *north_train_test* with respect to the accuracy, precision and f1-score, and therefore the mean of each of these performance metrics also becomes better. In Table 9, we observe that RFC generates better accuracy, precision, and f1-score in five out of the ten pairs of training and testing datasets. However, ν-SVC generates better mean for each of the accuracy precision and f1-score. As a matter of fact, considering all the ten combinations of training and testing datasets of *vadu_train_test*, ν-SVC eventually emerges as superior compared to all its competitors with respect to the mean values of accuracy, precision, and f1-score. In Table 10, ETC generates better accuracy, precision, and f1-score in four out of ten pairs of training and testing datasets, which is the maximum number of training and testing pairs for any classifier by

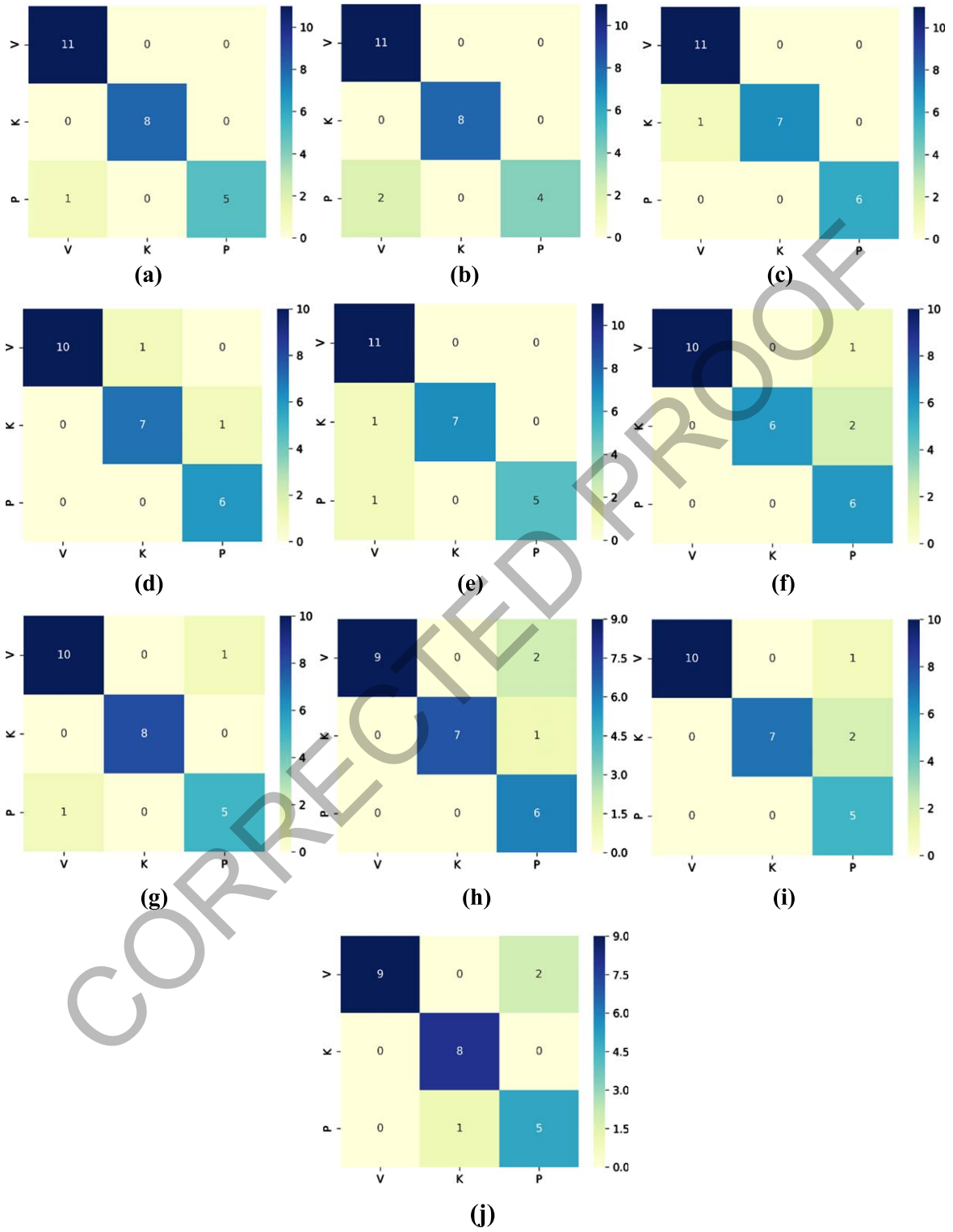

Fig. 5. The confusion matrices for *north_vadu_train_test* generated by ETC for (a) first testing dataset, (b) second testing dataset, (c) third testing dataset, (d) forth testing dataset, (e) fifth testing dataset, (f) sixth testing dataset, (g) seventh testing dataset, (h) eighth testing dataset, (i) ninth testing dataset and (j) tenth testing dataset. In this method the percentage of false negative answers is more than the above method.

