ИТМО

НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ
ноябрь–декабрь 2023 Том 23 № 6 http://ntv.ifmo.ru/
SCIENTIFIC AND TECHNICAL JOURNAL OF INFORMATION TECHNOLOGIES, MECHANICS AND OPTICS
November–December 2023 Vol. 23 No 6 http://ntv.ifmo.ru/en/
ISSN 2226-1494 (print) ISSN 2500-0373 (online)

НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

# КОМПЬЮТЕРНЫЕ СИСТЕМЫ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

# COMPUTER SCIENCE

doi: 10.17586/2226-1494-2023-23-6-1143-1151

## Emotion analysis of social network data using cluster based probabilistic neural network with data parallelism

**S. Starlin Jini[1]✉, N. Chenthalir Indra[2]**

[1,2] S.T Hindu College, Nagercoil, 629002, India

[1] starlinjini@gmail.com✉, https://orcid.org/0000-0002-8791-7481
[2] chenthalirindra@gmail.com, https://orcid.org/0009-0001-9121-5386

**Abstract**
Social media contains a huge amount of data that is used by various organizations to study people's emotions, thoughts and opinions. Users often use emoticons and emojis in addition to words to express their opinions on a topic. Emotion identification from text is no exception, but research in this area is still in its infancy. There are not many emotion annotated corpora available today. The complexity of the annotation task and the resulting inconsistent human comments are a challenge in developing emotion annotated corpora. Numerous studies have been carried out to solve these problems. The proposed methods were unable to perform emotion classification in a simple and cost-effective manner. To solve these problems, an efficient classification of emotions in recordings based on clustering is proposed. A dataset of social media posts is pre-processed to remove unwanted elements and then clustered. Semantic and emotional features are selected to improve classification efficiency. To reduce computation time and increase the efficiency of the system for predicting the probability of emotions, the concept of data parallelism in the classifier is proposed. The proposed model is tested using MATLAB software. The proposed model achieves 92 % accuracy on the annotated dataset and 94 % accuracy on the WASSA-2017 dataset. Performance comparison with other existing methods, such as Parallel K-Nearest Neighboring and Parallel Naive Byes Model methods, is performed. The comparison results showed that the proposed model is most effective in predicting emotions compared to existing models.

**Keywords**
emotions, clustering, feature extraction, probabilistic neural network and data parallelism

**For citation:** Starlin Jini S., Chenthalir Indra N. Emotion analysis of social network data using cluster based probabilistic neural network with data parallelism. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2023, vol. 23, no. 6, pp. 1143–1151. doi: 10.17586/2226-1494-2023-23-6-1143-1151

УДК 519.86

## Эмоциональный анализ данных социальных сетей с использованием кластерной вероятностной нейронной сети с параллелизмом данных

**С. Старлин Джини[1]✉, Н. Ченталир Индра[2]**

[1,2] Индуистский колледж Южного Траванкора, Нагеркойл, 629002, Индия

[1] starlinjini@gmail.com✉, https://orcid.org/0000-0002-8791-7481
[2] chenthalirindra@gmail.com, https://orcid.org/0009-0001-9121-5386

**Аннотация**
Социальные сети содержат огромное количество данных, которые используются различными организациями для изучения эмоций, мыслей и мнений людей. Пользователи часто используют смайлы и эмодзи в дополнение к словам, чтобы выразить свое мнение по обсуждаемой теме. Идентификация эмоций в тексте также требует изучения, однако исследования в этой области все еще находятся в начальном состоянии. Сегодня доступно недостаточно наборов данных с аннотациями интенсивностей эмоций. Сложность задачи аннотирования эмоций и дальнейшие комментарии пользователей становятся проблемами при разработке новых наборов данных. Для решения этих проблем выполняются многочисленные исследования. Разработанные методы

не смогли осуществить классификацию эмоций простым и экономичным способом. В настоящей работе представлена модель эффективной классификации эмоций в записях на основе кластеризации. Набор данных записей в социальных сетях предварительно обработан для удаления нежелательных элементов и далее кластеризован. С целью повышения эффективности классификации выбраны семантические и эмоциональные признаки. Для сокращения времени вычислений и повышения эффективности системы прогнозирования вероятности эмоций предложена концепция параллелизма данных в классификаторе. Предложенная модель апробирована с использованием программного обеспечения MATLAB. В результате модель обеспечила точности для аннотированного набора данных — 92 %, а для WASSA-2017 – 94 %. Выполнен анализ производительности описанной модели с существующими методами, такими как Parallel K-Nearest Neighboring и Parallel Naive Byes Model. Результаты сравнения показали, что предложенная модель наиболее эффективно предсказывает эмоции по сравнению с существующими.

**Ключевые слова**
эмоции, кластеризация, извлечение признаков, вероятностная нейронная сеть и параллелизм данных

**Ссылка для цитирования:** Старлин Джини С., Ченталир Индра Н. Эмоциональный анализ данных социальных сетей с использованием кластерной вероятностной нейронной сети с параллелизмом данных // Научно-технический вестник информационных технологий, механики и оптики. 2023. Т. 23, № 6. С. 1143–1151 (на англ. яз.). doi: 10.17586/2226-1494-2023-23-6-1143-1151

## Introduction

The growth of big data services and technologies is evaluated in growth in the market, and it is clearly seen that adopters like Yahoo, Facebook and eBay have briefly presented the value of mining complex datasets, as far as companies raised keen to release their value of own data [1]. Emotions like anger, happiness and sadness are states that humans experience and share as status in social media. The developed field of cognitive computing, which mimics and understands the functions of the human brain, has a wide area of research opportunities [2]. Experimenting with this big emotion data results in various challenges [3].

