RES MILITARIS
REVUE EUROPEENNE D ETUDES EUROPEAN JOURNAL OF MILITARY STUDIES
Social Science Journal

Recommendation System-Based Deep Learning Model for Scholarship Grant Eligibility Assessment from Crowdsourced Data with Traceability Focus

Priti Singh, Hari Om Sharan , C.S. Raghuvanshi
Faculty of Engineering and Technology, Rama University, Mandhana, Kanpur, Uttar Pradesh, India
preetirama05@gmail.com

Abstract

One of the rapidly advancing applications in information technology is the recommendation system, designed to navigate the vast sea of available data and aid in decision-making. While numerous recommendation systems exist for various domains such as food, movies, and other commodities, they often rely on graphs, filters, and AI techniques, resulting in alternative results that lack clarity. This study endeavors to develop a recommendation system focused on selecting the most appropriate beneficiary. The universal struggle of individuals for life's essentials is a pervasive reality worldwide. Governments bear the responsibility of ensuring that welfare programs reach the populace seamlessly, promptly, and fairly, although this entails significant challenges in data generation, storage, and utilization. This research employs a Recurrent Neural Network, a deep learning model, to analyze and predict outcomes using crowdsourced data obtained from diverse government sectors directly connected with the public. The experiment is conducted using Keras and Tensorflow Python libraries, and the results are verified and compared with other recommendation systems. The RNN model achieves an accuracy of 93.01%, precision of 98.2%, recall of 94.3%, and F1-Score of 95.8% in the experiment. The problem statement addressed in this paper represents a novel contribution, and the achieved experimental accuracy is deemed satisfactory.

Keywords: Deep Learning Algorithms, Recommendation System, Data Analytics, Distributed Data Analytics.

Introduction

The volume of Bigdata keeps increasing due to input from various online sources, and the gathered data is saved in the cloud. Internet usage has increased manifold in the last decade, corresponding to the number of internet users worldwide. It has resulted in crowding information on websites, social media, and smartphone apps. This humongous information is available under various forms such as userbase, opinions, comments, and other media form the fodder for crowdsourcing data. People working for private firms or freelancers make their data publicly available through various online media. Governments generally gather public data, store it, and process it to fulfill various requirements. This way, the Government gains complete control over citizens' data, which helps for better governance. Successful deliverance of Government schemes purported to benefit the people becomes easy for the Government. This paper focuses on predicting the people eligible for various Governmental welfare schemes. Identifying and classifying the recipients of said welfare schemes is a challenging task. Nowadays, the most common prediction process in vogue is based on sentiments or attributes. Regardless of the method or algorithm used for prediction, the elements' features play a crucial role in improving prediction accuracy, as mentioned earlier.

As engineering, the features are highly domain-dependent and tedious, and time-consuming. Only after repeated iterations do the predictive features emerge [1]. Even engineers who are domain experts need to incorporate the features from different data segments and learn and extract them by drawing a specific code.

Engineers of various domains have used several machine learning algorithms for integrating crowdsourced data with hybrid learning, and most of them have been satisfactory in the outcomes. Machine learning algorithms have been able to extract the feature spaces from the crowd data and help achieve unreachable statistical figures. The algorithms represent the statistical figures in the form of graphical images easily understandable to the human brain, as shown in Figure-1. For easy comprehension, the person faces in Figure-1 and Figure-3 are taken from internet sources. However, the contextual details in Figure-1 and Figure-3 are our contributions.

