



A Natural Language Processing-Based, Variation-Centric Input Handling Model for Social Computing Systems

Nor Azura Husin¹, Balasubramaniyan Divager²

¹Department of Computer science, Universiti Putra Malaysia, 43400 Serdang Selangor, Malaysia

²Faculty of computer science and information technology, University putra Malaysia
gs54205@student.upm.edu.my

Abstract:

Systems for social computing use environmental data to provide interactive applications that integrate software and computer power. Social computing systems with multi-input management have enhanced processing power and application speed because of their audible and readable characteristics. The multi-variant language processing problem in social computing that leads to computational failures is discussed in this article. A Variation-Centric Processing Model (VCPM) is used to mitigate breakdown errors caused by overlapping speech inputs in the aforementioned situation. By enhancing the similarity feature, the suggested model seeks to maximize the text input processing rate while accounting for different pronunciations and tones. The first input is regarded as the baseline upon which subsequent categorization and correlation are based. Based on the high processing rate, this initial reference is updated on a regular basis, after which additional classifications are carried out. Regressive mistakes, which are recognized as discrepancies in the processing of text and speech inputs, are used to reduce the variances in a regressive categorization pattern. By modifying the choices for regularized efficiency, this improves the efficiency of the social computing system. Processing rate, latency, error, accuracy, and dispersion are the measures used to validate the performance of this model.

Keywords: NLP, Regression Learning, Social Computing System, Speech Processing.

1. Introduction

An interconnection among social application and computation systems is known as a social computing system. Social computing creates a connection between software and technology which provides better communication and services to users which helps to improve the whole experience [1]. A social computing system is a collection of many smart applications which are available on the web. It also provides better security and privacy policies to users which help to improve the overall performance of the application [2]. The social computing system is used in merging applications and provides a collaborative process among the social application which is used to improve user experience. Many social applications are done based on a social computing system such as Facebook and YouTube [3]. Managing a larger amount of data in an application leads to many problems during the computation process. Scale-independent storage (SCADS) is used in a social computing system to enhance the storage capacity which helps to increase the efficiency and effectiveness of the system. SCADS used SQL language to queue the data which are stored in the database. It also reduces the duplicate rate in the storage process which helps to increase the storage space for further uses [4, 5].

Natural Language Processing (NLP) is a process that is used to understand and manipulate data that are connected to a computer to generate a proper dataset for the



data processing system. NLP is used to understand and analyze linguistic content which is sent by the users in an application [6]. NLP understands text or speech which is given by the users. NLP is mainly used to analyze human understanding capabilities to perform particular tasks. NLP is mostly used in web-based applications to understand users' thoughts to provide better service at a needed time without any delay or failure [7]. NLP is widely used to understand, learning and produce human language processes. NLP-based application mostly uses a deep learning approach to avoid unnecessary problems and failures. NLP is used in social computing-based applications to understand the communication process among the users and is used to improve the overall performance in the communication process [8]. NLP also provides a communication process among the users and the machines or devices to understand the exact needs of users. In a social computing system, NLP is mainly used for the word segmentation process and keyword extraction process which plays a vital role in social applications. The learning algorithm is used in a social computing system for the word extraction process which helps to identify every detail of content or text before providing service to the users [9, 10].

NLP is used in social computing system-based applications to enhance the accuracy rate in providing services to the users by understanding the exact requests. Deep learning techniques are widely used in NLP applications which are used to improve the overall performance and efficiency of the system [11]. Deep learning-based NLP application analysis various features by performing some outlining process. Artificial intelligence (AI) is also used in NLP to understand the content of the text-based on knowledge [12]. AI utilizes human knowledge which helps to understand or calculate the text from a human point of view and provides an accurate set of data for the computation process. Cross-language information retrieval (CLIR) is used in social computing applications which are used to translate a foreign language into an appropriate form to understand for providing services to the users [13, 14]. Natural language is used here to understand the queries which are given by the users. CLIR is a knowledge-based approach that is used to understand the lexicon-grammar to formalize unknown content. CLIR improves the overall accuracy rate in the translation process and also increases the efficiency of the applications [15].

2. Related Works

Li et al. [16] proposed a deep neural network-embedded Internet of Social Computing Things (NeSoc) for sustainability prediction process. Deep Neural Network approach is used here to calculate the computation process which is available in the model. It is used to calculate and collect data which are used for computation process. Experimental results show that the proposed NeSoc method increase the overall performance of the system and also reduces computation cost.

Ait-Mlouk et al. [17] introduced a knowledge graph by based on chatbot approach (KBot) for natural language understanding (NLU) process. Intent classification method is used in NLU for providing better understanding over the users. The proposed KBot method increases the overall flexibility, reliability and efficiency of NLU process. When



compared with other knowledge-based techniques, the proposed KBot method increases the performance of the system by providing proper interactive process.

Baydogan et al. [18] proposed a novel approach named Ant lion optimization (ALO) and moth flame optimization (MFO) algorithm for automatic Hate speech detection (HSD) process. The proposed approach is used to identify HSD issues such as swearing, intolerance, insults, and humiliation. ALO method is used to increase the accuracy rate in detection hate speech. It is an efficient and flexible approach while provides a better detection process for automatic HSD. Experimental results show that the proposed method increase the overall performance of the system.

Jindal et al. [19] introduced a novel semiautomatic for ontology or terrorism domain analysis process which is used in social media analytics. Terrorism attacks in social media are one of the dangerous attacks, the proposed method is used to prevent users from the attackers and secure user's personal details. Query-based system is used here to extract the knowledge or details which are available in the database. Fluent editor is used in the proposed method to identify the terrorism attack by understanding the content which are sending by the users.

Paredes et al. [20] proposed a new malicious behavior detection process for social medial application by using hybrid knowledge driven system. The proposed method is used to identify the uses that posted or published fake news in social media networks. Also detect the fake news reporter and prevents the users from viewing the fake news. Experimental results show that the proposed method increases the accuracy rate in detection process and improves the efficiency of the system.

Chen et al. [21] introduced a new method named FritzBot by bidirectional long short-term memory (BiLSTM) network. Conditional random field (CRF) is also in FritzBot to train the dataset which are available in the database for detecting fake texts. FritzBot generates a circuit by analysing the actual text which is sent by the users in social media. Experimental results show that the FritzBot method increases the overall performance of the system by reducing the computation cost, latency rate and energy consumption rate in detection process.

O'Halloran et al. [22] proposed a multimodal approach which is used for analyzing process in social media application. The proposed method is used to understand the communication process among the users and the organization which helps to produce a proper mapping system. The proposed method improves the overall computation process in big data analytics process.

Tsuji et al. [23] introduced a blueprint for computational model which is for social learners in social interaction process. The proposed method is used to identify the communication process among the users which widely used by social learns to identify the fake news reporters. Computational mechanism is used here to understand the exact content of the text and helps to extract the information which is available in the database. Experimental results show that the proposed method increases the overall performance of the system.

Alkouz et al. [24] proposed a road traffic analysis process named SNSJam by using cross-lingual data for social media networks. Cross-lingual data are collected from dynamic source which plays a vital role in classification process. Location recognizer is



used in SNSJam to locate the traffic jams which are occurred in roads. The proposed method also identifies the cause of traffic jams and produces a solution to avoid unwanted traffic jams. When compared with other methods, the proposed SNSJam method increases the accuracy rate in traffic jam detection process.