On these mechanisms, the big data process becomes problematic as they involve complex ecosystems. Thus, there is a need for a tool for big data management and infrastructure to deal with the issues for controlling the complicated environment [4]. Certain tools, namely Yarn and Mesos, help to solve the issues, but unfortunately, they were unable to compensate for the issue of cluster performance and optimizing the application. Big data get distributed among clusters for calculations, and parallelization was controlled through policies and its programming average [5]. However, there is a need for optimizing methods for data parallelism applications before the process of execution. The absence of this tool for data management increases network overhead and transaction costs that reduce the effectiveness of data portioning [6]. Hence, enhancing the performance of the data portioning algorithm is much needed. So, filling the gap among the partitioning schemes and volume of datasets, parallel schemes are executed in the clusters [7].

Text mining is a process to extract useful data and learn rate that eliminates noisy and disordered data from datasets [8]. Similarly, insignificant features create noise and they must be eliminated to minimize the size of data for producing improved clustering [9]. As it focuses only on design but fails to address the complexities that are generated in big data volume with complex data, issues are raised in these algorithms [10]. Thus, it resulted in a huge amount of storage of data in a relational database that made the performance of the system complicated. The main intention of this model is to improve the process of clustering big data with the help of the Bayesian clustering method. Optimizing this clustering approach is found to be helpful using a neural network scheme, and in the end, clustering is performed in an efficient way. Contributions of the proposed method are as follows.

- Develop an efficient data parallelism method for classifying emotions in social media comments or posts using neural networks.
- Initially a dataset is collected which have unwanted things like tag, stop word, upper case. The unwanted things are necessary to remove during the pre-processing period. Here six various models are used to pro-process the data, in order to improve the system performance.
- After that the pre-data is a cluster to label the class by the use of Bayesian finite mixture model clustering method. Then the features are extracted from the cluster data to recognize the emotions.
- Making use of the benefits of data parallelism to train the neural network and thus testing it for efficient prediction. Minimization of total time by partially splitting the data into subsets and feeding it to neural networks for training.

## Literature Review

Different big data clustering methods for classifying emotions have been presented; their limitation and techniques used have been explained in this section.

Mahmoodabadi et al. [11] have presented an epidemic model analyzed via Particle Swarm Optimization using dataset decomposition strategy. It was developed initially for standard Shepp-Logan phantom image reconstruction. It also had limitations over the limited angle of projections and results shown that this technique was not suitable for the reconstruction of acquired data and suitable only for standard medical imaging applications. Gupta et al. [12] have presented a compression framework called Dynamic Communication Thresholding for communication efficient hybrid training. As it incorporated an action technique to compress gradients, however, it fails to identify the most relevant neurons of the neural network for each training data. Ye et al. [13] have presented a clustering method based on the Bayesian Adversarial network and a low ranked model. It is then adopted to rank the cluster member estimation issue. However, it failed to reduce spectral clustering in the optimization procedure. Experimenting on different datasets showed that it required an optimal clustering unless it was unable to provide sufficient performance. Schneider et al. [14] have presented safe data parallelism for general streaming. However, it failed to ensure tuples as they always exit in parallel regions in similar order that affected the safety conditions.

Alguliyev et al. [15] have presented Semantic Driven Subtractive Clustering Method. In this method, clustering was done based on semantic strength, as opposed to the subtractive clustering method and Fuzzy C-means algorithm. Finally, the process of solving churn issues in big data context gets minimized due to the abundant implementation domain. Kinra et al. [16] have introduced a public policy decision-making algorithm to solve issues that were created by textual big data analysis. However, this algorithm failed to propose solutions for different propensity users having disclosed positive feelings. Bolla et al. [17] have presented a high-order Privacy Cost Optimization algorithm for big data clustering it in the objective function. Designing of this method depends on MapReduce for a very large dataset. However, this scheme failed to support larger actual datasets.

The above related models have some techniques to detect the emotion from the text data. Those models are not provide an effective outcome since they have some limitations like poor cluster performance, long execution time, failure to operate at long dataset, and failed to detect the emotions in various propensity. To overwhelm these limitations, an advanced cluster based machine learning technique is developed. The novel method offers an effective performance in any text data.

## Proposed Method for Emotion Recognition in Text Data

In this paper, a method for contribution in the field of classification of emotions is presented. Classification of emotion is done through a deep learning approach, and also it helps to identify certain emotions expressed in the means of verbal texts obtained from tweets. Two datasets consist of emotional tweets, and their intensity values are allowed for clustering, feature extraction and classification. Clustering was done through the Bayesian method, and feature extraction in terms of semantic and sentimental was done. Finally, the features were used for training Probabilistic Neural Network (PNN) and testing was performed. Testing results are classified as results of emotions obtained at the output layer of the neural network. The overall architecture of the proposed methodology is shown in Fig. 1.

As shown in Fig. 1, obtain social network emotion Datasets from a publicly accessible domain and acquired them for evaluation purposes. Pre-processing stage of eliminating unwanted terms was performed, and the dataset is permitted for clustering in order to obtain the label of data. Bayesian Finite Mixture Model is used to perform clustering, and this clustering was done with the help of Gaussian distribution and hence resulted in a group of clusters. The probability of each cluster is measured, and a label was found according to its probability. The training Phase of multi-layer probabilistic neural network occurs and it was done in parallel form. Learning was done through parallel subsets, and the input vector of each subset ($X$) is passed to the function of Gaussian for each class, and a group of hidden nodes for all Gaussian functions is computed at hidden nodes. All Gaussian functional values are grouped by each hidden node that is used to feed a single output node from that group and at each class output node. The sum of all inputs is multiplied by constant. The output nodes are determined by the maximum value of all summed functional values ($\Sigma$). Parallel execution of classification function was performed, and it is done through a trained Multi-layer Probabilistic Neural Network (MPNN) model. Probabilistic neural network then performs prediction at the testing phase, and at the final layer classification results have been obtained.

