Gradient Boost algorithms for Modelling Malayalam Poem Syllable Duration

JASIR M.P[1,2]
KANNAN BALAKRISHNAN[1]
JASEENA K.U[2]

[1]CUSAT - Cochin University of Science and Technology
DCA - Department of Computer Applications
Kerala-India-PIN CODE-682022
[2]MES College Marampally, Aluva
DCA-Department of Computer Applications
Kerala-India-PIN CODE-683105

Abstract. Emulating natural speech has been a top priority ever since the research activities began in the area of Natural Language Processing (NLP). Text To Speech Synthesis (TTS) consists of several stages, which include Text Normalization, Syllabification and Unit Selection, Duration Analysis Modelling, and Prosody Analysis Modelling. Proper syllabification was required earlier when rule-based concatenative synthesis was used as the main method to synthesize speech. Now statistical parametric speech synthesis is the state of the art. Supervised and unsupervised machine learning frameworks can be used to model different aspects of speech such as duration, prosody etc. The proposed work uses classical poem construct Vruta (meter) to identify the features determine syllable duration. Nineteen features are extracted from the orthographic representation of poem according to the Vruta definition. Kakali, Keka, and Manjari are the Vrutas considered. Also the contextual features of the syllables and the accoustic properties like the origin of the syllable are considered to build the feature set. The proposed work employs Gradient Boost Algorithms for modelling the duration of Malayalam poem syllables. All the models give superior values for the coefficient of determination (R2) compared to other major models. Simple Gradient Boost Machine (GBM) is able to produce 90.723 for R2. Similarly, XGBoost gives 90.726, LightBoost yields 90.693 and CatBoost delivers 90.819. Also, the models exhibit lesser values for different Statistical Error Indicators (SEI) - MAE, RMSE, and MAPE.

Keywords: Text To Speech Synthesis, Malayalam TTS, Duration Modelling, Ensemble Machine learning, Gradient Boost Machine, XGBoost, LightBoost, CatBoost

(Received April 07th, 2022 / Accepted May 2st, 2022)

1 Introduction

TTS is a branch of NLP that tries to emulate natural speech by artificially synthesizing speech from textual representations. It has got widespread applications as an assistive experience enhancing technology [14].

The researches in the area of TTSs have begun as early as 1950. At first the efforts were to approximate the articulatory aerodynamics of the vocal tract. The first electronic model that emulates vocal tract was developed in M.I.T in 1953. It required hand adjustments of a variable inductor for each section [53]. Dynamic control was added to this M.I.T model by Rosen.G in 1958 [47]. Due to the complexity of the model and the limitations of the available technology during the period, articulatory synthesis eventually frizzled out. Later rule based concatenative speech synthesis came in to the fore. context based rules were derived to concatenate speech units stored in the corpus to generate synthetic speech [14]. Finally, with the availability of new generation computing devices capable of performing complex computations, statistical parametric speech synthesis started to bloom from the 1990s [54].

To build an intelligible and natural TTS, a thorough knowledge of the domain is required [55, 58, 29, 46, 45, 44]. The contextual, accoustic and phonological features of the language must be thoroughly analysed before modelling the framework. As far as Malayalam poems are concerned, many of them are written in Vrutas. Vruta is the narrative meter of a poem. It determines many factors that decide the rhythmic utterance of syllables. It also defines the total instances of Laghu and Guru (Malayalam grammatical constructs that determines the duration of a syllable) syllables in a verse. Also it restricts the total number of syllables that can appear in a verse [37]. Three Vrutas namely Kakali, Manjari and Keka are analysed to identify extractible features from the text representation of poems written in these Vrutas. Nineteen such features are extracted for each syllable to derive cues about the duration of that syllable. The proposed work uses gradient boost algorithms GBM, XGBoost, LightGBM, CatBoost to model the duration of the syllables.

Section 2 gives a brief account of the research works in the field of Text To Speech Synthesis. Section 3 briefly explain the methodology, characteristics of the dataset, Malayalam Vruta constructs and performance evaluation metrics. Section 4 compares the performance of the proposed models against major duration models. Section 5 concludes the paper by citing future directions and scope of betterment.

2 Related Work

Various methods have been employed in building TTS systems over the period of time. As and when the research works in articulatory synthesis reached a state of impasse, rule based speech synthesis took over. Different stages in TTS like Text Normalization, Syllabification and Unit Selection, Duration analysis and Modelling, Prosody analysis and modelling were modelled by deriving context based rules in this approach. One of the earliest rule based speech synthesis system was developed by Dennis H. Klatt [13]. He used derived contextual rules from English language to develop an unrestricted TTS for the language. A detailed review of rule based TTS systems can be found at [14, 24].

Rule based speech synthesis has its own limitations in emulating natural speech. The clicks at the joins of syllable boundaries causes hindrance to the smooth transition of waveforms. Different concatenative technologies are employed to join speech units from the corpus viz Pitch Synchronous Overlap and Add (PSOLA), Time Domain PSOLA (TD-PSOLA), Multi-Band Resynthesis Overlap Add (MBROLA) etc. A detailed review on the methods can be found at [12]. Concatenative speech synthesis also had to deal with the problem of the choice of optimum speech unit to be stored in the memory. If the individual units are too small, it would compromise the intelligibility of synthesized speech. On the other hand bigger units like words or sentences would seriously affect the flexibility of the system.

It is to get rid of these limitations, statistical parametric speech synthesis introduced into the mix. Here statistical classifiers like Artificial Neural Networks (ANN), Support Vector Machines (SVM), Decision Tree (DT), Random Forest (RF) etc. are used to model different stages in TTS [20]. Later an intermediate representation such as mel spectrogram can be generated interpreting these features. These representations are then be fed to speech synthesizing vocoder networks such as Wavenet, WaveGlow, and ClariNet [30, 35, 33].

Many statistical parametric systems have been deployed to model different stages in TTS in English and Indian languages. N-gram models and Weighted Finite State Transducers (WFST) are used successfully to normalize English text by Richard Sproat et al. [52]. Fei Liu et al. have modelled English Text Normalization system employing K-means clustering algorithm [19]. Lan Huang et al. make use of a deep learning architecture to normalize English text [8].