Alarifi et al. [25] introduced an optimized cognitive-assisted machine translation (MT) approach for natural language processing. Machine learning approach is used in the proposed to understand the high-quality content and helps to translate the actual details of the text. A smart decision-making tool is used in the proposed MT approach to understand the efficiency and effectiveness of the content. Experimental results show that the proposed method increases the overall performance of the MT approach by reducing the computation cost and by increasing the accuracy rate in translation process.

Al-Makhadmeh et al. [26] proposed a new automatic hate speech detection method by using natural language processing. The proposed method is used in social media applications to predict the hate speech by analysing the good speech frames. An effective learning process is used here to classify the speech by based on good, bad, hate or informal language. Experimental results show that the proposed method increases the overall accuracy rate in detecting hate speech which helps to improve the performance of the social media application.

Ni et al. [27] proposed a natural language understanding (NLU) approach which is used for Internet of Things (IoT) voice interaction process by using joint task of intent detection method. Slot filling joint task is also used in NLU for human interaction process in IoT based applications. Intent detection process is used to improve the slot filling task which is used to provide significant building for human voice interaction process in IoT. When compared with other approaches, the proposed NLU increases the joint model's accuracy rate and provide better interaction process of users.

Fontes et al. [28] introduced a deep learning framework for utilizing Twitter data to examine urban mobility. It uses Natural Language Processing (NLP) methods for data cleansing, tweet classification, and emotion analysis. Using this approach, we may better understand public opinion and trends in urban mobility by comparing the accessibility of three major cities: London, Melbourne, and New York. Transportation planners can benefit from its high accuracy in text classification and ability to aggregate data in near real-time. However, there needs to be more research on scalability and generalizability.

Li et al. [29] proposed an interaction graph network (IGN) for adverse drug reaction (ADR) recognition process. ADR detection is widely used in social media to address the users by with the help of surveillance cameras. The proposed IGN is used to identify the lexicon type of drug which is one of the difficult tasks to perform. Lexicon is identified by encoding the context which is available in the database by using graph attention network. Experimental results show that the proposed IGN method increases the accuracy rate in detecting ADR.



3. Proposed Model

The proposed VCPM model is designed to improve the classification and correlation accuracy of the voice inputs for backhauling interactive applications. The NLP application model regulates through assimilating software and computational abilities in an available and readable feature on better accuracy for its functions. The proposed model of VCPM language detecting method is designed to control the functions of the computing systems employed in a social computing systems. In a social computing system environment, voice capturing devices such as Speakers, Cortana-Microsc, Apple Home pod, Amazon Echo, Google Home, etc. are used that is recorded or recognized through computing systems. The computing systems are a combination of software and hardware terminals to recognize and gather voice input accumulated from the computing system environment. The voice recognition is equipped with signals detecting units to record or recognize information such as voice, text, etc. In the proposed VCPM precise output and computing systems are devised using the processing recommendations based on the interactive applications. A social computing system framework consists of centralized computing system unit that manages the operations of the devices through voice input and text (V). The function of the computing systems is regulated using wireless signals and Google keyboard. The VCPM method functions between interactive applications and social computing systems. VCPM is one such method that ensures use of regression learning for attribute classification and correlation of voice input. The proposed model is illustrated as in Figure 1.

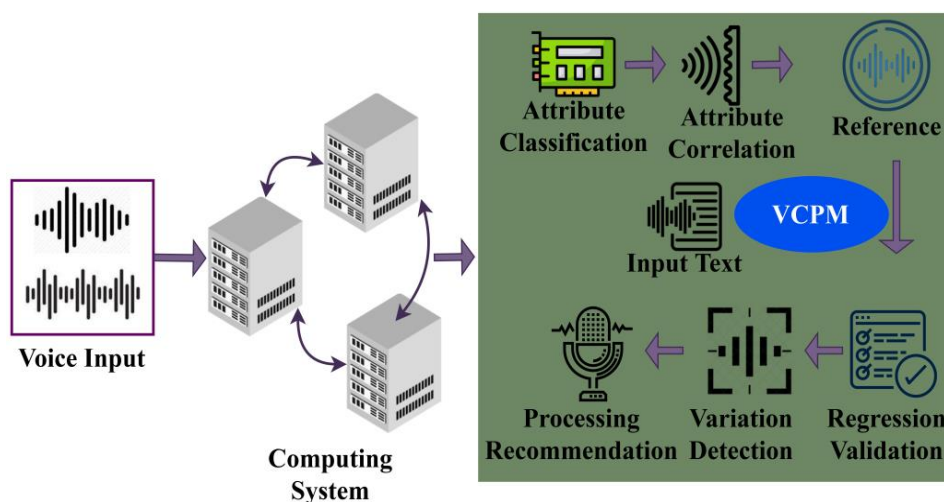


Fig 1. Proposed Model

The proposed model intakes voice commands or voice notes for performing defined operations. The computing system classifies the available attributes for correlation and reference finding. Based on the identified references, the regression learning and variation detection are performed. The recommendations are provided based on the achievable accuracy ensured in the process (Figure 1). The processing function of VCPM in NLP based application acquires inputs through voice signals (V_1) observed in the human. The voice inputs are performed for attributes classification and correlation using



regressive learning using the features audible and readable. The attributes classification output is aided for detecting the observation through attributes correlation from the gathered information. The function of VCPM is notices the multi-variant language processing problem in social computing that outputs in computational breakdowns where the voice initially filtered. The input of voice signal is represented as

$$V_I = \frac{1}{T} \left| \sum_{s=1}^T v_x(s) - t_y(s) \right| \left. \begin{array}{l} \text{where,} \\ v_x(T) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{v_x(T)}{v_x(T)} dt \\ \text{and} \\ t_y(T) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{t_y(T)}{T} dt \end{array} \right\} \quad (\text{Eq.1a})$$

Where, the variables $v_x(s)$ and $t_y(s)$ represent the voice and text of the V_I with respect to wavelength transform in x and y planes and s is the observed signal, respectively. If x and y planes are increasing and decreasing edges with respect to time and then $x \in [0, \infty]$ and $y \in [-\infty, 0]$ and therefore,

$$\left. \begin{array}{l} v_x(T) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{v_x(T)}{T} dt \\ \text{such that} \\ t_y(T) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{t_y(T)}{T} dt \end{array} \right\} \quad (\text{Eq.1b})$$

Based on the above equation (2), the initial voice input is suppressed for all $v_x(T) + t_y(T)$ that represents a overall instance relies on x and y wavelengths for a given interval of time on $(q \times T)$. Here, q is the total queuing process. The queuing process is computed to reduce the processing rate present in V_I . Voice input is due to the decent quality sound captured in the computing system while acquiring V_I in any time interval T . This computing system follows a high processing rate of queuing that is computed as follows.

$$\left. \begin{array}{l} v_x(T) = \frac{v_x(T)}{T} * 2^{\frac{1}{2}} \Delta_i [q \times T - L^2] \\ t_y(T) = \frac{t_y(T)}{T} * 2^{\frac{1}{2}} \Delta_j [q \times T - L^2] \\ \text{where,} \\ \Delta_i = f(T) \left(\frac{L}{2}\right) f(T)_{L-1} \text{ and} \\ \Delta_j = \bar{f}(T) \left(\frac{L}{2}\right) f(T)_{L-1} \end{array} \right\} \quad (\text{Eq.2})$$