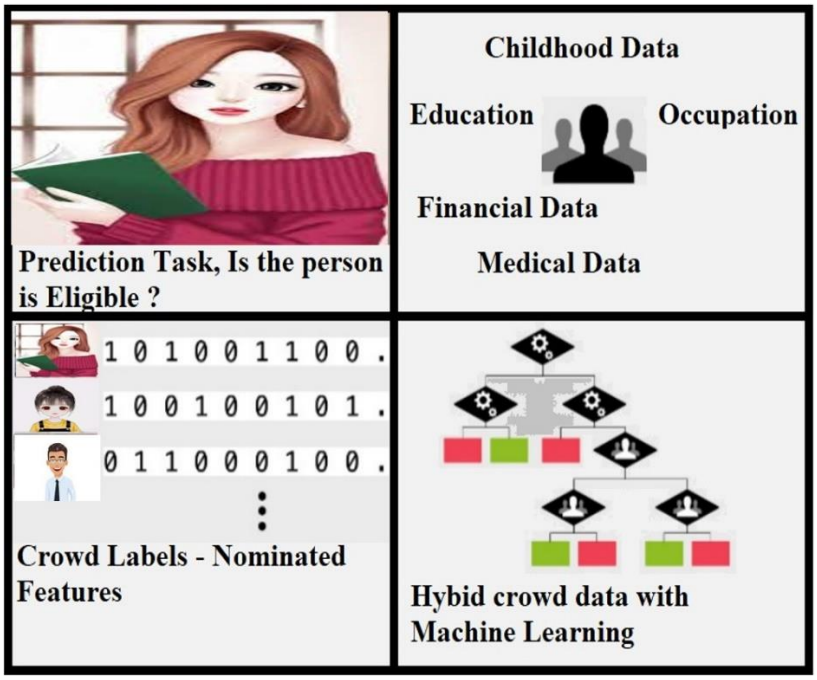


Figure-1. Integrating Crowd-Data with Machine Learning Algorithm

In doing so, a new prototype of a classifier is created for integrating machine learning into human performance in Figure 1. In this paper, a deep learning classifier is established to identify people below the poverty line, physically or mentally challenged, and eligible for Government grants or scholarships. The deep learning method performs assessing the features obtained from various crowdsources. Various existing works connect crowdsourcing and machine learning for optimal handling of crowd data [2-4]: the author suggested associating and integrating crowd features into classifiers to provide machine learning algorithms to be organized for new kinds of processes.

The prediction process is one data identification process that precisely extracts the important parameters and features from the big data. This paper has proposed a deep learning-based algorithm that processes the crowdsourced data containing the details of income, assets, medical history, and other factors that help to determine the eligibility of the recipients of various scholarships and Government assistance. Deep learning-based prediction helps establish the relationship between the data and its elements. It also helps in alluding to

important information and making assumptions from the available data. Cloud computing is one of the most powerful and efficient methods for solving critical data collection and productively merging human and machine computations [5-7]. Crowdsourcing is used for tasks that machines alone cannot perform, and human intervention is also called for [8]. The exclusive features are grouped, and the efficiency is illustrated by evaluating feature prediction processes. It includes all the features from all the data sources and various data types, such as images, text, and videos about people's activities and information. Instead of direct prediction, aggregating features from crowdsourced data is more accurate. Also, grouping the features integrated with machine learning algorithms (called a hybrid system) outperforms traditional methods in the data analytics industry. This paper reinforces the merits of a throughput synthesis. The classifiers designed through deep learning algorithms start functioning immediately on query requests and gradually train the deep learning algorithms for better data processing.

Contribution of the Paper

This work aims to create an efficient recommendation system for selecting eligible beneficiaries for Government scholarships. This work exploits a deep learning algorithm for analyzing the public data that selects the right people for receiving scholarships in times of need. The information is generated and integrated by various Government organizations. The primary process is, integrating the data using an index that helps to access the data from all the organizations concerned with the financial assistance. The proposed Recurrent Neural Network (RNN) architecture analyzes the data, focuses on the financial status of the person under consideration, and satisfies that the allocation is just and fair. Various steps are involved in this training and testing process, such as data generation, pre-processing, learning, and predicting data based on features using the RNN model. The implementation is done in Python with various important libraries that can perform neural network processes.

Literature Survey

This section provides detailed information about the earlier research that examined similar works for data analytics, prediction, and crowdsourcing. Earlier data analytics methods mainly used clustering algorithms to group the data by dividing the given dataset into a subset of items. Unsupervised learning algorithms have been a major part of clustering. The various data analytics applications involve customer data segmentation to bioinformatics [**9**]. When the input data increased massively, human computational methods were implemented, called intelligent human tasks [**10**]. A set of instructions is applied to the input data to instruct what to do over the data. The humane computation methods are performed by sending numerous ways to process bigdata and crowd data. Technological growth has accelerated the development of knowledge-based professions, societal paradigms, and business models in the past half-century. It was specifically valid for the academic sector, where it has changed in technologies, the use of the internet, and the introduction of various internet-based services like social networks. In social networks, the behavior and communication among teachers and students are varied [**11**].