There are a handful of works in Indian languages as well. Bayesian word sense induction is used by Anindya Sau et al. to normalize Bengali text [48]. Piyush Makhija et al. employ bidirectional Long Term Short Memory (LSTM) to normalize Hindi text [21].

Another major step in TTS is duration modelling of syllables. Olga Goubanova and Paul Taylor use Bayesian belief networks to model syllable durations in their English TTS [54]. The CHiVE model by Vincent Wan et al. learn a mapping between pairs of $\{X, Y\}$, where X is a sequence of input features, and Y is a sequence of prosodic parameters [11]. Zack Hodari et al. use the syntactic and semantic information extracted from the text to learn context-dependent prosodic knowledge to apply in a context-aware model of prosody [7]. Bidirectional LSTM is used to encode the syntactic features of the context while processing the text. Jonathan Shen et al. employs Fine Grained Variational Auto Encoder (FAVAE) architecture to model syllable duration in English [49].

Some of the notable works in Indian languages in duration modelling are explained below. N Sridhar Krishna et al. make use of Classification And Regression Tree (CART) to model the segmental duration of Hindi syllables [17]. An extension of the same method is later used on an enhanced dataset [16]. Deepa P Gopinath et al have carried out Duration analysis on the syllables in Malayalam [4]. Based on their findings they have developed a Probabilistic Distribution Model to predict the vowel duration [5]. Later they have also proposed a hybrid duration model combining CART and Hidden Markov Model (HMM) [6]. SVMs and ANNs are two commonly used machine learning models employed to predict syllable durations in Indian language TTS. K Srinivasa Rao and B Yegna narayana propose SVM for predicting the duration of syllables [39]. They have also developed an ANN framework to model the duration of syllables [40]. Later a two-stage hybrid model, where the first stage consisted of an SVM syllable classifier, to group syllables based on their duration, was proposed by them. Krothapalli S.Rao and Shashidhar G. Koolagudi analyze the factors that affect the duration of syllables in detail. They used the intuitions from their analysis to develop a Feed Forward Neural Network (FFNN) to model the duration [38]. In Malayalam also there have been efforts to model syllable duration, like the CART based systems developed by Jestin James et al. and Bindhu K. Rajan et. al [9, 36]. Shreekanth T et al. discuss a four-layer Feed Forward Neural Network based duration modelling system for an Hindi TTS [51].

K. Srinivasa Rao and B. Yegnanarayana enhanced their duration modelling work by proposing an ANN based intonation model to emulate prosody of speech in Hindi [41]. Similarly V Ramu Reddy et al. have made a FFNN framework to model intonation in Bengali TTS [42]. V. RamuReddy and K. Sreenivasa Rao propose a two-stage FFNN based approach to model the fundamental frequency (F0) of syllables [43]. Kumuth Thripathi et al. have developed an SVM and Convolutional Neural Network (CNN) combined model for Hindi TTS. [57]. There are some restricted analyses and modelling on limited datasets in Malayalam that takes specific aspects of prosody like pause or chaotic nature of utterance [34, 56]. Asoke Kumar Datta in his work studies the intonation patterns present in standard and colloquial Bengali, to model a natural-sounding TTS [2]. K. Pal and B.V Patel discuss a machine learning model for the classification of poems based on RAS (Emotion) [32]. They have later extended the model as an Automatic Multi-class Document Classification system of Hindi Poems based on the underlying emotion conveyed in the poem [31].

Until recently duration and prosody of the speech was modelled separately. Presently deep learning frameworks are the state of the art in TTS. Convolutional Neural Networks (CNN) are used to identify and extract patterns and dependencies in the input data. Advanced deep learning architectures like Tacotron, WaveNet and WaveGlow have enabled scholars to track down the problem of TTS by providing a single end to end model. Frameworks like Tacotron captures such time aligned features as an intermediate representation of mel spectrograms from a combination of text and speech data. These representations can be fed into generative deep learning vocoders like WaveNet or WaveGlow to produce synthetic audio samples [35].

As a low resource language, Malayalam faces a number of difficulties in modelling a natural sounding TTS. Majority of the works in the language have been performed on limited dataset [24]. Supervised machine learning models can only be developed after accumulating enough knowledge about the linguistic peculiarities of the target study. For example scholars like Noam Chomsky have performed thorough analysis to decode the syntactical structure of English language [1]. Computational linguists could then transcend these observations to the engineering domain to model different problems that come under the ambit of NLP. The statistical speech synthesizing methods for Malayalam can be implemented in two ways, supervised or unsupervised. In order to implement supervised statistical speech synthesizers a thorough knowledge about the domain is inevitable. The linguistic peculiarities of the language will play a significant role in determining the features. The proposed work uses the notion of Vruta constructs in Malayalam poems to model the duration for Malayalam poem syllables. Keka, Kakali and Manjari are the three Vrutas analysed in this study. More details about the Vruta definitions can be found at [37]. The identification and extraction of these features are explained in [25].

3 Proposed Methodology

3.1 Speech Dataset

. The speech corpus consists of 31568 syllables collected from poems that belong to Kakali, Manjari, and Keka Vrutas. 10656 syllables are collected from 888 lines of Kakali poems, 13244 syllables are collected from 946 lines of Keka poems and 7668 syllables are collected from 697 lines of Manjari poems. The syllables are tagged in the speech corpus according to the Vruta, position, and syllable.

The poems are recorded by a 20 year old young female with a melodious voice. The poem verses are sampled at frequency 44100 Hz using the speech processing freeware PRAAT. To ensure accurate syllable boundaries, they are manually annotated in the poem verses. The annotated syllables are segmented using PRAAT script to store in the speech corpus. The poems are collected from works of popular Malayalam writers. The Keka verses are collected from Thunchath Ramanjujan Ezhuthachan's Adhyathma Ramayanam, Balakandam Chapter [3]. The Kakali poem verses consist of Kishkindhakadam and Ayodhyakandam chapters of the same book. The Manjari poems are an anthology of works. It consists of poems like Krishnagadha written by Cherussery Namboothiri, Mambazham written by Vyloppilli Sreedhara Menon. [27, 23].