Based on equation (2), the variables Δ_i and Δ_j is the queuing transform function for low and high processing rate. The factors of $f(T)$ and $\bar{f}(T)$ represents the frequency of transformation and routine transformation functions of Δ_i and Δ_j . From the sequence of occurrence of the partial wavelength (i.e.) x or y , the frequency of input transformation and routine transformation functions are used. Where, the variable L denotes the voice input length used in both the processing of wavelength identification. Here, the frequency based V_I is defined as



$$\left. \begin{aligned}
 V_1[f(T)] &= \frac{1}{2^2} \frac{[(q \times T) - L^2]}{T^2} [\Delta_i - \Delta_j] \\
 &\text{and} \\
 V_1[f(T)] &= \frac{1}{T} \left[\int_0^\infty \frac{\Delta_i [(q \times T) - L^2]}{T} dt - \int_{-\infty}^0 \frac{\Delta_j [(q \times T) - L^2]}{T} dt \right]
 \end{aligned} \right\} \text{(Eq.3)}$$

From the computational breakdown of voice input less $V_1[f(T)]$ that is accessed after applying queue. From this multi-input management of $V_1[f(T)]$, two features namely audible and readable are extracted for further attribute classification. The audible (A_D) and readable (R_D) features is computed as

$$\left. \begin{aligned}
 A_D &= \frac{1}{2\pi(q \times T)} \left(\sum_{s=1}^t (v_x - t_y) e^{-\alpha s j} \right), \text{ if } \forall j = i + 1, i \in L \\
 &\text{Such that} \\
 R_D &= - \sum_{i=\beta_L}^{\beta_h} A_{D_i} \log A_{D_i}
 \end{aligned} \right\} \text{(Eq.4)}$$

Where, e is the error occurrence at the multi-input management processing, α is the point lies on the normal plane, β_h and β_L are the high and low frequency rate of A_D observed. The log recognition of A_D and R_D for the $f(T)$ is estimated as

$$R_D[f(T)] = \frac{R_D}{\log \left[\frac{T}{\beta_h - \beta_L} \right]} \text{(Eq.5)}$$

This log voice recognition is computed for synchronization $R_D[f(T)]$ and R_D alone with the various time intervals and time instances. The attribute classification process is based on A_D and $R_D[f(T)]$ using regression learning. The attribute classification process based on log recognition is presented in Figure 2.

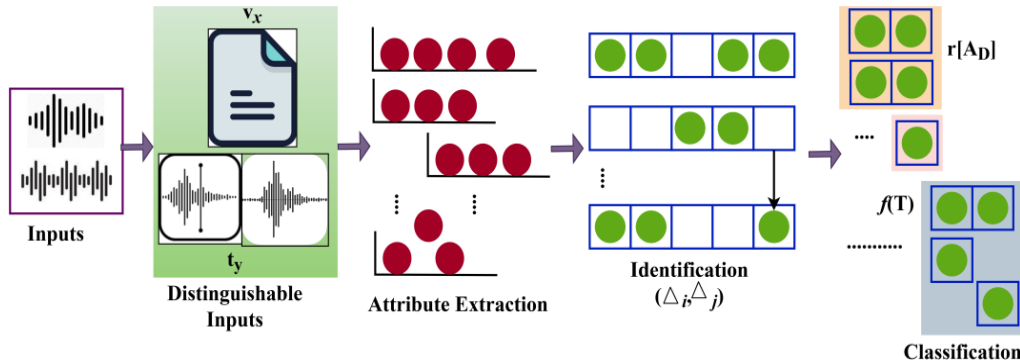


Fig 2. Attribute Classification Process

The inputs are first distinguished for v_x and t_y for extracting available attributes. The queuing process is responsible for defining the functions in distinguished intervals, for further classification. This attributes classification helps to maximize the text input by augmenting the similarity functions regardless of varying tones and pronunciations for all $v_x(T)$ or $t_y(T)$ or both wavelength (Figure 2). In this attribute classification, the similarity features are independently analyzed at each level followed by the joint attribute classification. The initial level is considered as the instance based on which the further processing take place. The first instance of reference are determined as in following equation (6a) and (6b) for A_D and $R_D[f(T)]$.



$$r[A_D, T, f(T)] = - \sum_{i=1}^q T_i - \sum_{j=1}^T T_j - \sum_{i=1}^q \sum_{j=1}^T T_i \alpha_i \left. \begin{array}{l} \text{where} \\ I[A_D, T] = \frac{e^{-r[A_D, T, f(T)]}}{\sum_{i=1}^{q \times T} e^{-r[A_D, T, f(T)]_i}} \end{array} \right\} \quad (\text{Eq. 6a})$$

As per the above equation (6a), r represent the regression function for A_D and I is the first input of reference at T . Similarly, the first instance of inputs and consequence are given for $R_D[f(T)]$ as

$$r[R_D[f(T)], A_D] = \left\{ \begin{array}{l} - \sum_{i=1}^q T_i \alpha_i \frac{1}{\Delta_i}, \text{ if } v_x(T) \in [0, \infty] \\ - \sum_{i=1}^q T_i \alpha_i \Delta_i, \text{ if } v_x(T) \notin [0, \infty] \end{array} \right\} \quad (\text{Eq. 6b})$$

and $I[R_D[f(T)]] = \frac{e^{-r[R_D[f(T)], A_D]}}{\sum_{i=1}^T e^{-r[R_D[f(T)], A_D]_i}}$

From equation (6b) the similarity verification and reference of the regression learning is presented such that $r[R_D[f(T)], A_D]$ is evaluated for both the wavelengths $\in V_1[f(T)]$. This process helps to determine the computing system based on T to ease possible attribute classification instances. The similarity verification of the regression learning process based on $V_1[f(T) \in [-\infty, \infty]]$ and $v_s \in [0, \infty]$ or $v_s \notin [0, \infty]$. The condition of $v_s \in [0, \infty]$ succeeds a stage of $[-\infty, 0]$ that inversely denotes t_y at a given time instance T . The condition of the $V_1[f(T)]$ is autonomously using $I[.]$ where in the sequence of $I[.]^*$ represents that the condition of $v_s \in [0, \infty]$ is experienced and therefore the existing (previous) $I[.]$ does not keep for further computations. As per the above derivation, $v_s \in [0, \infty]$ are identified as the regressive errors $t_y \notin [0, \infty]$ is another choice and therefore, now $v_s \in [0, \infty]$ and $v_s \notin [0, \infty]$ is considered. These regressive errors (RE) are mismatched with the voice and text input process to obtain the final solution (z). Hence, the attribute classification and correlation output sequence of $\{c_1 \text{ to } c_{q \times T}\}$ for both the voice input and text is estimated. The attribute classification and attribute correlation sequence output along with highest accuracy α that are estimated as

$$\left. \begin{array}{l} c_1 = V_1[f(1)] \\ c_2 = V_1[f(2)] - \frac{\alpha_1}{r[A_D, T, f(T)]_1} \\ \vdots \\ c_{q \times T} = V_1[f(q \times T)] - \frac{\alpha_{q \times (T-1)}}{r[A_D, T, f(T)]_{q \times (T-1)}} \end{array} \right\} \text{, if } \forall V_1 = v_x(T) + t_y(T) \quad (\text{Eq. 7})$$

Instead,

$$\left. \begin{array}{l} c_1 = V_1[f(1)] - \alpha_{\theta-1} \times r[R_D[f(1)], A_D]_{\theta-1} \\ c_2 = V_1[f(2)] - \alpha_{\theta} \times r[R_D[f(2)], A_D]_{\theta} \\ \vdots \\ c_{q \times T} = V_1[f(q \times T)] - \alpha_{q \times T} \times r[R_D[f(q \times T)], A_D]_{q \times T} \end{array} \right\} \text{, if } \forall V_1[f(T) = v_x(T) + t_y(T)] \quad (\text{Eq. 8})$$

In the above equations (7) and (8), θ is the point at which $V_1[f(T)]$ is determined from $V_1 = v_x(T) + t_y(T)$ to $v_x(T)$ or $t_y(T)$ alone. If it comes to the reference of regressive validation, then similarity is the considerable factor. The correlation process using regressive learning is presented in Figure 3.