considering *north_vadu_train_test*. Like Table 8, in Table 11, MNBC generates better AUROCCS, MCC and HL in nine out of ten training and testing pairs of datasets of *north_train_test*. Hence, the mean values of AUROCCS, MCC and HL also become superior for MNBC. In Table 12, for *vadu_train_test*, MNBC emerges as superior in six out of ten training and testing datasets for AUROCCS and MCC, and seven out of ten training and testing datasets for HL. However, so far, the mean values of AUROCCS, MCC and HL are concerned ν-SVC emerges as the superior classifier. Subsequently, in Table 13, for *north_vadu_train_test*, MCC and HL become better for RFC in four out of ten training and testing pairs of datasets, whereas, for AUROCCS, three out of ten training and testing pairs of datasets becomes better in RFC. However, considering the mean values of all the AUROCCS, MCC and HL, ETC outperforms all its competitors.

Furthermore, analysing the performances of the eight classifiers based on all the six performance metrics as reported in Table 8 through Table 13, it is observed that each of the performance metrics becomes better for MNBC, ν-SVC and ETC with respect to their mean values for *north_train_test*, *vadu_train_test* and *north_vadu_train_test*, respectively. Consequently, we provide the RFECV visualization plots corresponding to each of the ten training datasets for MNBC, ν-SVC and ETC, respectively in Fig. SII-1, Fig. SII-2 and Fig-II.3 of the *Supplementary II.docx*. These visualization plots are used to graphically represent the smallest number of features for which a classifier generates a highest percentage of correct classification of instances. Moreover, we present the confusion matrices of each testing dataset corresponding to the *Foldi*, $i = 1, 2, \ldots, 10$ for each of these three classifiers for *north_train_test*, *vadu_train_test* and *north_vadu_train_test*, respectively. These confusion matrices are depicted in Fig. 3 through Fig. 5.

6. Discussion of the results

Here we discuss the conclusive outcomes of the analysis reported in Section 5. From Table 1, it is observed that F5, F27, F101 are least important to AI and that can be removed from the dataset of Vadu cohort. From Tables 2 and 3, it is observed that the standard deviation of the number of optimal features is very high. We conclude that this is a drawback of the dataset in terms of the number of records. From Tables 5, 6 and 7, we found the most important feature(s) for the datasets are F2, F59, F74.

Metric is an important parameter to judge the capability of an AI. Here we have used 6 metrics namely accuracy, precision, f1-score, AUROCCS, MCC, and HL to evaluate the proposed method. From Tables 8, 9, 10, 11, 12, and 13, we observed that MNBC given the best mean value for North dataset, ν-SVC for Vadu dataset and ETC for the North-Vadu dataset.

Figure 3 shows the evidence of the performance of MNBC for the North dataset. Similarly Figs. 4 and 5 shows superiority of ν-SVC and ETC for Vadu and North-Vadu dataset respectively. This study argues for the uses of MNBC, ν-SVC, and ETC combined for better result.

7. Conclusion

This study conducts a comparative analysis of eight ML classifiers on clinical methods of Prakriti classification of Ayurveda system of medicine. Here, we consider two genetically homogeneous northern and western India cohorts for Prakriti predictions. Further, these classifiers are also applied on cross-dataset setup, where each of these classifiers is trained on datasets of the North Indian cohort and used for prediction in the West Indian or Vadu cohort. The comparative analysis of the algorithms suggests that out of eight classifiers MNBC, ν-SVC and ETC become superior in *north_train_test*, *vadu_train_test* and *north_vadu_train_test*, respectively for all the performance metrics. It is also observed from this study that a reduced feature set can efficiently train a classifier while predicting *Prakriti* of an individual. This, in fact, can be useful in the decision-making process of a trained Ayurveda physician.

In the future, the study can be extended to predict non-extreme Prakriti types. Furthermore, these ML classifiers will also help in heterogeneous populations and eventually help decode a novel link of genotypes to multisystem phenotypes in associated studies.