The orthographic representation of the input features are processed according to the Vruta definitions to extract the features. The dataset has been reviewed and published in Mendeley Data with DOI 10.17632/wh6fwmgccf.1. [26]. The dataset and the associated files are available at https://data.mendeley.com/datasets/wh6fwmgccf/1 under a Creative Commons Attribution 4.0 International license.

3.2 Malayalam Vrutas and Lakshana

The Malayalam word Varnam roughly translates to syllable. The Varnams are grouped into sets of two or three to form Ganam (Join) in Malayalam Bhasha Vrutas. The properties of each Ganams in a Padam (Couplet) are analysed using the Vruta Lakshana (identification protocol) to determine the Vruta the couplet belongs to [37]. A syllable in poem can be either Laghu or Guru. Laghus are short duration-ed syllables where the attached vowel sound is not prolonged in utterance. In the language terminology, they are called one Matra syllables. On the other hand Gurus are syllables attached with prolonged vowel sounds. They are denoted as two Matra syllables. Table 1 lists different types of Ganams possible in Malayalam poems. It also shows the Laghu and Guru combinations in these Ganams. The horizontal bar denotes Guru and the 'U' shaped symbol denotes Laghu.The three Bhasha Vrutas considered for the durational analysis of syllables in this work are Kakali, Manjari, and Keka.

According to the Vruta definitions provided by A.R Rajarajavarma, Kakali shall have four sets of three syllabled-five Matra Ganams in each of the verses in consideration. It also puts a restriction that the first Ganam must not begin with the six Matra Ganam-Ya. [37]. Figure 1 shows the syllable split up and Ganam grouping of a poem couplet written in Kakali. Manjari share all the Lakshanas of Kakali, except that it can only have two syllables less in the final Ganam of the second verse.

When it comes to Keka, the Ganam format is fixed to $[3, 2, 2, 3, 2, 2]$ with fourteen syllable in each of the verses. The Vruta Lakshana puts the restriction that each of the Ganams must contain one Guru syllable minimum. [37]. Figure 2 shows the Ganam grouping of a Keka couplet. It also shows some of the features like syllable position in a verse, Syllable position in a Ganam, total matra in a Ganam, and number of syllables in a Ganam.Table 2 summarizes the features defining Kakali, Manjari and Keka Vrutas.

Table 1: Ganams in Varna Vrutas

Symbol	Name	Remark	Total Matra
– ,– , –	Ma Ganam	All Guru	6
U, –, –	Ya Ganam	First Laghu	5
–, U, –	Ra Ganam	Mid Laghu	5
U, U, –	Sa Ganam	End Guru	4
–, –, U	Tha Ganam	End Laghu	5
U, –, U	Ja Ganam	Mid Guru	4
–, U, U	Bha Ganam	First Guru	4
U, U, U	Na Ganam	All Laghu	3

Table 2: Defining characteristics of Kakali, Manjari and Keka Vrutas. Abbreviations: SIV-Syllables In Verse, GPIV-Ganam Pattern In Verse, MIG-Matra in Ganam, MIV-Matra In Verse

Vruta	SIV1	SIV2	GPIV1	GPIV2	MIG	MIV1	MIV2
Kakali	12	12	[3,3,3,3]	[3,3,3,3]	5	20	20
Manjari	12	10	[3,3,3,3]	[3,3,3,1]	5	20	16
Keka	14	14	[3,2,2-3,2,2]	[3,2,2-3,2,2]	2/3	22-28	22-28

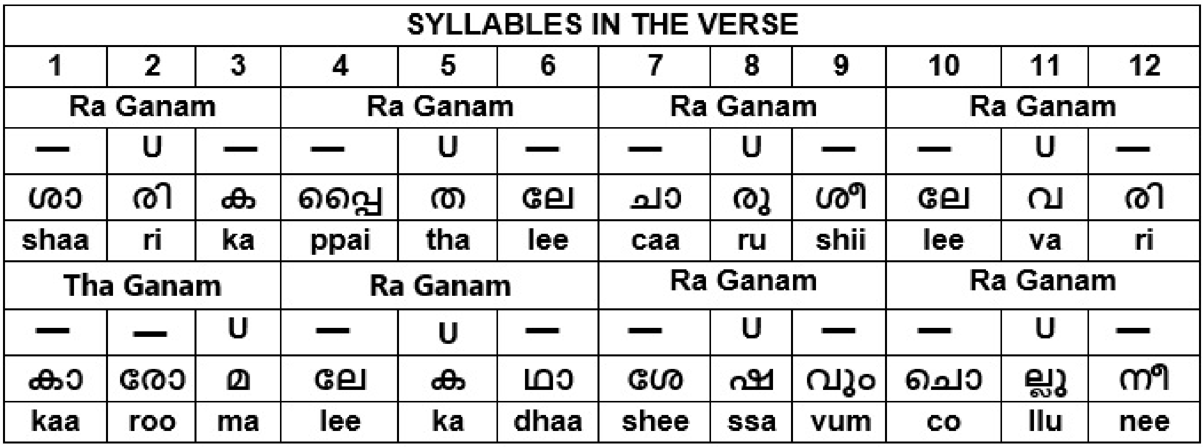
3.2.1 Features considered

The dataset consists of nineteen features and the the target duration for each of the syllables stored. These nineteen features are determined according to the Vruta Lakshanas discussed earlier, and contextual, phonological and accoustic properties of the syllables. The features considered for the proposed model are are listed below. More details about these features can be found Malayalam grammatical scripts like [37, 22].