### Data Pre-Processing

Datasets are obtained and they consist of different emotions and its intensity levels. The pre-sampled dataset

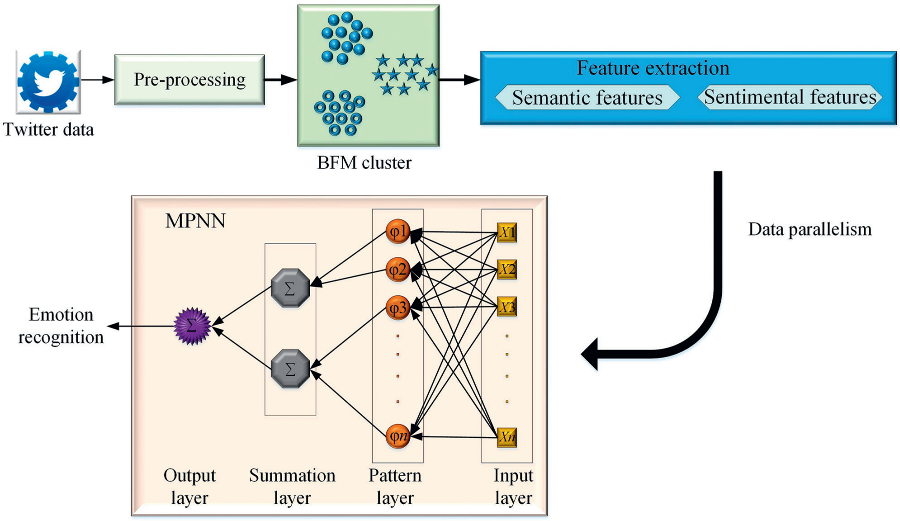

*Fig. 1*. Overall Architecture of proposed methodology

is shown in Table 1. As shown in Table 1, two datasets are obtained in parallel for classification and training. These datasets are obtained from the public access social network (e.g., *X*) dataset domain and used for evaluation purposes. These datasets are necessary to pre-process in order to improve the system performance. Here six various techniques were utilized to pre-process the raw data. There are lower conversion, Hypertext Markup Language or Extensible Markup Language (HTML/XML) tag removal, tokenization, stop word removal, normalization and numerical conversion.

Fig. 2 shows the architecture of data pre-processing. Initially the lower conversion methods are used to convert the upper case tweet words to lower case words. Then the data are passed to HTML/XML tag removal method to remove the tag presented in the dataset. Then the stop words are removed, afterward tokenization is done. Finally the data is normalized to a range, and conversion of the words into numbers for further process is done. That is, the string data conversion into numerical data depends on the unique count of the words. After pre-processing, the data is clustered to provide a label for classification.

**BFM model for clustering the pre-data**

In this Bayesian approach, probabilities of each class intensity value are analyzed, and its probability is calculated. Computing Probability values that belong to each cluster and Estimation-Maximization is performed with respect to the log-likelihood of each class [18]. Bayesian clustering is used to perform clustering in this proposed method. Let $D = \{x^{(1)}, \ldots, x^{(n)}\}$ which denotes numerical conversion dataset where $n$ represents the maximum number of input data. A cluster member number $K$ is selected and initialized to perform a hypothesis test. Group labels are initialized to corresponding $n$ observations. If $K$ component mixture derives the observation $i$, Bayesian distribution will be in the form of $p(y_i|\theta_k)$. Each element of $\theta = \{\theta_1, \theta_2, \ldots, \theta_k\}$ corresponds to a cluster. Additionally, this mixture has a weight or mixing probability is given as $P_r(c_i = k)$. As far now, the dataset is divided into subsets, and the number of clusters formation is denoted in the form of positive integer $n(p)$. In each cluster, a positive number $e_j$ is given, and a partition of $p$ number of things and device for this data clustering is defined as

$$\pi(y|p) = \prod_{j=1}^{n(p)} m(\{y_i, i \in C_j\}) = \prod_{j=1}^{n(p)} m(yC_j). \tag{1}$$

In equation (1), $m(yC_j)$ represents the joint distribution of responses for the item in the cluster $C_j$ as it protects an item index and cluster labels that require a function which does not depend on $j$ and its exchangeable function, $m$ represents von Mises distribution. $y_i = y_1$, $y_2$ and $y_1 = y$; the role of $y_2$ is to ensure that the $y_i$ to be spherical vector such that $||y_i||_2 = 1$. Here, the unlabelled datasets are required to perform clustering using covariate information between clusters. On the items, permuting pairs are jointly performed, and it can be given as shown in below

$$\begin{gathered} m(yC_j) = \int_u \prod_{i \in C_j} k(y_i|u_j) G_0 \mathrm{d}u_j, \\ \int_u \prod_{i \in C_j} k(y_i|u_j) G_0 \mathrm{d}u_j = \int_u \prod_{i \in C_j} k(y_i|u_j{:}x_i) G_0 \mathrm{d}u_j. \end{gathered} \tag{2}$$

In this equation (2), $k(y_i|u_j)$ is density function for $y_i$ with parameters and variables $u_j$. Distribution function $G_0$ of $u_j$ at space $U$. In Bayesian Formulation, an earlier probability that is allocated to each partition $p$, it gives an equation as shown in below

$$\pi(p|y)\alpha\Phi(p) = \pi(p) \prod_{j=1}^{n(p)} m(yC_j).$$

Where, ‘α’ is the Dirichlet prior parameter, $\Phi(p)$ represents parameters of a mixture model. After obtaining these output samples, partition data is essential to be left unaffected by label switching and continued till it reached the total number of clusters $K$. The members of the same cluster are likely to be of two data points when they frequently appear together. A pairwise posterior similarity is given, as shown below:

$$\pi_{ij} = p(c_i = c_j|y) = \frac{1}{M} \sum_{m=1}^{M} \{c_i^{(m)} = c_j^{(m)}\}.$$