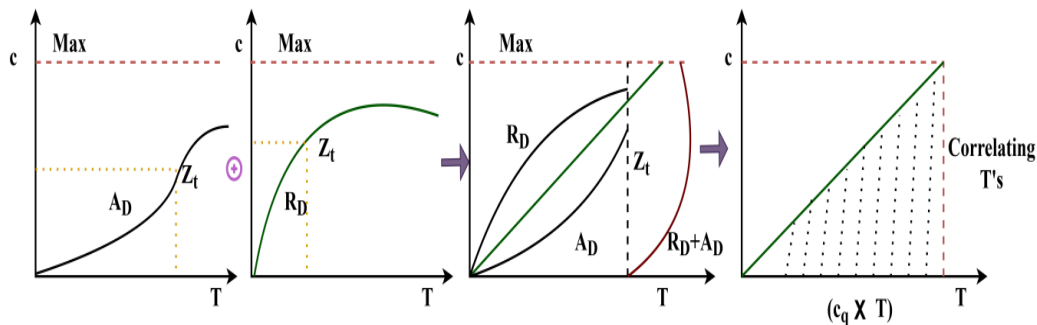


Fig 3. Correlation Process

In the correlation process, the maximization is focused for the reference inputs in different T. The correlation in $c_{q \times T}$ is achieved for non-overlapping R_D and A_D such that the joint interference are augmented for V_I . Therefore, $V_I[f(T)]$ is considered as the performance of T in improving processing rate (Figure 3). This is ensured for a time interval and then the first reference is changed periodically based on the high processing rate and then further classifications are performed. Here, the first reference output for $V_I[f(T)] \neq v_x(T)$ is given as

$$\left. \begin{aligned}
 c_1 &= V_I[f(1)] - \alpha_{\theta-1} \times \frac{1}{T} \{r[R_D[f(1)], A_D]_{\theta-1}\} \\
 c_2 &= V_I[f(2)] - \alpha_{\theta} \times \frac{1}{T} \left\{ \begin{aligned}
 &r[R_D[f(2)], A_D]_{\theta} - \\
 &r[R_D[f(2)], A_D]_{\theta-1}
 \end{aligned} \right\} \\
 &\vdots \\
 c_{q \times T} &= V_I[f(q \times T)] - \alpha_{q \times T} \times \frac{1}{T} \left\{ \begin{aligned}
 &r[R_D[f(q \times T)], A_D]_{q \times T} - \\
 &r[R_D[f(q \times (T - 1)), A_D]_{q \times (T-1)}
 \end{aligned} \right\}
 \end{aligned} \right\} , \forall V_I[f(T) = v_x(T) + t_y(T)] \quad (Eq.9)$$

Where, the variable α is estimated as in the following equation (9)

$$\alpha = \frac{1}{T} \left[\frac{c_{q \times T} \in v_x}{c_{q \times T} \in (v_x + t_y)} + \frac{c_{q \times T} \notin v_x}{c_{q \times T} \in (v_x + t_y) + c_{q \times T} \in v_x} \right] \quad (Eq.10)$$

The condition on the x,y plane represents the periodical changes of θ such that if $c_{q \times T} \notin v_x$ or $c_{q \times T} \in v_x \leq \theta < c_{q \times T} \in (v_x + t_y)$ is the precise output for classifying and correlating $V_I[f(T)]$. If the above mentioned condition is not satisfied, then the delay increases by one. Similarly, if $c_{q \times T} \in v_x$ is obtained, then $RE = RE + 1$ else processing rate (i.e.) $PR = PR + 1$. In the attribute classification process, it is prominent to increase the PR other than the existing RE or features. For this assessment, both attribute classification and correlation is performed for all the overlapping speech outputs as in above equations. The regression validation between the references with the already available attributes helps to validate the occurrence of RE, PR and breakdown error BE, that helps to augments the precision of classification. In the computation of regressive validation, the output is discontinuous due to θ take place. Hence, the output does not share the routine iterations for all $T = \{1, \text{to } T \times q\}$. If an overlapping speech inputs occur, then $RE = RE + 1$ and this evaluation is consolidated with respect to time t as



$$\left. \begin{aligned} \frac{zt}{dt} &= \left(\frac{C_{V_1}t}{dt} + \frac{(1-\alpha)C_x t}{dt} \right) - \alpha \frac{C_y t}{dt} \\ \text{Where} \\ C_{V_1} &= c_{q \times T} \in (v_x + t_y) \\ \text{and} \\ C_x &= c_{q \times T} \in v_x \\ C_y &= c_{q \times T} \notin t_y \end{aligned} \right\} \text{(Eq.11)}$$

The autonomous computing system with respect to t differentiation is increased as the single swiftness of the result z . This solution is not the last as the last obtaining is based on $\frac{zt}{dt}$ using the attribute classification and correlation with high accuracy estimation. Similarly, periodical change in reference differ the precision of RE and PR by evaluating α with the C_{V_1} based on the solution of $\frac{zt}{dt}$ computing system analysis. The variation due to overlapping using regressive learning is presented in Figure 4. The correlation detection method is presented for Z and T wherein the α maximization is required. In the different combinational inputs, the precise Z is alone identified for reducing c_y and its variations. The method of $C_{V_1} + (1 - \alpha)C_x$ overlaps at any t such that, either C_{V_1} or C_x is increased (Figure 4). In this model, the reference with high PR is augmented for the social computing system. The following equation represents the regressive validation for deriving z based on the system efficiency, respectively.

$$\left. \begin{aligned} \int_{-\infty}^{\infty} \frac{zt}{dt} &= \int_{-\infty}^{\infty} \frac{C_{V_1}t}{dt} = \int_{-\infty}^{\infty} \left(V_1[f(q \times T)] - \frac{\alpha_{q \times (T-1)}}{r_{[A_D, T, f(T)]_{q \times (T-1)}}} \right) \frac{T}{dt} \\ z &= \int_{-\infty}^{\infty} \left(V_1[f(q \times T)] - \frac{\alpha_{q \times (T-1)}}{r_{[A_D, T, f(T)]_{q \times (T-1)}}} \right) \frac{T}{dt}, \text{ if } \alpha_N = 1 \forall T = 1 \text{ to } N \times t \\ z &= \int_{-\infty}^{\infty} V_1[f(q \times T)] \frac{T}{dt}, \forall r[.] = 1 \text{ and } C_{V_1}[f(T)] = v_x(T) + t_y(T) \end{aligned} \right\} \text{(Eq.12)}$$

In this sequence of output of the computing system function as $\alpha = r[.] = 1$, the solution does not adapting any readable or RE. In the similar manner, the computing system assessment of C_x , C_y and z as per α is denoted, respectively. In this time the variation detection of the classification pattern is as follows

$$\left. \begin{aligned} \int_0^{\infty} \frac{zt}{dt} &= \frac{1}{T} \int_0^{\infty} C_x \times \sum_{i=\theta-1}^N (1 - \alpha_i) \times \frac{T}{dt}, \\ z &= \frac{1}{T} \int_0^{\infty} V_1[f(q \times T)] - \alpha_0 r[R_D[f(q \times T)]] \times \sum_{i=\theta-1}^N (1 - \alpha_i) \frac{T}{dt} \end{aligned} \right\} \text{(Eq.13)}$$