1. Number of syllables in verse.

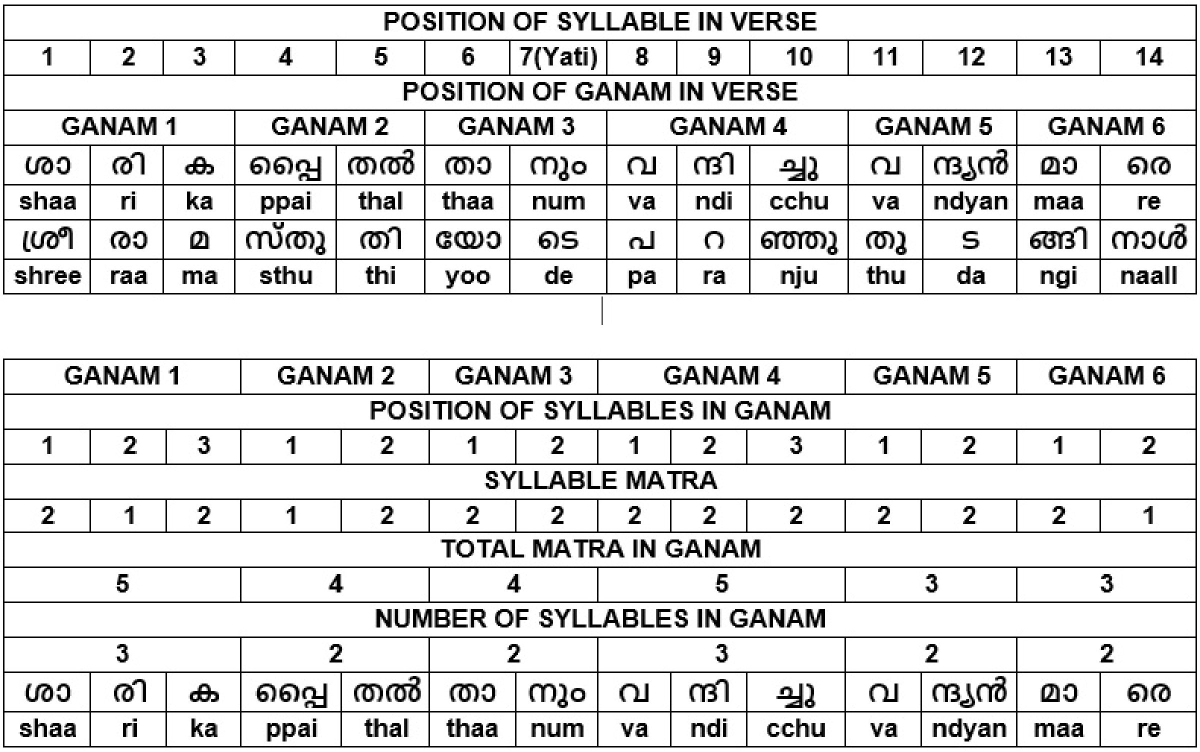
2. Position of syllable in verse.

3. Number of syllables in Ganam.



SYLLABLES IN THE VERSE											
1	2	3	4	5	6	7	8	9	10	11	12
Ra Ganam			Ra Ganam			Ra Ganam			Ra Ganam		
—	U	—	—	U	—	—	U	—	—	U	—
ശാ	രി	ക	പ്പൈ	ത	ലേ	ചാ	രു	ശീ	ലേ	വ	രി
shaa	ri	ka	ppai	tha	lee	caa	ru	shii	lee	va	ri
Tha Ganam			Ra Ganam			Ra Ganam			Ra Ganam		
—	—	U	—	U	—	—	U	—	—	U	—
കാ	രോ	മ	ലേ	ക	ഥാ	ശേ	ഷ	വും	ചൊ	ല്ലു	നീ
kaa	roo	ma	lee	ka	dhaa	shee	ssa	vum	co	llu	nee

Figure 1: Identification of the Vruta from the given couplet of a poem



POSITION OF SYLLABLE IN VERSE													
1	2	3	4	5	6	7(Yati)	8	9	10	11	12	13	14
POSITION OF GANAM IN VERSE													
GANAM 1			GANAM 2		GANAM 3		GANAM 4			GANAM 5		GANAM 6	
ശാ	രി	ക	പ്പൈ	തൽ	താ	നും	വ	ന്ദി	ച്ചു	വ	ന്ദ്യൻ	മാ	രെ
shaa	ri	ka	ppai	thal	thaa	num	va	ndi	cchu	va	ndyan	maa	re
ശ്രീ	രാ	മ	സ്തു	തി	യോ	ടെ	പ	റ	ഞ്ഞു	തു	ട	ങ്ങി	നാൾ
shree	raa	ma	sthu	thi	yoo	de	pa	ra	nju	thu	da	ngi	naall

GANAM 1			GANAM 2		GANAM 3		GANAM 4			GANAM 5		GANAM 6	
POSITION OF SYLLABLES IN GANAM													
1	2	3	1	2	1	2	1	2	3	1	2	1	2
SYLLABLE MATRA													
2	1	2	1	2	2	2	2	2	2	2	2	2	1
TOTAL MATRA IN GANAM													
5			4		4		5			3		3	
NUMBER OF SYLLABLES IN GANAM													
3			2		2		3			2		2	
ശാ	രി	ക	പ്പൈ	തൽ	താ	നും	വ	ന്ദി	ച്ചു	വ	ന്ദ്യൻ	മാ	രെ
shaa	ri	ka	ppai	thal	thaa	num	va	ndi	cchu	va	ndyan	maa	re

Figure 2: Syllable and Ganam related features in Keka

4. Total Matra in Ganam.
5. Position of Ganam in verse.
6. Position of syllable in Ganam.
7. Whether Yati or not.
8. Distance from Yati.
9. Yati Position.
10. Origin of Previous syllable.
11. Origin of Next syllable.
12. Origin of Current syllable.
13. Matra of Previous syllable.
14. Matra of Next syllable.
15. Matra of current syllable.
16. Whether previous syllable is Joint.
17. Whether next syllable is Joint.
18. whether current syllable is Joint.
19. Total Matra in verse.

3.3 Gradient Boost Ensembles

Four machine learning regressor estimators considered to model the duration of poem syllables are listed below. They all fall into the category of ensemble estimators where an ensemble of weak predictors are used to model the predictions.

1. Gradient Boost Machine
2. XGBoost
3. LightGBM
4. CatBoost

3.3.1 Gradient Boosting Machine(GBM)

GBM is a machine learning technique used for classification and regression problems [28]. The prediction model produced by them are formed as an ensemble of weak predictors, typically decision stumps. It is used to minimize the bias error of the model. The hyperparameters are optimized with GridsearchCV algorithm from the scikit-learn machine learning library [15].