In this equation, $M$ is the number of elements in the cluster, $c_i$ and $c_j$ are cluster assignments of observations $y_i$ and $y_j$ and thus, the true probabilities were left unknown and estimated using this equation. Here $\{c_i^{(m)} = c_j^{(m)}\} = 1$ if $c_i^{(m)} = c_j^{(m)} = 0$ otherwise. The clusters are combined in a step-by-step manner until the entire items are clustered together. The two clusters $C_{j1}$ and $C_{j2}$ are being maximized as shown below:

$$\frac{m(yC_{j1} \cup C_{j2})}{m(yC_{j1})m(yC_{j2})} \times \mu(e_{j1}, e_{j2}),$$

where

$$\mu(e_{j1}, e_{j2}) = \frac{\Gamma(e_{j1} + e_{j2} + \text{Ϫ})}{\Gamma(e_{j1} + \text{Ϫ})\Gamma(e_{j2} + \text{Ϫ})},$$

where Ϫ is variable which relies each data point, uses one of the mixture component likelihoods and raises the rest to the power zero, Γ is a parameter that affects the strength of the prior, and the cluster is generated by decentralizing the total sample data and by finding a cluster point.

To implement the spectral clustering method, every sample is generated from label $Z$. The Bayesian approach for finite mixture model clustering has been done and estimated for a high probability value. With the help of this probability values label for each group of clusters was determined as shown in Table 1.

**Feature Extraction**

In this proposed method, the feature extraction takes place after performing cluster operations. Both semantic features and sentimental features are extracted from the clustering results. Features are extracted automatically from the cluster data, as textual information is in natural

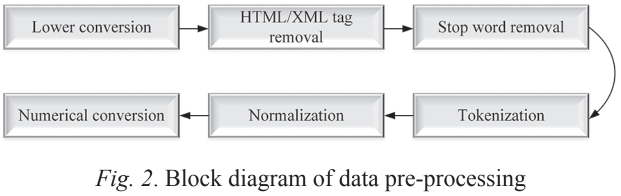

*Fig. 2.* Block diagram of data pre-processing

*Table 1.* Sample Datasets

| Samples | Labels |
|---|---|
| Wrath; umbrage offense; pique temper irritation; Lividity; irascibility short_temper spleen quick_temper | Anger |
| Unassertiveness; Trepidation; timidity; timidness timorousness; Suspense; stage_fright | Fear |
| worship adoration; Triumph; softheartedness tenderness; Rejoicing; pride | Happy |
| world-weariness Weltschmerz; woe woefulness; Weight; weepiness tearfulness; sorrow | Sad |

language format; a natural language processing technique is applied.

— ***Semantic Feature Extraction***

Semantic features refer to those semantically hidden concepts extracted from tweets. The semantic concepts of entities extracted from tweets can be used to measure the overall correlation of a group of entities. The semantic features are theoretical units which mean holding components that are used to represent the meaning of the word. In proposed model, the semantic features are extracted by providing a count of values in the clustering data, its length and then class it belongs to. It is represented as $(\Delta f_1, \ldots, \Delta f_m)$ features. For example, the entities "iPad", "iPod" and "Mac Book Pro" appeared more often in tweets of positive polarity, and they are all mapped to the semantic concept PRODUCT/APPLE. As a result, the tweet from the test set "Finally, I got my iPhone. What a product!" is more likely has a positive polarity because it contains the entity "iPhone" which is also mapped to the concept PRODUCT/APPLE.

— ***Sentimental Feature Extraction***

Sentimental features like Happy, Sorrow, Anger and Fear are extracted in various methods. By applying dictionary methods, synonyms and antonyms are found. To extract the sentimental features efficiently, Term Frequency-Inverse Document Frequency is used. The importance of a data in a cluster is measured, and its frequency determines the feature ratio. Term Frequency is computed by finding a fraction value between the number of times a data appears in the clustering and the total number of data. Likewise, Inverse Document Frequency is measured by measuring the number of input data with the same feature and its redundancy. It is a fraction between the total number of data and the total number of redundant data $t$, it is represented as $(\Delta f_m, \ldots, \Delta f_n)$.

**Parallelism based Multi-Layer-Probabilistic Neural Network for emotion recognition**

After extracting the features, the dataset was split into two groups such as training (80 %) and testing (20 %). Only the trained datasets are sent to train the model of MPNN. MPNN models are mostly used for classification types. A basic structure of MPNN is enhanced to multilayer and utilized for classification, as explained in the next section. Fig. 3 shows the MPNN architecture. Probabilistic neural network is represented by a parallel algorithm designed based on Probability density function estimation and Bayes classification rule, and as it is feed forward, a neural network developed from radial basis function network. It consists of four layers: pattern layer, input layer, an output layer and summation layer [19].

Extracted features are given as input to the input layer, PNN computes variations between sample input Eigenvector, i.e., input data and predicted Eigenvector that is classifier output. In here, the sample was considered as feature extracted data which was an matrix format, and the Eigenvector was considered as the s-variable of matrix. In the final layer, the classification of samples was computed to predict the output.

— ***Input Layer***

Input Layer is comprised of two groups of nodes that consist of semantic features and sentimental features. The feature vector is carried out by the input layer and delivers to the pattern layer. Input $\tilde{X}$ is made for the process of drilling the variables.

$$\tilde{X} = [x^T, b^T, v^T, a^T]^T,$$

where, $x$, $b$, $v$ and $a$ are input feature of both semantic and sentimental features. The input layer receives Eigenvector from samples and transfers the data to the pattern layer. In the input layer, the number of neurons presented is equal to Eigenvectors samples.