In above equation (13), the condition $\frac{(1-\alpha)}{T}$ serves as the processing recommendations. This condition is identified as mismatched in processing voice and text inputs using RE that aids the social computing system efficiency by changing the decisions for regularized efficiency of $\frac{\alpha}{T}$ for processing recommendations C_y outputs computed as

$$\left. \begin{aligned} \int_{-\infty}^0 \frac{zt}{dt} &= \frac{1}{T} \int_{-\infty}^0 C_y \frac{T}{dt} = \frac{1}{T} \int_{-\infty}^0 V_1[f(q \times T)] - \frac{\alpha_n}{T} \{ r[.]_{\theta} - r[.]_{q(T-1)} \} \\ z &= \frac{1}{T} \int_{-\infty}^0 \frac{zT}{dt} = \frac{1}{T} \int_{-\infty}^0 V_1[f(q \times T)] - \frac{\alpha_N r[.]_{\theta}}{T} = \frac{1}{T^2} \int_{-\infty}^0 T \times V_1[f(q \times T)] - \alpha_N B[.]_{\theta} \end{aligned} \right\} \text{(Eq.14)}$$

From the classification pattern functions solution of z for all $c_{q \times T} \notin v_x$ or $c_{q \times T} \in v_x \leq \theta < c_{q \times T} \in (v_x + t_y)$. It represents that $V_1[f(q \times T)]$ is a condition of t such that in the various sequences are responsible for increasing the evaluation. Instead in a variation detection of z relies on α , the entire factor is $\frac{1+\alpha_N-1-2\alpha_N}{T}$ is the estimation factor. If $c_{q \times T} \notin$



$v_x \notin c_{q \times T} \in (v_x + t_y)$, the RE is incremented that is compressed for its estimation using $\frac{1 + \alpha_{N-1} - 2\alpha_N}{T}$. This augments the processing rate, accuracy, distribution and reduces the delay and error. Table 1 presents the A_D and R_D observations for different features.

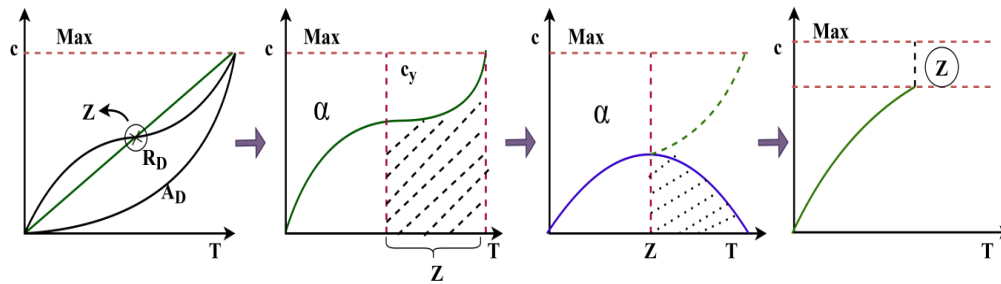


Fig 4. Variations in Regressive Pattern

Table 1 presents the R_D and A_D classifications for differed features observed. The $R_D[f(T)]$ is indefinite as R_D and A_D is uncertain wherein the further processing is initiated. This distinguishes Z based on classification alone and thus, if A_D varieties, then R_D shows up the difference. This is observed until $v_x \leq \theta$ is satisfied. On the other contrary, if $\theta < c_{q \times T}$, then R_D is high and A_D is less. This is balanced in distributing the inputs without observing α . The different reference instance required for multiple features is tabulated in Table 2.

Table 1: A_D and R_D Observations for Features

Features	Log Recognition	A_D	R_D
4	0.46	4	12
8	0.79	6	9
12	0.61	5	11
16	0.82	7	8
20	0.93	10	5
24	1	11	0

Table 2: Reference Instances for Features

Features	Correlating Sequences	Overlapping Sequences	Reference Instances
4	4	0	1
8	6	0	2
12	8	1	2
16	12	1	3
20	16	2	4
24	19	5	4

An observation analysis for the reference instances are presented in Table 2 for different features. The overlapping in $\frac{Zt}{dt}$ is identified based on the C_1 to $c_{q \times T}$ instances. This is observed in different sequence is augmented without compromising Z in c_y outcomes, maximizing the references. The distribution factor observed for different features is presented in Table 3.



Table 3: DISTRIBUTION FACTOR FOR FEATURES

Features	A_D Reference	R_D	Distribution
4	1	2	0.86
8	1	2	0.88
12	2	2	0.91
16	2	1	0.92
20	3	1	0.95
24	3	0	1

The input distribution observed in different features extraction is presented in Table 3. The distribution relies on A_D and R_D based on $(q \times T)$ observed. In the $\frac{\alpha}{T}$ maximizing instances, the $R_D[f(T)]$ is considered for defining further Δ_i or Δ_j . Therefore based on the queuing factor, the reference is defined for $r[.]$ until $c = 1$ is achieved. This requires concurrency in Z detection wherein the input is overloaded for the observed $\frac{Z_t}{dt}$. Therefore, the distribution is high if both A_D and R_D is balanced with the previous and current $f(T)$. Figure 5 illustrates the analysis for variations and queuing factor under different features.

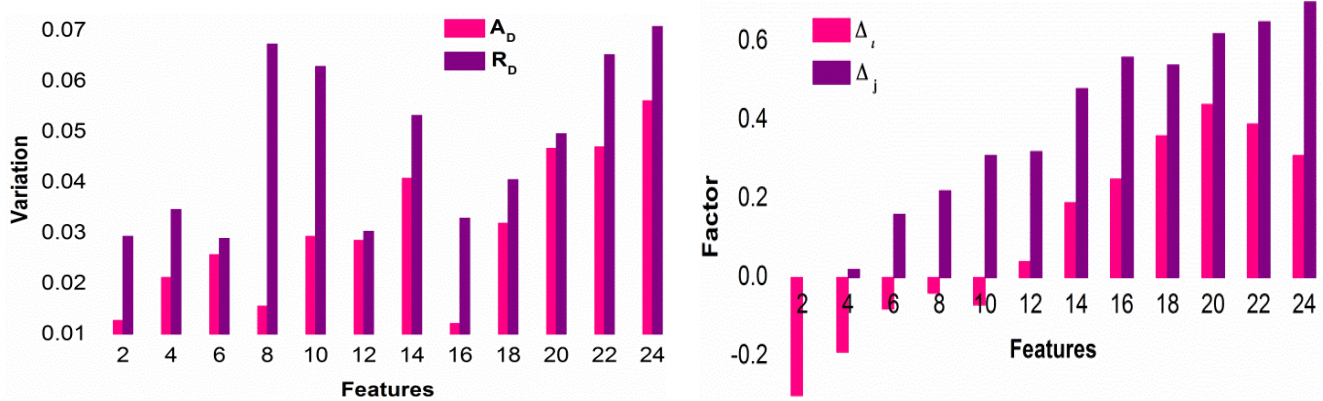


Fig 5. Variations and Queuing Factor Analysis

Figure 5 presents the (A_D, R_D) variation, and (Δ_i, Δ_j) factor for different V_i features observed. The A_D feature is classified based on $r[A_D, T, f(T)]$ for which Z in equation (9) and (13) are considered. An $\frac{Z_t}{dt}$ is required of a R_D overlaps A_D in which a non-linear results is obtained. The Δ_i or Δ_j is utilized for any one of A_d or R_D for which the $f(T)$ is positive. Contrarily for both the inputs, the A_D, R_D based on the reference, the distribution is performed. In distinct classification sequences, the (Δ_i, Δ_j) factor is balanced to be attained in a positive hike maximizing the accuracy.