3.3.2 XGBoost

XGBoost is principally a Gradient Boost method. It uses software and hardware optimizations to fasten the calculations. XGBoost implements the process of sequential tree building using parallelization. The stopping criterion for tree splitting within GBM framework is greedy in nature and depends on the negative loss criterion at the point of split. XGBoost uses max-depth parameter as specified and prunes trees backward. It improves the computational efficiency considerably. Also the algorithm uses cache awareness by allocating internal buffers in each thread to store gradient statistics.

3.3.3 LightGBM

LightGBM uses histogram-based algorithms, which sample continuous feature values into discrete bins. This aids in reducing training time and memory requirement. [10, 18]. LightGBM employs a leaf-wise (best-first) tree grow algorithm. It selects a leaf with maximum delta loss to grow. Holding the leaf fixed, leaf-wise algorithms tend to achieve lower loss than level-wise algorithms.[50]. LightGBM also optimizes network communication, makes use of distributed learning, implements Parallelization in finding best fit decision tree and decision learning to optimize the boosting process.

3.3.4 CatBoost

. CatBoost is an open source implementation of Gradient Boost algorithm by the engineers at Yandex. CatBoost makes use of distributed Graphic Processing Units (GPU) to fasten the training process. The CatBoost algorithm introduces a unique system called Minimal Variance Sampling (MVS), which is a weighted sampling version of Stochastic Gradient Boosting. With this technique, the number of examples needed for each iteration of boosting can be reduced considerably. The CatBoost algorithm grows a balanced tree. The feature-split pair is performed to choose a leaf in the tree structure. The split with minimum penalty is selected for all the level's nodes. Enough iterations are performed level by level until the leaves match the depth of the tree. The symmetric trees are faster and provides better quality compared to non symmetric trees.

3.4 Performance Evaluation Metrics

The performance of different Gradient Boost ensembles are compared using statistical error indicators MAE, RMSE, and MAPE. The strength of association between the points in the dataset is described using the coefficient of determination R^2 as in equation (1). This statistic indicates the percentage of the variance in the dependent variable (target duration in our case) that the independent variables (contextual features) explain collectively. R-squared measures the strength of the relationship between the model and the dependent variable on a convenient 0-100% scale. R-squared is a positive statistical indicator, the higher the percentage, the better the model fits the data. y_j is the actual duration of the syllables, $\hat{y_j}$ is the predicted duration and $\bar{y}$ is the mean duration of the syllables in the training sample set.

$$R^2 = \frac{\sum_{j=1}^{L}(y_j - \hat{y_j})^2}{\sum_{j=1}^{L}(y_j - \bar{y_j})^2} \tag{1}$$

The different statistical error indices used to evaluate the performances of the duration models are shown below from equation (2) to (5). All these indices are negative indicators, which means lower values are preferred. Mean absolute error (MAE)(2) is a measure of errors between pairs of actual and predicted observations. It is the arithmetic average of the absolute errors.

$$MAE = \frac{1}{n}\sum_{j=1}^{n} y_j - \hat{y_j} \tag{2}$$

The Mean Squared Error (MSE)(3) of an estimator measures the average of the squares of the errors between the estimated values and the actual value. MSE is a risk function, corresponding to the expected value of the squared error loss.

$$MSE = \frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y_j})^2 \tag{3}$$

The Root Mean Squared Error (RMSE)(4) represents the square root of the second sample moment of the differences between the predicted values and the actual values. The RMSE serves to aggregate the magnitudes of the errors in predictions for various data points into a single measure of predictive power.

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y_j})^2} \tag{4}$$

The Mean Absolute Percentage Error (MAPE)(5) is a measure of the accuracy of an estimator. The accuracy is measured as a percentage.

$$MAPE = \frac{1}{n}\sum_{j=1}^{n}|\frac{y_j - \hat{y_j}}{y_j}| \tag{5}$$

4 Results and Discussions

4.1 Model Training

The entire dataset is divided into two datasets for training and testing. The training dataset consists of 80% of the samples, whereas testing dataset holds the remaining 20%. A total of 25254 syllables are used for training and 6314 syllables are used for testing the models. Syllable duration modelling can be perceived as a regression problem, where the predicted duration is modelled as a function of the selected syllable features. All the Gradient Boost ensemble models are optimized using the GridsearchCV algorithm from scikit-learn library [15]. By exploring different permutations and combinations of the hyperparameters specified in the parameter grid, GridsearchCV() can optimize the values of these parameters.

4.2 Optimization of Hyperparameters

As for GBM regressor, the hyperparameters considered are the number of trees (n-estimators), the maximum depth of the tree (max-depth), the minimum number of samples required to split an internal node (min-samples-split), and the weight applied to each classifier at each boosting iteration (learning-rate). As for the XGBoost n-estimators, max-depth, min-samples-split, learning rate, and squared error loss function are used for tuning. The same parameters are used for LightGBM as well. As for CatBoost, n-estimators, max-depth, learning rate are used as hyperparameters along with Root Mean Squared Error as the loss function.

Table 3: Hyperparameters used in GBM optimization with their values

$Hyperparameter$	$Value$
n-estimators	500
max-depth	4
min-samples-split	5
learning-rate	0.01

Table 4: Hyperparameters used XGBoost optimization with their values

$Hyperparameter$	$Value$
n-estimators	500
max-depth	4
min-samples-split	5
learning-rate	0.01
loss	squared-error

Table 5: Hyperparameters used in LightBoost optimization with their values

Hyperparameter	*Value*
n-estimators	500
max-depth	4
min-samples-split	5
learning-rate	0.01
loss	squared-error