Compliance with ethical standards

Ethical approval

This article does not contain any studies with human participants or animals performed by any of the authors.

Funding details

The authors declare that no funding will be received for publishing this paper.

Conflict of interest

Both authors of this research paper declare that there is no conflict of interest.

Informed consent was obtained from all individual participants included in the study.

Authorship contributions

The first author (**Saibal Majumder**): Conceptualization, Formal analysis, Visualization, Methodology, Resources, Writing - original draft, review & editing.

The second author (**Rintu Kutum**): Conceptualization, Data generation, Validation, Writing - original draft, comparative study.

The third and fourth authors (**Debnarayan Khatua & Arif Ahmed Sekh**): Methodology, Writing – review & editing.

The fifth author (**Samarjit Kar**): Formal analysis, Methodology, Validation, Writing – review & editing.

The sixth author (**Mitali Mukerji**): Resources, Data generation, Supervision, proofread, drafting.

The seventh author (**Bhavana Prasher**): Resources, Data generation, Supervision, proofread, Writing – review & editing.

Supplementary material

The supplementary material is available in the electronic version of this article: https://dx.doi.org/10.3233/JIFS-220990.

References

[1] M.M. Pandey, S. Rastogi and A.K.S. Rawat, Indian Traditional Ayurvedic System of Medicine and Nutritional Supplementation, in *Evidence-Based Complement Altern Med,* Editor T. Kummalue, 2013, 1–12.

[2] H. Sharma and R. Keith Wallace, Ayurveda and Epigenetics, *Med* (2020). doi:10.3390/medicina56120687

[3] V. Vianu, Invited Article Foreword, *J ACM* **62** (2015), 297–303.

[4] B. Prasher, G. Gibson and M. Mukerji, Genomic insights into ayurvedic and western approaches to personalized medicine, *Journal of Genetics* **95** (2016), 209–228.

[5] M. Mukerji and B. Prasher, Genomics and traditional Indian ayurvedic medicine, In: Kumar D, Chadwick R, eds. *Genomics and Society: Ethical, Legal, Cultural and Socioeconomic Implications*, Cambridge, MA: Academic Press; 271–292, (2016).

[6] N. Lemonnier, G.B. Zhou, B. Prasher, et al., Traditional knowledge-based medicine: a review of history, principles and relevance in the present context of P4 systems medicine, *Prog Prev Med* **7** (2017), e0011. doi: 10.1097/pp9.0000000000000011

[7] K. Bhadresha, M. Patel, J. Brahmbhatt, et al., Ayurgenomics: a Brief Note on Ayurveda and Their Cross Kingdom Genomics, *Indian Aced Sci* **5** (2020), 168–177.

[8] S. Shilpa and C.G. Venkatesha Murthy, Understanding Personality from Ayurvedic Perspective for Psychological Assessment: A Case, *Ayu* **32** (2011), 12–19.

[9] V.N. Sumantran and G. Tillu, Insights on Personalized Medicine from Ayurveda, *J Altern Complement Med* **19** (2013), 370–375.

[10] B. Prasher, B. Varma, A. Kumar, B.K. Khuntia, R. Pandey, A. Narang, et al., Ayurgenomics for Stratified Medicine: TRISUTRA Consortium Initiative across Ethnically and Geographically Diverse Indian Populations, *J Ethnopharmacology* **197** (2017), 274–293.

[11] T.P. Sethi, B. Prasher and M. Mukerji, Ayurgenomics: a new way of threading molecular variability for stratified medicine, *ACS Chem Biol* **6** (2011), 875±880. https://doi.org/10.1021/cb2003016 PMID: 21923095.

[12] H. Rotti, K.P. Guruprasad, J. Nayak, S.P. Kabekkodu, H. Kukreja, S. Mallya, et al., Immunophenotyping of normal Individuals Classified on the Basis of Human Dosha Prakriti, *J Ayurveda Integr Med* **5** (2014), 43–49.

[13] Prasher B, Negi S, Aggarwal S, A.K. Mandal, T.P. Sethi, et al., Whole genome expression and biochemical correlates of extreme constitutional types defined in Ayurveda, *Journal of Translational Medicine* **6**(48) (2008). doi: 10.1186/1479-5876-6-48

[14] H. Sharma, Ayurveda: Science of Life, Genetics and Epigenetics, *Ayu* **37** (2016), 87–91.