Nowadays, the whole data analytics industry mainly focuses on crowdsourcing to solve various critical tasks without creating any training model or prior knowledge discovery. However, most of the tasks are not solved efficiently by amateurish crowd workers. For

example, not answering accurately the web searching for complicated queries [**12**] and condensing the overall queries [**13**]. Some of the researchers from [**12, 14-17**] proposed some methods of crowdsourcing where it decomposes the whole tasks into smaller chunks that can be solved speedily. Those critical unsolved problems need a cautious strategy and engineering of task-specific roadmaps. Deep learning algorithms can provide a better solution for acute and complex problems. Crowd sources involve dividing the tasks into sub-tasks and grouping and labeling them. The authors in [**18**] discussed hybrid classifiers, where crowdsources are integrated into machine learning algorithms and efficiently perform feature engineering and classifying. The author used Flock's interactive machine-learning environment, which focuses on feature analysis using machine learning with an interactive process.

Machine learning algorithms perform critical tasks iteratively and need focused input, meaning machine learning-based data analytics is a semi-automatic process. Machine learning algorithms can do generation [**19**], labelling [**20**], re-ranking [**21**], and feature selection [**22**]. However, the entire data analytics industry requires efficient feature exploration and labeling, where machine learning algorithms cannot provide efficiency to the expected standard. Artificial intelligence methods are widely applied for optimizing the effort of crowdsourcing. Some improve the aggregation process and worker quality AI techniques, such as Expectation Maximization [**23-25**]. Some of the other algorithms create behavioral models for predicting future performances [**26**] or estimating work appropriateness [**27**]. Thus, this paper aims to utilize automatons, deep learning, feature engineering, classification, and labeling. It incorporates crowdsources into deep learning algorithms for real-time applications.

The authors' primary objective [28] is to minimize the problems occurring in the core systems with the enhancement in the mobility of the nodes in the wireless networks with the implementation of Wireless Proxy Internet Protocol 6 (WPIPv6). In [29], a Genetic Algorithm with Hill climbing (GAHC) is proposed to find the optimal route in the multipath environment, and the optimal route is traversed by deciding the Cluster Head (CH). Chaotic Pigeon Inspired Optimization (CPIO) is employed for the feature selection process in bigdata classification [30]. This CPIO is implemented with an optimal Deep Belief Network (DBN). Additionally, the Harris hawks optimization (HHO)-based DBN model is deployed for class label allocation for the classifier. Dynamic data security can be utilized to avoid exploiting unauthorized users, as represented in [31]. This process is an evaluation mechanism for dynamic tracing and finding evidence of the data operation. The authors propose an adaptable Service Application Programming Interface (ASAPI) in [32] to provide interactive services to the Application Programming Interface (API) through a middleware framework in a Heterogeneous Environment (HE). The authors of [33] have suggested that some of the Metaheuristic algorithms, such as Particle Swarm Optimization (PSO), Genetic Algorithm (GA), and State Transition Algorithm (STA), are accurate models for classification processes.

In [34], the authors have proposed Continuous Wavelet Transform (CWT) to perform three essential processes: signal representation, feature extraction, and classification. To perform these processes, the Intelligent Industrial Fault Diagnosis using Sailfish Optimized Inception with Residual Network (IIFD-SOIR) Model has achieved a higher accuracy of

99.6% on the gearbox application. Fake hotel review recognition has experimented with Term Frequency-Inverse Document Frequency (TF-IDF) for detection and classification [35]. A compact band-pass filter is deployed to improve the band properties in wireless applications [36]. In [37], the authors have proposed a Fast Fuzzy C-Means-based Watershed Transform algorithm for feature extraction through clustering concepts. CDIO (Conceive, Design, Implementation, and Operation) methodology is proposed in [38] to address complex problems and obtain the best option for live-streaming during the Covid – 19 period. A low-cost VLSI architecture is designed for non-linear picture scaling. This scaling mechanism aids in the reduction of complexity and provides efficient memory utilization.