Table 6: Hyperparameters used in CatBoost optimization with their values

Hyperparameter	*Value*
n-estimators	500
max-depth	4
learning-rate	0.01
loss	RMSE

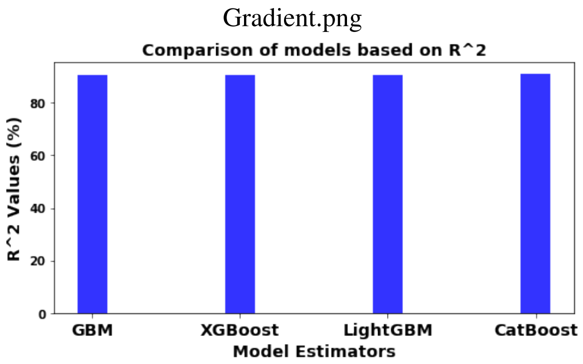


Figure 3: Comparison of Models based on R^2

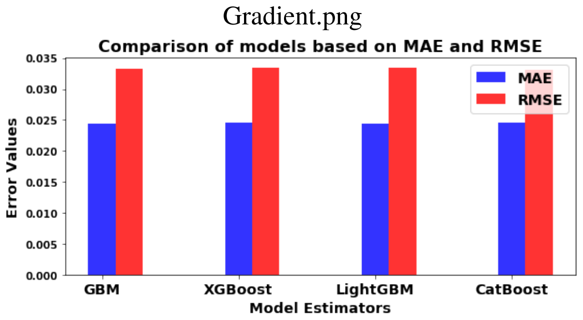


Figure 4: Comparison of Models Based on MAE and RMSE

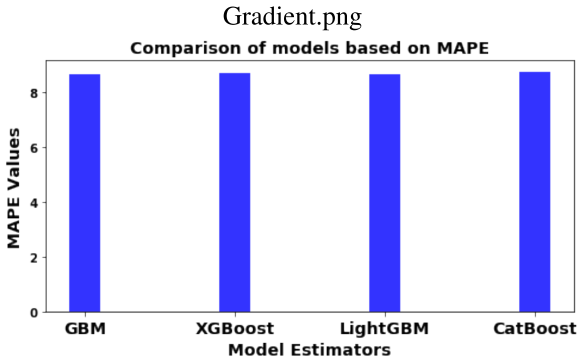


Figure 5: Comparison of Models based on MAPE

Tables 3-6 show the hyperparameters used along with the optimum values observed. Table 3 shows the hyperparameters tuned for optimization in GBM estimator. Table 4 shows the hyperparameters tuned for optimization in XGBoost estimator. Table 5 shows the hyperparameters tuned for optimization in LightBoost estimator. Table 6 shows the hyperparameters tuned for optimization in CatBoost estimator. Figure 3 compares the coefficient of determination for the four models. Figure 4 shows the comparison bar chart of MAE and RMSE for GBM, XGBoost, LightBoost and CatBoost ensembles.Figure 5 shows the comparison bar chart of MAPE for the ensembles.

Line graphs are excellent to compare changes of a variable over the same period of time for more than one group. Figure 6 and figure7 show the line plot and semilog plot of the actual and predicted duration of the first 100 values for GBM, XGBoost, LightBoost, and CatBoost ensembles.

4.3 Results and Performance Comparison

Table 7 shows the performance comparison metrics of the models. All the Gradient Boost algorithms yield similar values for R^2, MAE, RMSE, and MAPE. It can be said that, all the boosting techniques enhance the base estimator equally.

This is a pioneer effort to model Malayalam poem syllable duration. The comparison of duration models across languages can not be objective in nature as the contextual features, accoustic properties and interdependencies between the syllables are different in each language. Table 8 gives a a comparative analysis of the

Table 7: Performance of Gradient Boost Models

Model	R^2	MAE	RMSE	MAPE
GBM GridsearchCV	90.723	0.0244	0.033	08.659
XGBoost GridsearchCV	90.726	0.0245	0.0333	08.711
LightBoost GridsearchCV	90.693	0.0244	0.0333	08.658
CatBoost GridsearchCV	90.723	0.0244	0.0333	08.659

Table 8: Performance of Gradient Boosting Models Against Major Duration Models

Model	Author	R^2
CART	K. Srinivasa Rao and B. Yegnanarayana[40]	81
Two-Stage-SVM&ANN	K. Srinivasa Rao and B. Yegnanarayana[40]	82
ANN	K. Srinivasa Rao and B. Yegnanarayana[40]	82
CART	N. Shridhar Krishna et al.[17]	79.87
GBM-GridsearchCV	Proposed model	90.723
XGBoost-GridsearchCV	Proposed model	90.726
LightBoost-GridsearchCV	Proposed model	90.693
CatBoost-GridsearchCV	Proposed model	90.819

proposed model against different duration models proposed by K. srinivasa Rao and B. Yegnanarayana and N. Sridhar Krishna et al. [40, 17]. The duration model developed by K. Srinivasa Rao and B. Yegnanarayana is for Tamil language and the duration model suggested by N. Sridhar Krishna et al.is for Hindi language. To analyze the performances of different models objectively, standard datasets have to be established. It is evident from the table that Gradient Boost algorithms perform superior to ANN, Support Vector Regressors (SVR) or CART when the dataset is of small or medium scale in size.