— ***Pattern Layer***

The second layer is the pattern layer, having $N$ neurons, here the number of training data is assumed as $N_i$ of the $i^{th}$ class, it takes a pattern $z$ from the input layer and assigns it to one of the $k$ classes. All smoothing parameters are almost similar to $\sigma_1 = \sigma_2 = \ldots = \sigma_d$ as well as the weight factor is estimated utilizing the bell-shaped Gaussian function.

$$\varphi_j^i(z) = \frac{1}{(2\pi)^{d/2}} \times \exp\left[-\frac{(z - x_j^i)^T(z - x_j^i)}{2\sigma^2}\right].$$

In the formula above, $(z - x_j^i)$ denoted as a function of $\varphi_j^i$, which indicates the prospect of $z$ pattern which is equal to $x_j^i$ and $j^{th}$ data training is represented as $x_j^i$ in the $j^{th}$ training vector belonging to the $i^{th}$ class. The smoothing parameter σ describes the spread of the Gaussian function and it can take a value among 0 and 1.

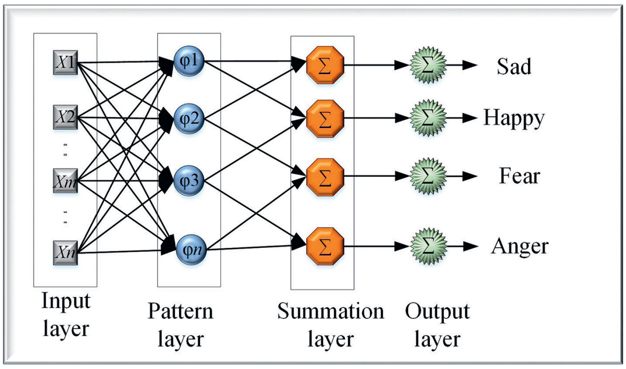

*Fig. 3.* MPNN architecture

— ***Summation Layer***

The third layer is the summation layer containing $k$ neurons which calculate the maximum likelihood of pattern $z$ belonging to the $i^{th}$ class, as given below.

$$P_i(z) = \frac{1}{N_i}\sum_{j=1}^{N_i} \varphi_j^i(z)\ ,$$

where $N$ is the dimension of the input vectors, φ is output of pattern layer and $z$ is input layer pattern.

— ***Output Layer***

The final layer is the decision layer that assigns to $z$ class with maximum likelihood as follows.

$$C(z) = \underset{i}{\operatorname{argmax}}\{P_i(z)\},\ i = 1, 2, \ldots, k,$$

where $C(z)$ is the class that belongs to $z$. Thus the feature vectors in each class may be reduced by thinning those that are too close to another one and making $f$ larger. The training set is made up of given exemplar feature vectors. For each one, it is known the class to which it belongs.

## Result and discussion

This section describes the evaluation of dataset details and compares it to previous techniques, namely Parallel K-Nearest Neighbouring techniques (PKNN) and Parallel Naïve Byes Model (PNBM). These methods are evaluated in MATLAB platform and implemented in Intel(R) Core (TM) i5-3570S CPU@ 3.10 GHz and memory 8 GB with a 64-bit operating system and 64 based processor.

**Dataset description**

Text can be annotated for emotions (such as happiness, fear, or surprise) or polarity orientation (positive/negative)[1]. While certain words have emotional significance in relation to a specific story, the affective potency of many others is a product of our collective imagination (e.g. words such as “mum”, “ghost”, “war”). Categorise the titles using the relevant emotion label and a valence indicator given, a set of predefined emotion labels (such as pleasure, fear, and surprise). WASSA-2017 Shared Task on Emotion Intensity is described in[2]. There are databases for four emotions: happiness, sorrow, fear, and rage. A real-valued score between 0 and 1 reflecting the level of anger felt by the speaker is included in the anger training dataset, for instance, along with tweets.

Further the proposed model performance is compared to some other existing models like PKNN and PNBM. The PKNN is one of the developed algorithms for classification. It can naturally handle multi-class cases and can be used for both classification and regression problems. The parallel implementation speeds up the algorithm to minimize the computational time. Similarly, PNBM is also another developed method for minimizing the computational time. It is known to scale linearly with the number of predictors and rows, also it handles categorization difficulties. For this reason, PKNN and PNBM are chosen for comparing the proposed model performance. Table 2 shows the Simulation parameter of proposed and existing model.

*Table 2.* Simulation parameter of proposed and exisiting models

| Methods | Parameter | Ranges |
|---|---|---|
| PNBM | var_smoothing | 0.00012 |
| PKNN | n_neighbors | 3 |
| | Standardize | 1 |
| MPNN | No. of hidden layers | 4 |
| | Training algorithm | Radial bias |
| | Scaling | Normalization |

In Fig. 4, the confusion matrix of proposed model is demonstrated. It is based on True positive, False positive, True negative, False negative. Using those values, we made a comparison for evaluation matrices, such as Sensitivity, Specificity, Accuracy, F-measure, Recall and Error analysis.

Accuracy values are determined and compared to the previous methods and presented in Table 3. From the table, it is understandable that the proposed MPNN classifier performs better than other previous classification techniques. MPNN model reached an accuracy level of 92.3 % value, and the PKNN model gives an accuracy value of 87 %, and the PNBM have accuracy value of 89 %. In contrast to accuracy measurement, error analysis also plays a vital role to show that a model is more effective. Then error values are analyzed for proposed as well as previous methods. From Table 3, it is visible that MPNN exhibits less error value in a range of 7.7 % and PKNN method exhibits 13 % of error values, and PNBM exhibits 11 % of error values.