The author [39] used RNN algorithm to predict the diseases through the big data analytics. It mainly focussed on removing the noise in the data and tried to improve the quality of it. The authors adopted Rough Set Theory to select relevant features from the big data. The quality of the data was less compared to other reinforcement algorithms. RNN algorithms are suitable for real-time application that needs the processing of continuous data. RNNs widely used to build automation systems that provide the efficient and apt results. It is also implemented in smart grids where the energy consumption needs to be optimized. The author in [40] used RNN-LSTM algorithm in the smart cities that used smart grids for optimization of energy consumption. The model developed was fully automatic and was able to efficiently manage the continuous data provided by the smart grid. It was tested in real-time with 112 smart homes and the results evaluated over hourly basis provided better energy consumption management. [41] It was also compared with the other regression algorithms in tracking the information of the people. The author used several web-scraping and online data tools to collect the data of the people and processed them through RNN-LSTM algorithm. It also provided better results in the sentiments of the users and helped in better tracking of the users.

Limitations and Motivation

The literature surveys discussed above reveal that clustering, machine learning, and aggregation methods are used for analyzing large data. The methods discussed do not provide significant results in analyzing crowdsources, including knowledge discovery, labeling, and feature engineering. Some of the earlier research works have noted that feature engineering input with machine learning can provide aggregation and labeling that may not be suitable for all kinds of data. Thus, by analyzing deep learning and crowdsourcing efficiency, this paper has integrated both to create a hybrid model for crowdsourcing data analytics. This hybrid crowdsourcing with deep learning methods can efficiently do feature engineering, feature creation, feature aggregation, and labeling. This paper implements the best deep learning algorithm, RNN, to retain and process the previous output with the current input.

Though, the real-time problem of analysing crowdsource data has an universal implication, very few countries are good in managing them. This lengthy process that comprises of collecting, storing, analysing, computing, comparing and validating is not only time consuming, but also expensive. Research works until now have focussed on various methods for efficient management of the crowdsource data, but, with limited practical utility. The algorithm devised and proposed for application in this paper aims to bridge the gap by

providing with an automatic, cost-effective system of updating with minimum consumption of time.

Problem Statement

The RNN model explored for uncertain data has not been examined in earlier works. A portion of the network is observable in the proposed data model, and the remaining is unobservable. This paper investigates the applicability of state-of-the-art methods and offers a new architecture of deep learning algorithms for people selection. Consider an RNN model based on a graph $\boldsymbol{G}$ $(\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{X})$ called a network:

$$\boldsymbol{P} = \{\boldsymbol{P_1}, \boldsymbol{P_2}, \dots \boldsymbol{P_i}, \dots, \boldsymbol{P_n}\}, \forall \boldsymbol{i} = \boldsymbol{1\ to\ n}$$

indicates the set of nodes (people) record their data from various locations

$$\boldsymbol{L} = \{\boldsymbol{L_1}, \boldsymbol{L_2}, \dots \boldsymbol{L_i}, \dots, \boldsymbol{L_m}\}, \forall \boldsymbol{i} = \boldsymbol{1\ to\ m}$$

Each node ($\boldsymbol{P_i}$) represents an individual network element, and the graph $\boldsymbol{G}$ is a Government organization. Each $\boldsymbol{P}$ is interconnected using an index $\boldsymbol{\varepsilon}$.

$$\boldsymbol{\varepsilon} = \boldsymbol{P} \times \boldsymbol{P}$$

The possibilities of similar $\boldsymbol{P}$ obtained by a weight function $\boldsymbol{w}$are:

$$\boldsymbol{w} = \boldsymbol{\varepsilon} \to \{\boldsymbol{0}, \boldsymbol{1}\}$$

Where $\boldsymbol{w}(\boldsymbol{i}, \boldsymbol{j})$ is the similarity data $\boldsymbol{P_j}$ being obtained by $\boldsymbol{P_i}$. Sometimes, $\boldsymbol{w}$ may be in matrix form, but here $\boldsymbol{w}$ is defined as the matrix.