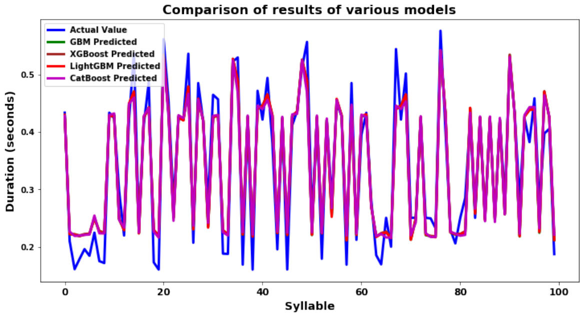


Figure 6: Line plot of the first 100 duration values

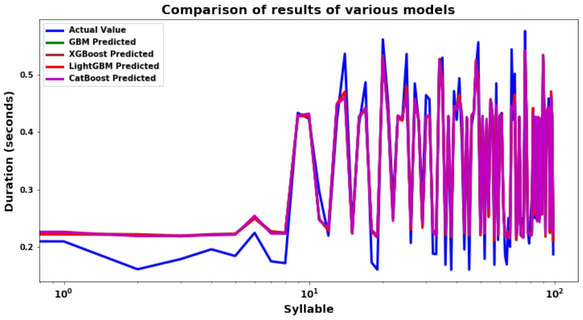


Figure 7: semilog plot of the first 100 duration values

5 Conclusion and Future Work

In this paper, the duration characteristics of Malayalam poem syllables written in three Vrutas are modelled using Gradient Boosting ensemble machine learning algorithms. A standard dataset consisting of 31568 syllables' features is published as part of this work. . This is a pioneer work in a low resource language like Malayalam. Reviewed and published datasets are a scarcity in Malayalam [24]. The shortage of standard datasets is the main restricting constraint to take the research activities in the language to next level. The proposed ensembles are able to emulate the durational characteristics of the language. Presently, only the textual parameters extracted from the orthographic representation of the poems are considered to model the duration. The research in the area can be further pushed by designing deep learning architectures to capture the features and transform them to time aligned intermediate representations such as mel spectrograms. These mel spectrograms can then be used as inputs to vocoder deep learning frameworks to synthesize speech [59, 35].

References

[1] Chomsky, N. *Syntactic structures*. De Gruyter Mouton, 2009.

[2] Datta, A. K. Intonation rules for text reading. In *Epoch Synchronous Overlap Add (ESOLA)*, pages 135–176. Springer, New York, USA, 2018.

[3] Ezhuthachan, T. *Adhyathma Ramayanam*. DC Books, Kottayam, Kerala, 2015.

[4] Gopinath, D. P., Sree, J. D., Mathew, R., Rekhila, S., and Nair, A. S. Duration analysis for malayalam text-to-speech systems. In *9th International Conference on Information Technology (ICIT'06)*, pages 129–132, New York, USA, 2006. IEEE.

[5] Gopinath, D. P., Veena, S., and Nair, A. S. Modeling of vowel duration in malayalam speech using probability distribution. *Proceedings of the Speech Prosody, Campinas, Brazil*, pages 6–9, 2008.

[6] Gopinath, D. P., Vinod, C. S., Veena, S., and Achuthsankar, S. N. A hybrid duration model using cart and hmm. In *TENCON 2008-2008 IEEE Region 10 Conference*, pages 1–4, New York, USA, 2008. IEEE.

[7] Hodari, Z., Moinet, A., Karlapati, S., Lorenzo-Trueba, J., Merritt, T., Joly, A., Abbas, A., Karanasou, P., and Drugman, T. Camp: a two-stage approach to modelling prosody in context. *arXiv preprint arXiv:2011.01175*, 2020.

[8] Huang, L., Zhuang, S., and Wang, K. A text normalization method for speech synthesis based on local attention mechanism. *IEEE Access*, 8:36202–36209, 2020.

[9] James, J. and Gopinath, D. P. Pause duration model for malayalam tts. In *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 2206–2210, New York, USA, 2015. IEEE.

[10] Jin, R. and Agrawal, G. Communication and memory efficient parallel decision tree construction. In *Proceedings of the 2003 SIAM international conference on data mining*, pages 119–129. SIAM, 2003.

[11] Kenter, T., Wan, V., Chan, C.-A., Clark, R., and Vit, J. Chive: Varying prosody in speech synthesis with a linguistically driven dynamic hierarchical conditional variational network. In *International Conference on Machine Learning*, pages 3331–3340. PMLR, 2019.

[12] Khan, R. A. and Chitode, J. Concatenative speech synthesis: A review. *International Journal of Computer Applications*, 136(3):6, 2016.

[13] Klatt, D. H. Synthesis of stop consonants in initial position. *The Journal of the Acoustical Society of America*, 47(1A):93–94, 1970.

[14] Klatt, D. H. Review of text-to-speech conversion for english. *The Journal of the Acoustical Society of America*, 82(3):737–793, 1987.

[15] Kramer, O. Scikit-learn. In *Machine learning for evolution strategies*, pages 45–53. Springer, New York, USA, 2016.

[16] Krishna, N. S. and Murthy, H. A. Duration modeling of indian languages hindi and telugu. In *Fifth ISCA Workshop on Speech Synthesis*, pages 197–202, Pittsburgh, USA, 2004. ISCA.

[17] Krishna, N. S., Talukdar, P. P., Bali, K., and Ramakrishnan, A. Duration modeling for hindi text-to-speech synthesis system. In *Proc. ICSLP*. ICSLP, 2004.

[18] Li, P., Wu, Q., and Burges, C. Mcrank: Learning to rank using multiple classification and gradient boosting. *Advances in neural information processing systems*, 20:897–904, 2007.

[19] Liu, F., Weng, F., Wang, B., and Liu, Y. Insertion, deletion, or substitution?: normalizing text messages without pre-categorization nor supervision. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 71–76. Association for Computational Linguistics, 2011.

[20] Mache, S. R., Baheti, M. R., and Mahender, C. N. Review on text-to-speech synthesizer. *International Journal of Advanced Research in Computer and Communication Engineering*, 4(8):54–59, 2015.

[21] Makhija, P., Kumar, A., and Gupta, A. hinglishnorm–a corpus of hindi-english code mixed sentences for text normalization. *arXiv preprint arXiv:2010.08974*, 2020.

[22] Marar, K. *Vrutha Shilpam*. The Mathrubhumi Printing and Publishing co, Ernakulam, Kerala, 1964.

[23] Menon, V. S. *Vyloppilli Kavithakal*. DC Books, Kottayam, Kerala, 2000.

[24] M.P, J. and Balakrishnan, K. Text to speech systems in south indian languages: A survey. In *National Conference On Indian Language Computing 2014*. Citeseer, 2014.

[25] M.P, J. and Balakrishnan, K. Identification and extraction of features to model duration of malayalam poem syllables. 2020.

[26] M.P, J. and Balakrishnan, K. Malayalam poem syllable duration dataset, 2021.

[27] Namboothiri, C. *Krishna Gadha*. DC Books, Kottayam, Kerala, 2020.

[28] Natekin, A. and Knoll, A. Gradient boosting machines, a tutorial. *Frontiers in neurorobotics*, 7:21, 2013.

[29] Nunes, R. D., Rosa, R. L., and Rodríguez, D. Z. Performance improvement of a non-intrusive voice quality metric in lossy networks. *IET Communications*, 13(20):3401–3408, 2019.