4. Results and Discussion

This subsection describes the VCPM’s performance using MATLAB simulations. For this simulation, voice command and text inputs from [30] are utilized for operating mobile devices and Alexia. The processing is performed for 260 combined inputs, from which 24 features are extracted. The voice notes vary from 1s to 14s for providing commands in operating devices. With this implication, the metrics processing rate, delay, error, accuracy, and distribution are analyzed. For a better



decision, above metrics for the existing ALO+MFO [18], KNLPEDNN [26], and OSMT-SMT [25] are compared.

a. Processing Rate

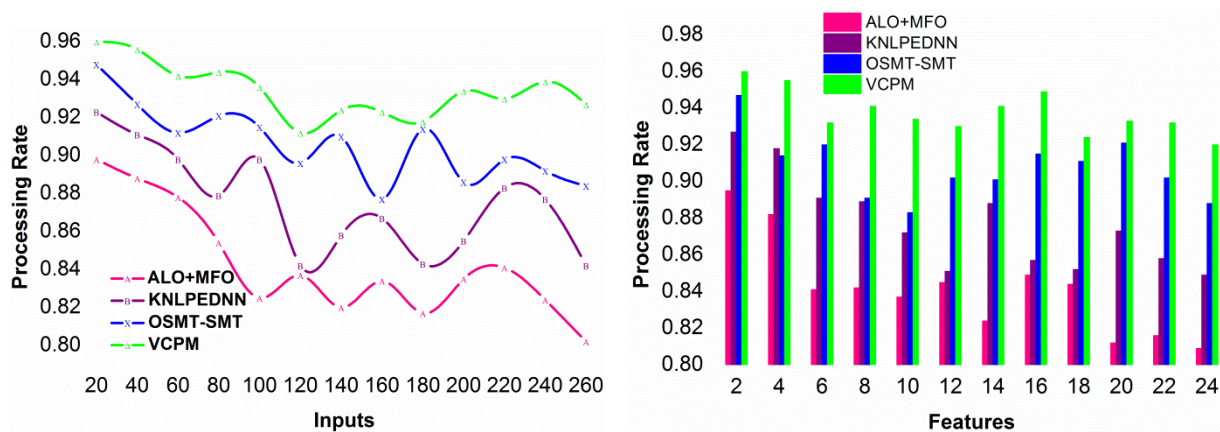


Fig 6. Analysis for Processing Rate

Figure 6 illustrates the comparative analysis for processing rate under different inputs and features. In the proposed model, $v_x(T)$ and $t_y(T)$ are distinguished for their A_D and R_D respectively, the $c_{q \times T}$ process is independently performed for the above features. In the first process, $r[A_D, T]$ and $r[R_D[f(T)]]$ is estimated without α consideration. Contrarily, in the $q \times T$ validation, the $\frac{Zt}{dt}$ defines the existence of Z under regressive incidents, preventing its influence on α . Therefore, in the variation detection process the processing is improved from the $R_D[f(T)]$ for R_D in a non-regressive manner. In the consecutive sequences, the $\frac{Zt}{dt}$ is identified under controlled variations, preventing failures. Therefore, (Δ_i, Δ_j) instances for single input based on features are improved. This improves the feature based correlations as in equations (7), (8) and (9). The empty slot for correlating sequence validation requires further processing under distinguished A_D, R_D maximizing the processing rate.

b. Delay

A comparative analysis for delay is presented in Figure 7 observed for different features and inputs. As seen in the illustration, the proposed model achieves less delay compared to the other methods. In the proposed model, the c 's sequence is distinguished for $V_1 = v_x(T) + t_y(T), \theta$ and $\frac{1}{T}$ respectively. This classification is required for $V_1[f(T) \neq v_x(T)]$ and hence the variations are suppressed such that Δ_i and Δ_j are varied based on multiple A_D and R_D . The regressive sequence verifies α based on all these instances such that $R_D[f(T)]$ is required in mitigating $\frac{Zt}{dt}$. Therefore, the $(V_x + t_y)$ induced in error factor is confined under either of the equation (7), (8), or (9). Therefore, one sequence is suspended without maximizing the variation. Hence forth, delay is observed in any one instance, preventing additional computations.



Depending on $\frac{\alpha}{T}$ in all the three variations, the $f(T)$ is remodeled for adapting delay less processing. Hence the α retained intervals are modeled based on distinct c_y for preventing additional delay. This is harmonized for the features and inputs observed in different instances.