The neural network of RNN computes the weight matrix based on the feature similarity. Thus, the proposed RNN model performs the functions completely according to $\boldsymbol{w}(\boldsymbol{i}, \boldsymbol{j})$ and is called a weighted cascaded model. The seed set $\boldsymbol{P} \subset \boldsymbol{V}$ is activated initially. Each activated node in the network activates its neighbors based on the autonomous cascade model [18]. The activated node $\boldsymbol{i} \in \boldsymbol{V}$activates the neighbors $\boldsymbol{j}$ with the probability $\boldsymbol{w}(\boldsymbol{i}, \boldsymbol{j})$. In this model, we can obtain the weight matrix as,

$$\boldsymbol{w}(\boldsymbol{i}, \boldsymbol{j}) = \frac{\boldsymbol{1}}{|\{(\boldsymbol{k}, \boldsymbol{j}) \in \boldsymbol{\varepsilon}\}|},$$

where, $|\{(\boldsymbol{k}, \boldsymbol{j}) \in \boldsymbol{\varepsilon}\}|$ represents the degree of $\mathbf{j}$. Figure-2 illustrates the architectural form of crowdsourced data generation in the study. The data is generated from various government organizations, schools, and healthcare industries. The information thus collected is pre-processed each time, and the attributes are used for selecting the process. The selected scholarship grants will be assigned to the grantees at various locations on various levels. Predicting the right people for scholarship grants is obtained by the proposed RNN algorithm.

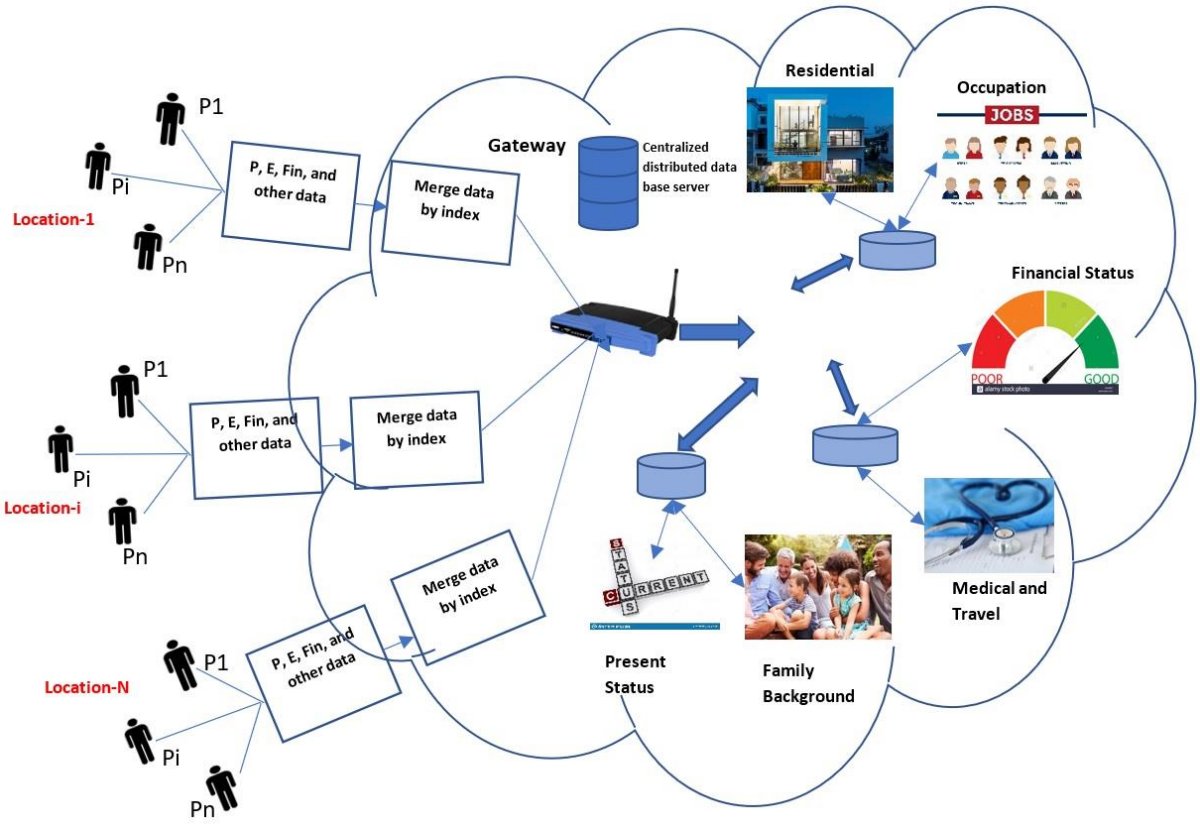


Figure-2. The architecture of the Proposed Model

$\boldsymbol{P_i} \subset \boldsymbol{P}$// personal data

$\boldsymbol{R_i} \subset \boldsymbol{R}$// residential data

$\boldsymbol{E_i} \subset \boldsymbol{E}$// educational data

$\boldsymbol{Fin_i} \subset \boldsymbol{Fin}$// financial data

Crowd-Source Data Generation

Data generation and reprocessing are shown in Figure-3(a) to Figure-3(f). It is assumed that Data $\boldsymbol{D}$ is generated at various stages in a person's life. Some data may change according to age and living style, and others may not.