[15] M. Haider, D. Dholakia, A. Panwar, P. Garg, V. Anand, A. Gheware, et al., Traditional Use of Cissampelos Pareira L. For Hormone Disorder and Fever Provides Molecular Links of ESR1 Modulation to Viral Inhibition, *Sci Rep* **11**(1) (2021), 20095.

[16] A. Jelenkovic, R. Sund, Y.M. Hur, Y. Yokoyama, J.V. Hjelmborg, S. Möller, et al., Genetic and Environmental Influences on Height from Infancy to Early Adulthood: An Individual-Based Pooled Analysis of 45 Twin Cohorts, *Sci Rep* **6** (2016), 28496.

[17] Y. Cheng, C. He, M. Wang, X. Ma, F. Mo, S. Yang, et al., Targeting Epigenetic Regulators for Cancer Therapy: Mechanisms and Advances in Clinical Trials, *Signal Transduct Target Ther* **4** (2019), 62.

[18] B. Prasher, C. Sachidanandan, M. Mukerji, et al., Ayurgenomics: Bringing Age-Old Wisdom to the Healthcare of the Future [Internet]. CSIR. 2022 [cited 2022 Jan 20]. (2022). Available from: https://www.csir.res.in/ayurgenomics-bringingage-old-wisdom-healthcare-future

[19] P. Sharma, Charaka Samhita (4 volumes with English translation), 7th Edition, Chaukamba Orientalia, Varanasi, (2003).

[20] S. Datar, Murthy Development and standardization of the Mysore Tridosha Scale, *AYU—An International Quarterly Journal of Research in Ayurveda* **32** (2011), 308–314. doi: 10.4103/0974-8520.93905

[21] J. Anderson, J. Parikh and D. Shenfeld, Reverse Engineering and Evaluation of Prediction Models for Progression to Type 2 Diabetes: Application of Machine Learning Using Electronic Health Records, *Journal of Diabetes* (2016).

[22] A. Esteva, B. Kuprel and R.A. Novoa, Dermatologist-level classification of skin cancer with deep neural networks, *Nature* **542**(7639) (2017), 115–118. https://doi.org/10.1038/nature21056

[23] P. Tiwari, R. Kutum, T. Sethi, et al., Recapitulation of Ayurveda constitution types by machine learning of phenotypic traits, *PLoS ONE* **12**(10) (2017), e0185380. https://doi.org/10.1371/journal.pone.0185380

[24] D. Khatua, A.A. Sekh, R. Kutum, M. Mukherji, B. Prasher and S. Kar, Classification of Ayurveda Constitution Types: A Deep Learning Approach, *Soft Computing* (2023).

[25] A.E. Hoerl and R.W. Kennard, Ridge regression: Biased Estimation for Nonorthogonal Problems, *Technometrics* **12**(1) (1970), 55–67.

[26] A.N. Tikhonov, Translated in "solution of incorrectly formulated problems and the regularization method, *Soviet Mathematics* **4** (1963), 1035–1038.

[27] A. McCallum and K. Nigam, A comparison of event models for Naive Bayes text classification, In *Proceedings: AAAI/ICML-98 Workshop on Learning for Text Categorization*, pp. 41–48. (1998).

[28] C.D. Manning, P. Raghavan and H. Schuetze, Introduction to Information Retrieval, Cambridge University Press, pp. 234–265. (2008).

[29] H. Zhang, The optimality of Naive Bayes, In *Proceedings: Seventeenth International Florida Artificial Intelligence Research Society Conference*, FLAIRS, 2004.

[30] L. Breiman, Random Forests, *Machine Learning* **45**(1) (2001), 5–32.

[31] P. Geurts, D. Ernst and L. Wehenkel, Extremely Randomized Trees, *Machine Learning* **63** (2006), 3–42.

[32] B. Schölkopf, A.J. Smola, R.C. Williamson and P.L. Bartlett, New support vector algorithms, *Neural Computing* **12**(5) (2000), 1207–1245.

[33] K. Crammer, O. Dekel, J. Keshet, S. Shaley-Shwartz and Y. Singer, Online Passive-Aggressive Algorithms, *Journal of Machine Learning Research* **7** (2006), 551–585.

[34] Y. Tsuruoka, J. Tsujii and S. Ananiadou, Stochastic Gradient Descent Training for L1-regularized Log-linear models with Cumulative Penalty, In *Proceedings: Forty Seventh Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP*, pp. 477–485. (2009).

[35] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, (2006).