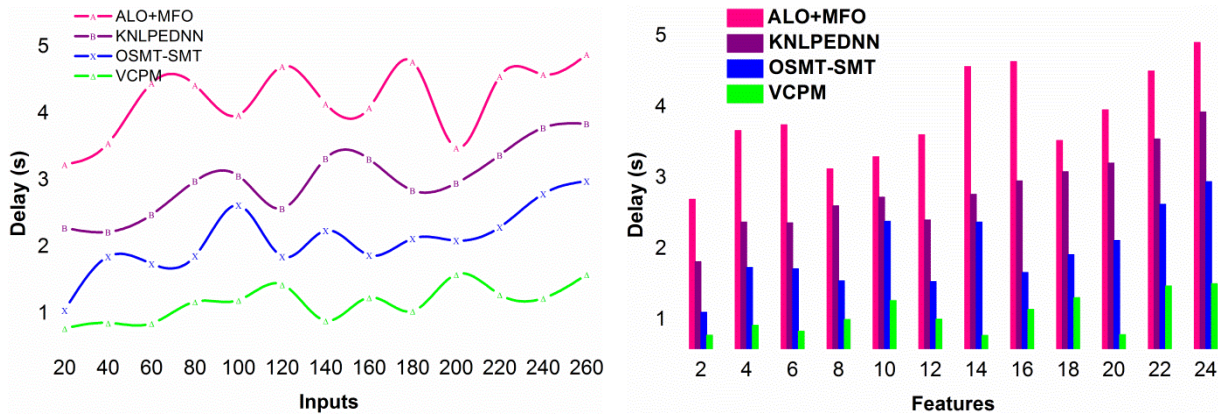


Fig 7. Analysis for Delay

c. Error

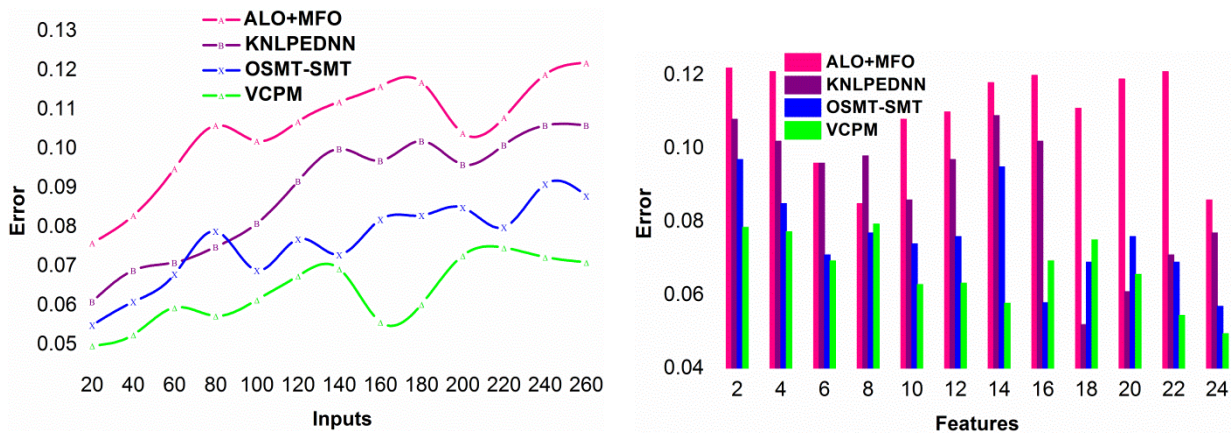


Fig 8. Analysis for Error

The proposed model achieves less error by maximizing α by mitigating Z under different conditions. First the $\frac{Zt}{dt}$ is analyzed for variations in regression and classification patterns. In the continuous iterations, $\forall T \in 1$ to $T \times q$, the overlapping case is identified for $R \in$ instances. Based on the iterations, $c_x \in v_x$ and $c_x \notin t_y$ are identified without c_{v_x} estimation. This estimation does not detect any possible correlation in c_1 to $c_{q \times T}$ instances. Contrarily, the overlapping $(1 - \alpha)$ in c_x is detected for identifying Z in different classifications. This is mitigated under $\frac{\alpha}{T}$ variations. The consecutive iterations correlates with $R_D[f(T)]$ or Δ_i and Δ_j for preventing final errors. The regression function distinguishes $\frac{Zt}{dt}$ and Z for c_y for reducing $(1 - \frac{\alpha}{T})$ and hence improving $c_{(q \times T)}$. The correlation for distinct features increases the error decreases



and hence the successive possibilities are validated. The consecutive sequences are validated based on $R_D[f(T)]$ such that Δ_i and Δ_j are reorganized in RE. This implication is pursued only if the inputs increase for which Z [as in equation (13)] is observed. Therefore, the classification is performed based on features, for reducing errors (Figure 8).

d. Accuracy

The proposed model achieves high accuracy for different features and inputs as presented in Figure 9. The α focused performance of the regression is presented using Z and $\frac{Zt}{dt}$ validations in this model. Depending on the available $\frac{(1-\alpha)}{T}$ estimations, the further R_D and A_D differentiation is performed. In the sequential process, $c_{q \times T}$ is ensured for satisfying $\alpha = r[.] = 1$ wherein the accuracy is maximized. Regardless of the $f(T)$ for the modified sequences, $R_D[f(T)]$ is confined based on the independent references estimated in equations (6a) and (6b). Therefore, as defined for the regressive sequences, c_1 to $c_{q \times T}$ is classified for either of Z as estimated in equations (12), (13) or (14). In this process, $C_{v_i} \in c_{q \times T}$, $\alpha_N = 1$, and $v_x \leq \theta c_{q \times T}$ are independently defined for leveraging the α factor. However as the inputs increase, the α factor is initially unstable due to classification issues wherein it defaces. This is mitigated based on different intervals such that error due to $\frac{Zt}{dt}$ is confined. The process is repeated until the c for its maximum limit is achieved as distinguished by the correlating T factor. Hence the pursuing regression is modified based on Z and previous $r[.] = 1$ as defined by $f(T)$. This maximizes the accuracy factor for the proposed model.

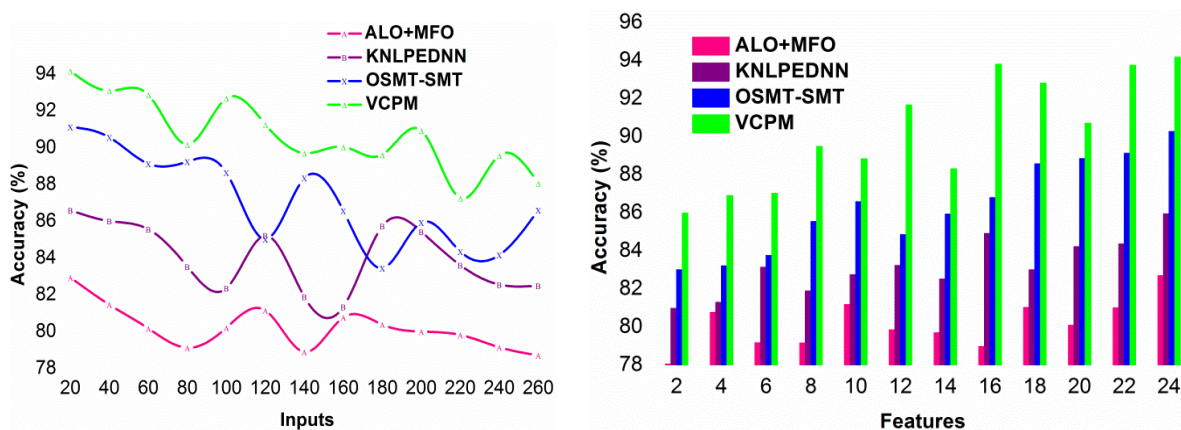


Fig 9. Analysis for Accuracy

e. Distribution

The proposed model is found to have high input distribution factor compared to the other methods. In the computational instances, the Δ_i and Δ_j determines the input requirement and $R_D[f(T)]$ validation. The input is initially not classified wherein $R_D \oplus A_D$ jointly performs the classification without leveraging the error. In the successive classifications as in equation (12), (13) and (14), the process is



independent accepting multiple inputs. Therefore, the need for inputs and feature extraction is high compared to the previous $f(T)$. This jointly exploits $v_I = v_x(T) + t_y(T)$ in $f(T)$ defined instances and hence $(q \times T)$ is shared for the above A_D and R_D defined regressions. In the proposed model, the incoming V_I are allocated with the existing R_D or A_D processing instances, classified by the features. This provides space for further inputs in the class increasing the distribution. Another classification is the concurrent utilization of the $r[.]$ as defined in equations (6a) and (6b). Therefore, the available and Δ_i and Δ_j induced V_I are distributed swiftly awaiting further $q \times T$. Therefore, this factor leverages without variations in distinguished inputs and features. Tables 4 and 5 present the comparative analysis summary for different inputs and features respectively.