$$\boldsymbol{D}=\{\boldsymbol{D_1}, \boldsymbol{D_2}, \ldots, \boldsymbol{D_i}, \ldots, \boldsymbol{D_k}\}, \forall \boldsymbol{i = 1\ to\ k}$$

People $\boldsymbol{P_i}$ from a different location, enters their personal, parental, and location information ($\boldsymbol{L_i}$ / $\boldsymbol{R_i}$) when they were born. Data $\boldsymbol{D_1}$ is written as

$$\boldsymbol{D_1} = \{\boldsymbol{P}, \boldsymbol{R}, \boldsymbol{Fin}\}$$

Other than $\boldsymbol{D_1}$, the other people's data $\boldsymbol{D_i}$ are written as

$$\boldsymbol{D_i} = \{\boldsymbol{P_i}, \boldsymbol{R_i}, \boldsymbol{E_i}, \boldsymbol{Fin_i}\}, \boldsymbol{i} \in \boldsymbol{n}$$

$\boldsymbol{D_i} \subset \boldsymbol{D}$// data

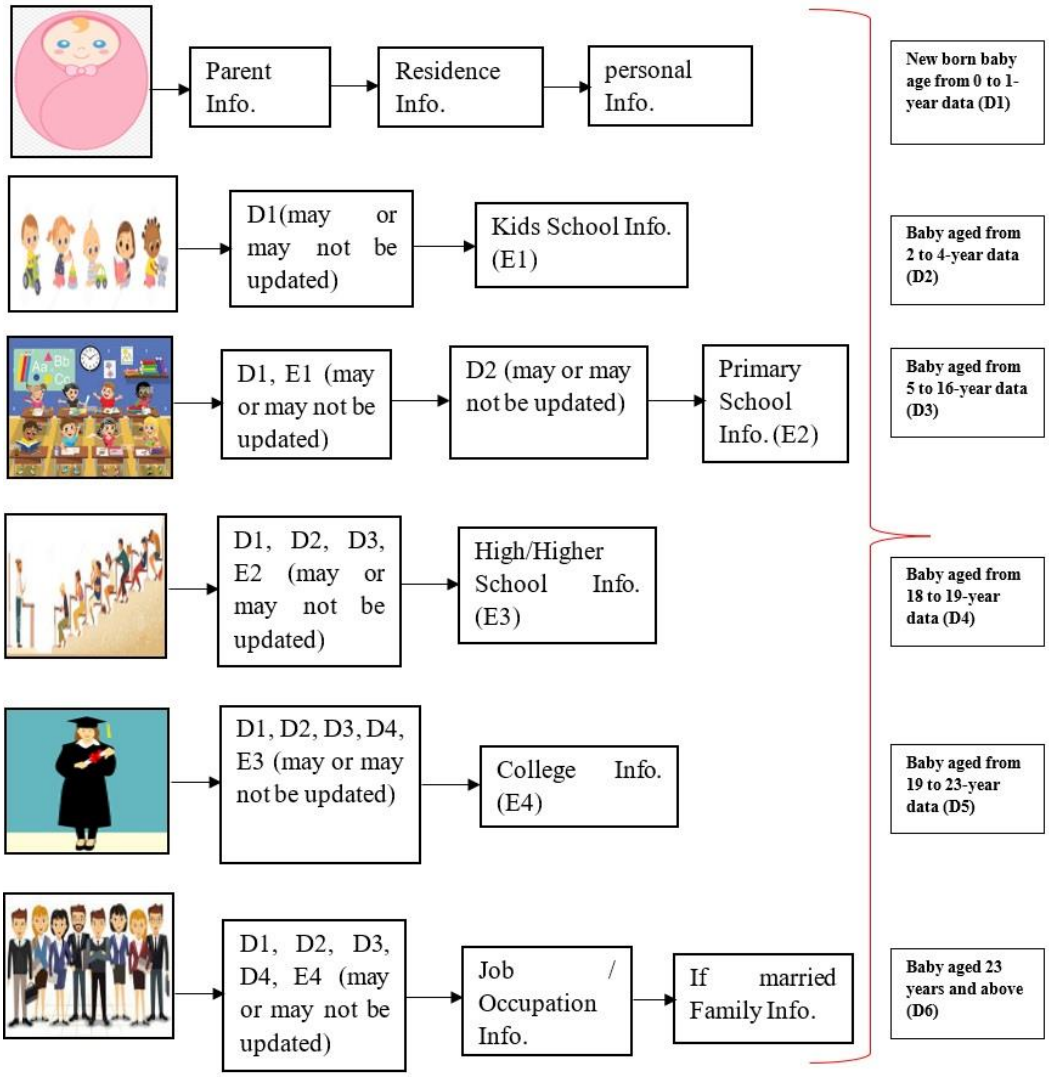


Figure-3. Data Generation Model

Figure-3 illustrates the data $\boldsymbol{D_i}$ generated at various intervals in a person's life. It includes parent, residence, personal, and occupation/job. The educational data is represented as $\boldsymbol{E_i}$. After each interval of time is completed, the data $\boldsymbol{D_i}$ and $\boldsymbol{E_i}$ are updated and given as:

$$\boldsymbol{D_1 \leftarrow D_2 \leftarrow D_3 \leftarrow D_4 \leftarrow D_5 \leftarrow D_6}$$