By measuring Precision and recall values, the F-measure was then calculated. F-measure value evaluated from the proposed model is 90 %, PKNN is 82 %, and PNBM is 80 % of the measured value. Similarly, the recall measurement values of the proposed, as well as previous

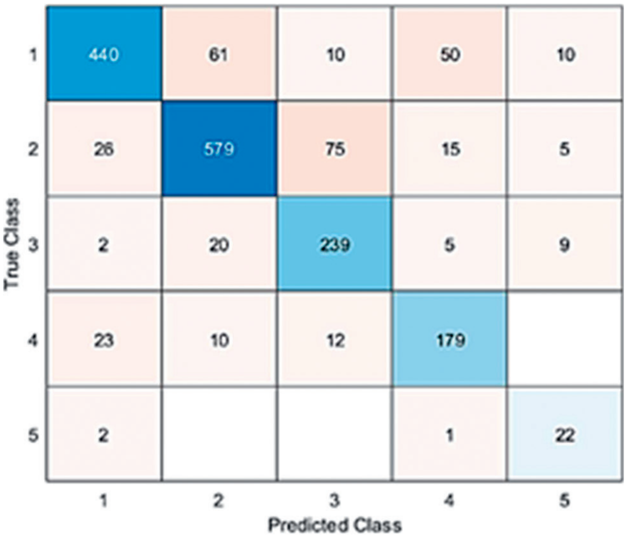

*Fig. 4.* Confusion matrix of proposed model

[1] Dataset 1: SemEval2007 affective text dataset. Available at: http://web.eecs.umich.edu/~mihalcea/affectivetext/ (accessed: 03.02.2023).

[2] Dataset 2: WASSA’17 Shared Task on Emotion intensity dataset: Available at: https://saifmohammad.com/WebPages/TweetEmotionIntensity-dataviz.html (accessed: 06.02.2023).

*Table 3.* Performance metrics comparison, %

| Metrics | MPNN | PKNN | PNBM |
|---|---|---|---|
| Accuracy | 92.3 | 87 | 89 |
| Error | 7.7 | 13 | 11 |
| F-measure | 90 | 82 | 80 |
| Recall | 90 | 82 | 80 |
| Sensitivity | 89 | 75 | 70 |
| Specificity | 90 | 80 | 85 |

models MPNN classifier, exhibits 90 % of recall values, PKNN gives 82 % of recall, and PNBM gives 80 % of the value. Then evaluating the results of sensitivity analysis of the proposed and existing methods is made. The proposed method MPNN classifier shows 89 % of sensitivity value and the PKNN model gives 75 % of sensitivity value. PNBM classifier exhibits 70 % of sensitivity value. Thus overall performance on sensitivity gets improved in the MPNN model. It is visible that the MPNN model gives 90 % of the specificity value, PKNN exhibits 80 % and PNBM about 85 % of the specificity value. Thus, from sensitivity and specificity analysis it has been clearly visible that the proposed model outperforms previous classification methods.

**Evaluation Results of MPNN classifier**

The proposed model is evaluated with respect to two datasets for parallel execution, and their comparison of results is presented here, as shown in Table 4.

From Fig. 5, it is quite desirable to say that the proposed model performs optimally during parallel execution time, and these techniques result in changes with respect to increase in data size. For 20 kB it results in 5 s and for 40 kB it exhibits 6 s of execution time. Comparatively, as without parallelization, a data size of 100 kB results in 42.5 s of execution time, but with parallelism, it minimizes up to 35 s. Thus, the proposed Bayesian clustering-based classification approach results better while executing parallel.

Table 5 shows the computational comparison of proposed and existing models after using parallelization. The proposed PNN take 42.5 s to complete the process for 100 kB, whereas the existing techniques like PKNN and PNBM consumed 47.9 s and 54.5 s respectively to complete 100 kB. The comparison proves the proposed model offers a better outcome as compared to other methods.

*Table 4.* Evaluation Results for Dataset 1 and 2, %

| Evaluation Metrics | Dataset 1 | | Dataset 2 | |
|---|---|---|---|---|
| | With Parallelism | Without Parallelism | With Parallelism | Without Parallelism |
| Accuracy | 92.08 | 88.14 | 94.53 | 90.91 |
| Error | 7.82 | 11.90 | 5.52 | 9.20 |
| F-measure | 90.33 | 87.21 | 91.07 | 80.83 |
| Recall | 91.90 | 85.09 | 92.07 | 79.83 |

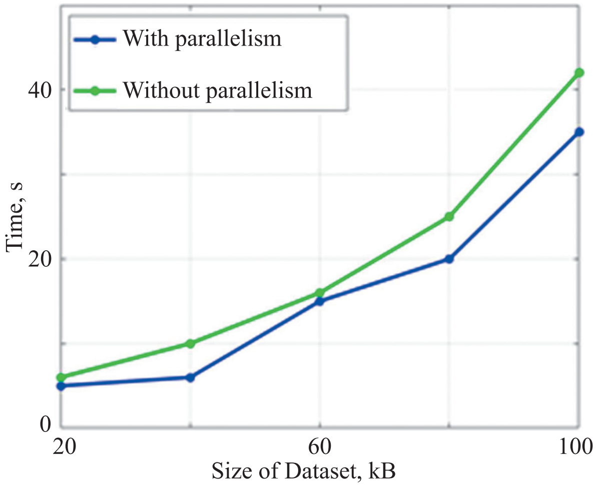

*Fig. 5.* Execution time comparison results for without parallelization and with parallelization

*Table 5.* Execution time comparison for 100 kB data size

| Methods | Time, s |
|---|---|
| Proposed PNN | 42.5 |
| PKNN | 47.9 |
| PNBM | 54.5 |

## Conclusion

In this paper, a novel cluster based supervised learning model is developed to predict the emotions from the tweet data. By reducing the complexity of annotation task, a parallelism based PNN model is developed to predict the suitable emotion. Two various datasets are used for analysis of the system performance individually such as annotated dataset and WASSA-2017 dataset. The dataset contain unwanted symbols or signs that are removed from pre-processing. Then the pre-data are clustered to provide a label through the use of Bayesian Finite mixture. Then the semantic and sentimental features are extracted from the cluster data to improve the system performance. These features are fed to the classifier to predict the emotions. A novel data parallelism is used in the classifier to reduce the computation time of the system. The proposed model provides 92 % and 94 % accuracy for annotated dataset and WASSA-2017 dataset, respectively, and the results are compared with previous techniques like PKNN and PNBM. It has even been evaluated that the proposed Bayesian Clustering-based MPNN classifier over-performs other models.