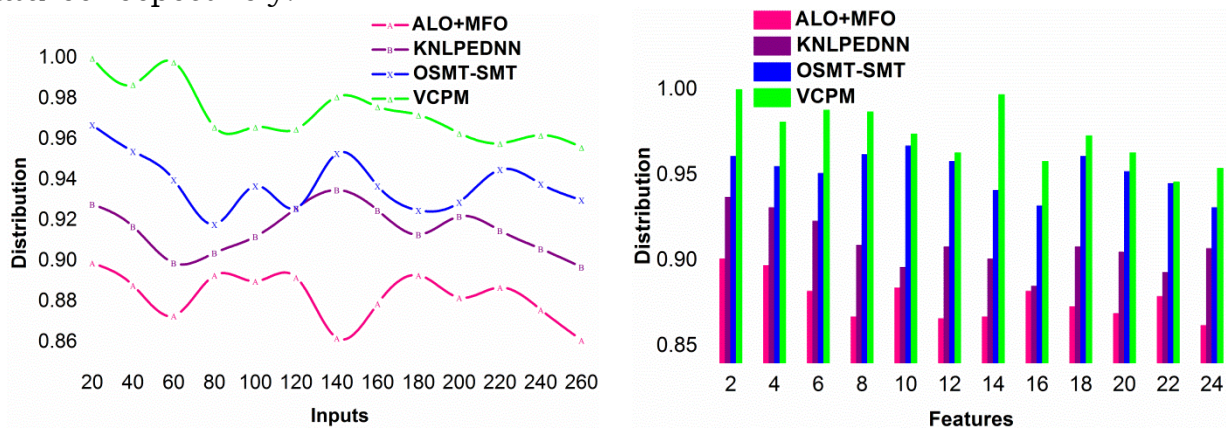


Fig 9. Analysis for Distribution

5. Conclusion

For addressing the error-based issues and variations in multi-input voice processing, this article introduced a variation-centric processing model. In this model, the voice and text inputs are classified for their available features. The identified features are used for defining independent reference plots for detecting variations. In the plot mapping, the linear regression learning is induced for identifying correlation and overlapping sequences. Based on the regression output, the reference points are modified for which the variations are processed using different conditions. In this process, accuracy and processing rate are operated for modifying the reference and hence the variations in joint input processing are mitigated. The regressive classifications result in error, variation, and overlapping reference estimation for which the computations are repeated until maximum processing and accuracy are achieved. From the experimental results, it is seen that for the varying features, the proposed model achieves 7.13% high processing rate, 10.26% less delay, 6.79% less error, 7.88% high accuracy, and 5.4% high distribution.



6. References

- [1]. Seneviratne, O., Adams, K., & McGuinness, D. L. (2023). Accountable Bench-to-Bedside Data-Sharing Mechanism for Researchers. *ACM Transactions on Social Computing*.
- [2]. Dragos, V., Battistelli, D., & Kellodjoue, E. (2020). A formal representation of appraisal categories for social data analysis. *Procedia Computer Science*, 176, 928-937.
- [3]. Zhang, Q., Tang, C., Bai, T., Meng, Z., Zhan, Y., Niu, J., & Deen, M. J. (2021). A two-layer optimal scheduling framework for energy savings in a data center for Cyber-Physical-Social Systems. *Journal of Systems Architecture*, 116, 102050.
- [4]. Valinejad, J., Mili, L., Yu, X., van der Wal, C. N., & Xu, Y. (2023). Computational social science in smart power systems: Reliability, resilience, and restoration. *Energy Conversion and Economics*.
- [5]. Nikravan, M., & Kashani, M. H. (2022). A review on trust management in fog/edge computing: Techniques, trends, and challenges. *Journal of Network and Computer Applications*, 204, 103402.
- [6]. Shahbazi, Z., & Byun, Y. C. (2021). Fake Media Detection Based on Natural Language Processing and Blockchain Approaches. *IEEE Access*, 9, 128442-128453.
- [7]. Yuvaraj, D., Ahamed, A. M. U., & Sivaram, M. (2021). A study on the role of natural language processing in the healthcare sector. *Materials Today: Proceedings*.
- [8]. Nicolas, C., Kim, J., & Chi, S. (2021). Natural language processing-based characterization of top-down communication in smart cities for enhancing citizen alignment. *Sustainable Cities and Society*, 66, 102674.
- [9]. Mazzei, D., Chiarello, F., & Fantoni, G. (2021). Analyzing social robotics research with natural language processing techniques. *Cognitive Computation*, 13(2), 308-321.
- [10]. Franzosi, R., Dong, W., & Dong, Y. (2021). Qualitative and quantitative research in the humanities and social sciences: how natural language processing (NLP) can help. *Quality & Quantity*, 1-31.
- [11]. Song, W., Guo, J., Fu, R., Liu, T., & Liu, L. (2020). A Knowledge Graph Embedding Approach for Metaphor Processing. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29, 406-420.
- [12]. Dessì, D., Osborne, F., Recupero, D. R., Buscaldi, D., & Motta, E. (2021). Generating knowledge graphs by employing Natural Language Processing and Machine Learning techniques within the scholarly domain. *Future Generation Computer Systems*, 116, 253-264.
- [13]. Jackson, P. C. (2021). On achieving human-level knowledge representation by developing a natural language of thought. *Procedia Computer Science*, 190, 388-407.
- [14]. Li, H. X., Lu, B., Chen, X., Li, X. Y., Castellanos, F. X., & Yan, C. G. (2021). Exploring self-generated thoughts in a resting state with natural language processing. *Behavior research methods*, 1-19.
- [15]. Alhogail, A., & Alsabih, A. (2021). Applying machine learning and natural language processing to detect phishing email. *Computers & Security*, 110, 102414.
- [16]. Li, Q., Song, Y., Du, B., Shen, Y., & Tian, Y. (2020). Deep Neural Network-Embedded Internet of Social Computing Things for Sustainability Prediction. *IEEE Access*, 8, 60737-60746.
- [17]. Ait-Mlouk, A., & Jiang, L. (2020). KBot: a Knowledge graph based chatBot for natural language understanding over linked data. *IEEE Access*, 8, 149220-149230.
- [18]. Baydogan, C., & Alatas, B. (2021). Metaheuristic Ant Lion and Moth Flame Optimization-Based Novel Approach for Automatic Detection of Hate Speech in Online Social Networks. *IEEE Access*, 9, 110047-110062.
- [19]. Jindal, R., Seeja, K. R., & Jain, S. (2020). Construction of domain ontology utilizing formal concept analysis and social media analytics. *International Journal of Cognitive Computing in Engineering*, 1, 62-69.
- [20]. Paredes, J. N., Simari, G. I., Martinez, M. V., & Falappa, M. A. (2021). Detecting malicious behavior in social platforms via hybrid knowledge-and data-driven systems. *Future Generation Computer Systems*, 125, 232-246.



-
- [21]. Chen, T., Xu, L., & Zhu, K. (2021). FritzBot: A data-driven conversational agent for physical-computing system design. *International Journal of Human-Computer Studies*, 155, 102699.
- [22]. O'Halloran, K. L., Pal, G., & Jin, M. (2021). Multimodal approach to analysing big social and news media data. *Discourse, Context & Media*, 40, 100467.
- [23]. Tsuji, S., Cristia, A., & Dupoux, E. (2021). SCALa: A blueprint for computational models of language acquisition in social context. *Cognition*, 104779.
- [24]. Alkouz, B., & Al Aghbari, Z. (2020). SNSJam: Road traffic analysis and prediction by fusing data from multiple social networks. *Information Processing & Management*, 57(1), 102139.
- [25]. Alarifi, A., & Alwadain, A. (2020). An optimized cognitive-assisted machine translation approach for natural language processing. *Computing*, 102(3), 605-622.
- [26]. Al-Makhadmeh, Z., & Tolba, A. (2020). Automatic hate speech detection using killer natural language processing optimizing ensemble deep learning approach. *Computing*, 102(2), 501-522.
- [27]. Ni, P., Li, Y., Li, G., & Chang, V. (2020). Natural language understanding approaches based on joint task of intent detection and slot filling for IoT voice interaction. *Neural Computing & Applications*, 32(20).
- [28]. Fontes, T., Murçós, F., Carneiro, E., Ribeiro, J., & Rossetti, R. J. (2023). Leveraging Social Media as a Source of Mobility Intelligence: An NLP-Based Approach. *IEEE Open Journal of Intelligent Transportation Systems*.
- [29]. Li, Z., Yang, Z., Wang, L., Zhang, Y., Lin, H., & Wang, J. (2020). Lexicon Knowledge Boosted Interaction Graph Network for Adverse Drug Reaction Recognition from Social Media. *IEEE Journal of Biomedical and Health Informatics*.
- [30]. Speech_commands. (n.d.). Retrieved October 20, 2023, from TensorFlow website: https://www.tensorflow.org/datasets/catalog/speech_commands