$$\boldsymbol{E_1 \leftarrow E_2 \leftarrow E_3 \leftarrow E_4}$$

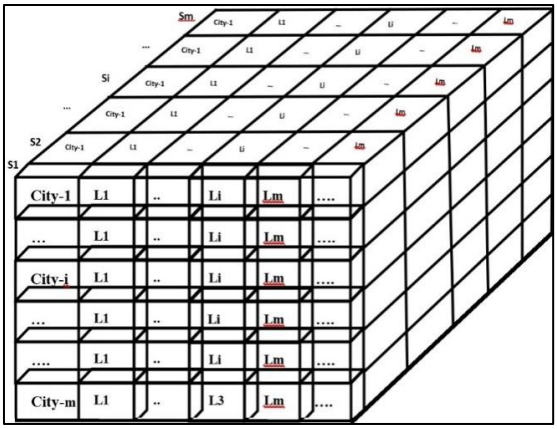


Figure-4. Multi-dimensional Data Generation

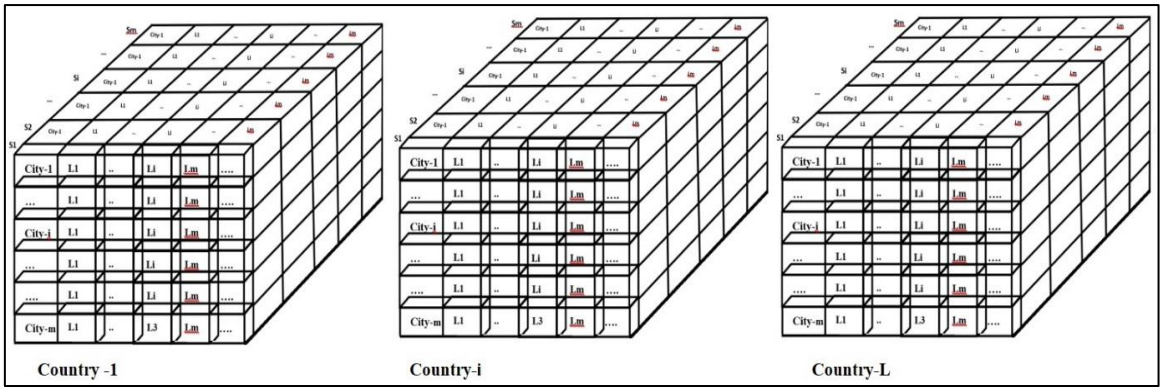


Figure-5. Crowd-sourced Data Generation

People are selected for scholarships based on their financial status and medical report. The entire dataset generated in crowdsourcing is multi-dimensional, represented in Figure-4 for one country. The multi-dimensional data generation is extended for multiple countries, making the data crowd-sourced-data. Figure-5 shows the crowdsourced data , which includes various countries' data. The data collected from each person $\boldsymbol{P}$, living in location $\boldsymbol{L}$, city $\boldsymbol{Cy_i}$, state $\boldsymbol{S_i}$, and country $\boldsymbol{C_i}$ is represented in the matrix form as $\boldsymbol{n \times k \times m \times r}$ where $\boldsymbol{n}$ denotes the number of people, $\boldsymbol{k}$ the number of locations, $\boldsymbol{m}$ the number of cities, $\boldsymbol{r}$ the number of states, and $\boldsymbol{L}$ the number of countries. The crowdsourced data is also considered as time-series data since it is generated continuously. Among $\boldsymbol{n}$ number of people, the set of all eligible people $\boldsymbol{E_p}$ ($\boldsymbol{E_p \subseteq P}$) are selected by the RNN algorithm by analyzing the whole dataset. The crowdsourced dataset is generally multi-dimensional in nature, where data elements are enormous.