## References

1. Lee N., Ajanthan T., Torr P.H., Jaggi M. Understanding the effects of data parallelism and sparsity on neural network training. *arXiv*, 2021, arXiv:2003.11316. https://doi.org/10.48550/arXiv.2003.11316
2. Xun Y., Zhang J., Qin X., Zhao X. FiDoop-DP: Data partitioning in frequent itemset mining on hadoop clusters. *IEEE Transactions on Parallel and Distributed Systems*, 2017, vol. 28, no. 1, pp. 101–114. https://doi.org/10.1109/tpds.2016.2560176
3. Kulkarni M., Pingali K., Ramanarayanan G., Walter B., Bala K., Chew L.P. Optimistic parallelism benefits from data partitioning. *ACM SIGPLAN Notices*, 2008, vol. 43, no. 3, pp. 233–243. https://doi.org/10.1145/1353536.1346311
4. Hernández Á.B., Perez M.S., Gupta S., Muntés-Mulero V. Using machine learning to optimize parallelism in big data applications. *Future Generation Computer Systems*, 2018, vol. 86, pp. 1076–1092. https://doi.org/10.1016/j.future.2017.07.003
5. Karthick S. Semi supervised hierarchy forest clustering and KNN based metric learning technique for machine learning system. *Journal of Advanced Research in Dynamical and Control Systems*, 2017, vol. 9, pp. 2679–2690.
6. Chatterjee A., Gupta U., Chinnakotla M.K., Srikanth R., Galley M., Agrawal P. Understanding emotions in text using deep learning and big data. *Computers in Human Behavior*, 2019, vol. 93, pp. 309–317. https://doi.org/10.1016/j.chb.2018.12.029
7. Marimuthu M., Rajalakshmi M., Phil M.C.A.M. A big data clustering algorithm for sentiment analysis to search the crucial statistics for decision making. *International Journal for Research and Development in Technology (IJRDT)*, 2017, vol. 7, no. 2, pp. 132–138.
8. Feng N., Xu S., Liang Y., Liu K. A probabilistic process neural network and its application in ECG classification. *IEEE Access*, 2019, vol. 7, pp. 50431–50439. https://doi.org/10.1109/access.2019.2910880
9. He Q., Zhuang F., Li J., Shi Z. Parallel implementation of classification algorithms based on MapReduce. *Lecture Notes in Computer Science*, 2010, vol. 6401, pp. 655–662. https://doi.org/10.1007/978-3-642-16248-0_89
10. Tang D., Wei F., Yang N., Zhou M., Liu T., Qin B. Learning sentiment-specific word embedding for twitter sentiment classification. *Proc. of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2014, pp. 1555–1565. https://doi.org/10.3115/v1/p14-1146
11. Mahmoodabadi M.J. Epidemic model analyzed via particle swarm optimization based homotopy perturbation method. *Informatics in Medicine Unlocked*, 2020, vol. 18, pp. 100293. https://doi.org/10.1016/j.imu.2020.100293
12. Gupta V., Choudhary D., Tang P.T.P., Wei X., Wang X., Huang Y., Kejariwal A., Ramchandran K., Mahoney M.W. Training recommender systems at scale: Communication-efficient model and data parallelism. *arXiv*, 2020, arXiv:2010.08899. https://doi.org/10.48550/arXiv.2010.08899
13. Ye X., Zhao J., Chen Y., Guo L.J. Bayesian adversarial spectral clustering with unknown cluster number. *IEEE Transactions on Image Processing*, 2020, vol. 29, pp. 8506–8518. https://doi.org/10.1109/tip.2020.3016491
14. Schneider S., Hirzel M., Gedik B., Wu K.L. Safe data parallelism for general streaming. *IEEE Transactions on Computers*, 2015, vol. 64, no. 2, pp. 504–517. https://doi.org/10.1109/tc.2013.221
15. Alguliyev R.M., Aliguliyev R.M., Sukhostat L.V. Efficient algorithm for big data clustering on single machine. *CAAI Transactions on Intelligence Technology*, 2020, vol. 5, no. 1, pp. 9–14. https://doi.org/10.1049/trit.2019.0048
16. Kinra A., Beheshti-Kashi S., Buch R., Nielsen T.A.S., Pereira F. Examining the potential of textual big data analytics for public policy decision-making: A case study with driverless cars in Denmark. *Transport Policy*, 2020, vol. 98, pp. 68–78. https://doi.org/10.1016/j.tranpol.2020.05.026
17. Bolla S., Anandan R. Privacy preservation of data using efficient group cost optimization method with big data clustering. *International Journal of Advanced Research in Engineering and Technology (IJARET)*, 2020, vol. 11, no. 11, pp. 748–760. https://doi.org/10.34218/IJARET.11.11.2020.071
18. Fan W., Bouguila N. Spherical data clustering and feature selection through nonparametric Bayesian mixture models with von Mises distributions. *Engineering Applications of Artificial Intelligence*, 2020, vol. 94, pp. 103781. https://doi.org/10.1016/j.engappai.2020.103781