[30] Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[31] Pal, K. and Patel, B. V. Automatic multiclass document classification of hindi poems using machine learning techniques. In *2020 International Conference for Emerging Technology (INCET)*, pages 1–5, New York, USA, 2020. IEEE.

[32] Pal, K. and Patel, B. V. Model for classification of poems in hindi language based on ras. In *Smart Systems and IoT: Innovations in Computing*, pages 655–661. Springer, New York, USA, 2020.

[33] Ping, W., Peng, K., and Chen, J. Clarinet: Parallel wave generation in end-to-end text-to-speech. *arXiv preprint arXiv:1807.07281*, 2018.

[34] Prakash, J. J. and Murthy, H. A. Analysis of inter-pausal units in indian languages and its application to text-to-speech synthesis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(10):1616–1628, 2019.

[35] Prenger, R., Valle, R., and Catanzaro, B. Waveglow: A flow-based generative network for speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3617–3621. IEEE, 2019.

[36] Rajan, B. K., Rijoy, V., Gopinath, D. P., and George, N. Duration modeling for text to speech synthesis system using festival speech engine developed for malayalam language. In *2015 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015]*, pages 1–5, New York, USA, 2015. IEEE.

[37] Rajaraja Varma, A. *vruthamanjari*. Current Books, Kottayam, Kerala 686001, 1904.

[38] Rao, K. S. and Koolagudi, S. G. Selection of suitable features for modeling the durations of syllables. *Journal of Software Engineering and Applications*, 3(12):1107, 2010.

[39] Rao, K. S. and Yegnanarayana, B. Modeling syllable duration in indian languages using support vector machines. In *Intelligent Sensing and Information Processing, 2005. Proceedings of 2005 International Conference on*, pages 258–263, New York, USA, 2005. IEEE.

[40] Rao, K. S. and Yegnanarayana, B. Modeling durations of syllables using neural networks. *Computer Speech & Language*, 21(2):282–295, 2007.

[41] Rao, K. S. and Yegnanarayana, B. Intonation modeling for indian languages. *Computer Speech & Language*, 23(2):240–256, 2009.

[42] Reddy, V. R. and Rao, K. S. Intonation modeling using ffnn for syllable based bengali text to speech synthesis. In *Computer and Communication Technology (ICCCT), 2011 2nd International Conference on*, pages 334–339. IEEE, 2011.

[43] Reddy, V. R. and Rao, K. S. Two-stage intonation modeling using feedforward neural networks for syllable based text-to-speech synthesis. *Computer Speech & Language*, 27(5):1105–1126, 2013.

[44] Rodríguez, D. Z., da Silva, M. J., Silva, F. J. M., and Junior, L. C. B. Assessment of transmitted speech signal degradations in rician and rayleigh channel models. *INFOCOMP Journal of Computer Science*, 17(2):23–31, 2018.

[45] Rodríguez, D. Z. and Junior, L. C. B. Determining a non-intrusive voice quality model using machine learning and signal analysis in time. *INFOCOMP Journal of Computer Science*, 18(2), 2019.

[46] Rodríguez, D. Z., Rosa, R. L., Almeida, F. L., Mittag, G., and Möller, S. Speech quality assessment in wireless communications with mimo systems using a parametric model. *IEEE Access*, 7:35719–35730, 2019.

[47] Rosen, G. Dynamic analog speech synthesizer. *The Journal of the Acoustical Society of America*, 30(3):201–209, 1958.

[48] Sau, A., Amin, T. A., Barman, N., and Pal, A. R. Word sense disambiguation in bengali using sense induction. In *2019 International Conference on Applied Machine Learning (ICAML)*, pages 170–174. IEEE, 2019.

[49] Shen, J., Jia, Y., Chrzanowski, M., Zhang, Y., Elias, I., Zen, H., and Wu, Y. Non-attentive tacotron: Robust and controllable neural tts synthesis including unsupervised duration modeling. *arXiv preprint arXiv:2010.04301*, 2020.

[50] Shi, H. *Best-first decision tree learning*. PhD thesis, The University of Waikato, 2007.

[51] Shreekanth, T., Udayashankara, V., and Chandrika, M. Duration modelling using neural networks for hindi tts system considering position of syllable in a word. *Procedia Computer Science*, 46:60–67, 2015.

[52] Sproat, R., Black, A. W., Chen, S., Kumar, S., Ostendorf, M., and Richards, C. Normalization of non-standard words. *Computer speech & language*, 15(3):287–333, 2001.

[53] Stevens, K. N., Kasowski, S., and Fant, C. G. M. An electrical analog of the vocal tract. *The Journal of the Acoustical Society of America*, 25(4):734–742, 1953.

[54] Taylor, P. *Text-to-speech synthesis*. Cambridge university press, 2009.

[55] Terra Vieira, S., Lopes Rosa, R., Zegarra Rodríguez, D., Arjona Ramírez, M., Saadi, M., and Wuttisittikulkij, L. Q-meter: Quality monitoring system for telecommunication services based on sentiment analysis using deep learning. *Sensors*, 21(5):1880, 2021.

[56] Thomas, A. and Gopinath, D. P. Analysis of the chaotic nature of speech prosody and music. In *India Conference (INDICON), 2012 Annual IEEE*, pages 210–215. IEEE, 2012.

[57] Tripathi, K., Sarkar, P., and Rao, K. S. Sentence based discourse classification for hindi story text-to-speech (tts) system. In *Proceedings of the 13th International Conference on Natural Language Processing*, pages 46–54, 2016.

[58] Vieira, S. T., Rosa, R. L., and Rodríguez, D. Z. A speech quality classifier based on tree-cnn algorithm that considers network degradations. *Journal of Communications Software and Systems*, 16(2):180–187, 2020.

[59] Wang, Y., Skerry-Ryan, R., Stanton, D., Wu, Y., Weiss, R. J., Jaitly, N., Yang, Z., Xiao, Y., Chen, Z., Bengio, S., et al. Tacotron: Towards end-to-end speech synthesis. *arXiv preprint arXiv:1703.10135*, 2017.