## Литература

1. Lee N., Ajanthan T., Torr P.H., Jaggi M. Understanding the effects of data parallelism and sparsity on neural network training // arXiv. 2021. arXiv:2003.11316. https://doi.org/10.48550/arXiv.2003.11316
2. Xun Y., Zhang J., Qin X., Zhao X. FiDoop-DP: Data partitioning in frequent itemset mining on hadoop clusters // IEEE Transactions on Parallel and Distributed Systems. 2017. V. 28. N 1. P. 101–114. https://doi.org/10.1109/tpds.2016.2560176
3. Kulkarni M., Pingali K., Ramanarayanan G., Walter B., Bala K., Chew L.P. Optimistic parallelism benefits from data partitioning // ACM SIGPLAN Notices. 2008. V. 43. N 3. P. 233–243. https://doi.org/10.1145/1353536.1346311
4. Hernández Á.B., Perez M.S., Gupta S., Muntés-Mulero V. Using machine learning to optimize parallelism in big data applications // Future Generation Computer Systems. 2018. V. 86. P. 1076–1092. https://doi.org/10.1016/j.future.2017.07.003
5. Karthick S. Semi supervised hierarchy forest clustering and KNN based metric learning technique for machine learning system // Journal of Advanced Research in Dynamical and Control Systems. 2017. V. 9. P. 2679–2690.
6. Chatterjee A., Gupta U., Chinnakotla M.K., Srikanth R., Galley M., Agrawal P. Understanding emotions in text using deep learning and big data // Computers in Human Behavior. 2019. V. 93. P. 309–317. https://doi.org/10.1016/j.chb.2018.12.029
7. Marimuthu M., Rajalakshmi M., Phil M.C.A.M. A big data clustering algorithm for sentiment analysis to search the crucial statistics for decision making // International Journal for Research and Development in Technology (IJRDT). 2017. V. 7. N 2. P. 132–138.
8. Feng N., Xu S., Liang Y., Liu K. A probabilistic process neural network and its application in ECG classification // IEEE Access. 2019. V. 7. P. 50431–50439. https://doi.org/10.1109/access.2019.2910880
9. He Q., Zhuang F., Li J., Shi Z. Parallel implementation of classification algorithms based on MapReduce // Lecture Notes in Computer Science. 2010. V. 6401. P. 655–662. https://doi.org/10.1007/978-3-642-16248-0_89
10. Tang D., Wei F., Yang N., Zhou M., Liu T., Qin B. Learning sentiment-specific word embedding for twitter sentiment classification // Proc. of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2014. P. 1555–1565. https://doi.org/10.3115/v1/p14-1146
11. Mahmoodabadi M.J. Epidemic model analyzed via particle swarm optimization based homotopy perturbation method // Informatics in Medicine Unlocked. 2020. V. 18. P. 100293. https://doi.org/10.1016/j.imu.2020.100293
12. Gupta V., Choudhary D., Tang P.T.P., Wei X., Wang X., Huang Y., Kejariwal A., Ramchandran K., Mahoney M.W. Training recommender systems at scale: Communication-efficient model and data parallelism // arXiv. 2020. arXiv:2010.08899. https://doi.org/10.48550/arXiv.2010.08899
13. Ye X., Zhao J., Chen Y., Guo L.J. Bayesian adversarial spectral clustering with unknown cluster number // IEEE Transactions on Image Processing. 2020. V. 29. P. 8506–8518. https://doi.org/10.1109/tip.2020.3016491
14. Schneider S., Hirzel M., Gedik B., Wu K.L. Safe data parallelism for general streaming // IEEE Transactions on Computers. 2015. V. 64. N 2. P. 504–517. https://doi.org/10.1109/tc.2013.221
15. Alguliyev R.M., Aliguliyev R.M., Sukhostat L.V. Efficient algorithm for big data clustering on single machine // CAAI Transactions on Intelligence Technology. 2020. V. 5. N 1. P. 9–14. https://doi.org/10.1049/trit.2019.0048
16. Kinra A., Beheshti-Kashi S., Buch R., Nielsen T.A.S., Pereira F. Examining the potential of textual big data analytics for public policy decision-making: A case study with driverless cars in Denmark // Transport Policy. 2020. V. 98. P. 68–78. https://doi.org/10.1016/j.tranpol.2020.05.026
17. Bolla S., Anandan R. Privacy preservation of data using efficient group cost optimization method with big data clustering // International Journal of Advanced Research in Engineering and Technology (IJARET). 2020. V. 11. N 11. P. 748–760. https://doi.org/10.34218/IJARET.11.11.2020.071
18. Fan W., Bouguila N. Spherical data clustering and feature selection through nonparametric Bayesian mixture models with von Mises distributions // Engineering Applications of Artificial Intelligence. 2020. V. 94. P. 103781. https://doi.org/10.1016/j.engappai.2020.103781

19. Alotaibi N., Al-onazi B.B., Nour M.K., Mohamed A., Motwakel A., Mohammed G.P., Yaseen I., Rizwanullah M. Political optimizer with probabilistic neural network-based Arabic comparative opinion mining. *Intelligent Automation & Soft Computing*, 2023, vol. 36, no. 3, pp. 3121–3137. https://doi.org/10.32604/iasc.2023.033915

## Authors

**S. Starlin Jini** — Researcher, S.T Hindu College, Nagercoil, 629002, India, sc 57214932690, https://orcid.org/0000-0002-8791-7481, starlinjini@gmail.com

**N. Chenthalir Indra** — Supervisor, Assistant Professor, S.T Hindu College, Nagercoil, 629002, India, sc 55803553600, https://orcid.org/0009-0001-9121-5386, chenthalirindra@gmail.com

*Received 07.07.2023*
*Approved after reviewing 27.10.2023*
*Accepted 23.11.2023*

19. Alotaibi N., Al-onazi B.B., Nour M.K., Mohamed A., Motwakel A., Mohammed G.P., Yaseen I., Rizwanullah M. Political optimizer with probabilistic neural network-based Arabic comparative opinion mining // Intelligent Automation & Soft Computing. 2023. V. 36. N 3. P. 3121–3137. https://doi.org/10.32604/iasc.2023.033915

## Авторы

**Старлин Джини С.** — исследователь, Индуистский колледж Южного Траванкора, Нагеркойл, 629002, Индия, sc 57214932690, https://orcid.org/0000-0002-8791-7481, starlinjini@gmail.com

**Ченталир Индра Н.** — руководитель, доцент, Индуистский колледж Южного Траванкора, Нагеркойл, 629002, Индия, sc 55803553600, https://orcid.org/0009-0001-9121-5386, chenthalirindra@gmail.com

*Статья поступила в редакцию 07.07.2023*
*Одобрена после рецензирования 27.10.2023*
*Принята к печати 23.11.2023*