Hence, it takes two initial processes, pre-processing and normalization. It can represent any multi-dimensional data in a cubic format, but retrieving a data element from the cubic representation requires a complex and nested query. OLAP provides more options to create cube views for cubic data. Cube views comprise dimensions, parents, hierarchy, levels, attributes, and measures. Compared with the earlier neural network models discussed in the literature, this model is partly observable. Notably, it is assumed that the selected $\boldsymbol{P}$ can observe only a portion of the network represented as

$$\boldsymbol{G' = (V', \varepsilon', w')}.$$

It is known that

$$\boldsymbol{v' \subseteq V, \varepsilon' \subseteq \varepsilon \subseteq (\subseteq v' \times v')}.$$

also,

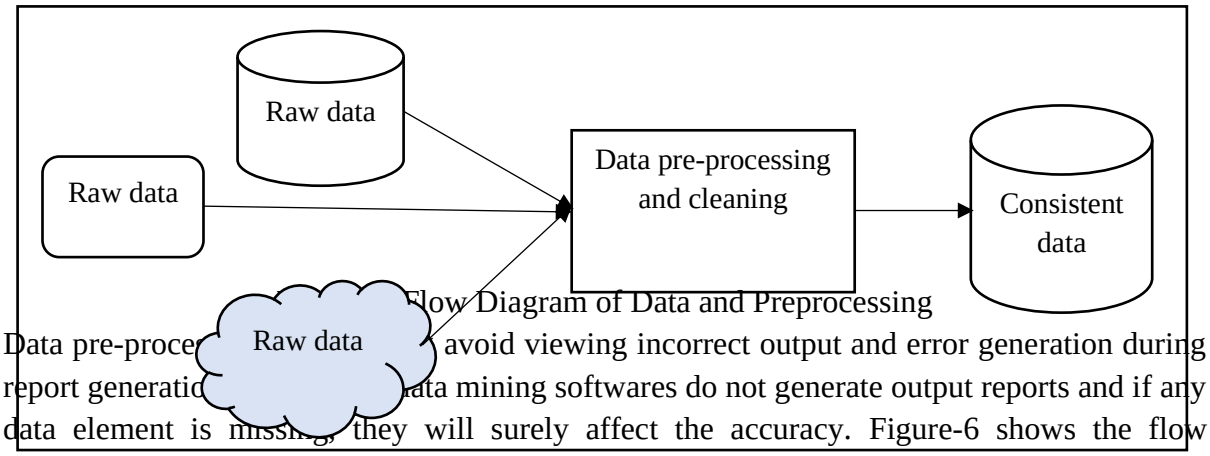
$$\boldsymbol{w' : \varepsilon' \rightarrow \{0, 1\}}.$$

From this, it is identified that the range of $\boldsymbol{w'}$ involves the weight values of $\boldsymbol{\varepsilon'}$. The activation and spreading process over neighbor nodes is done using a threshold value. The threshold value indentifies the right people to be selected as the output. Threshold values might be the features of the features [minimum, maximum], determining the people's eligibility for scholarship grants.

Pre-processing the data

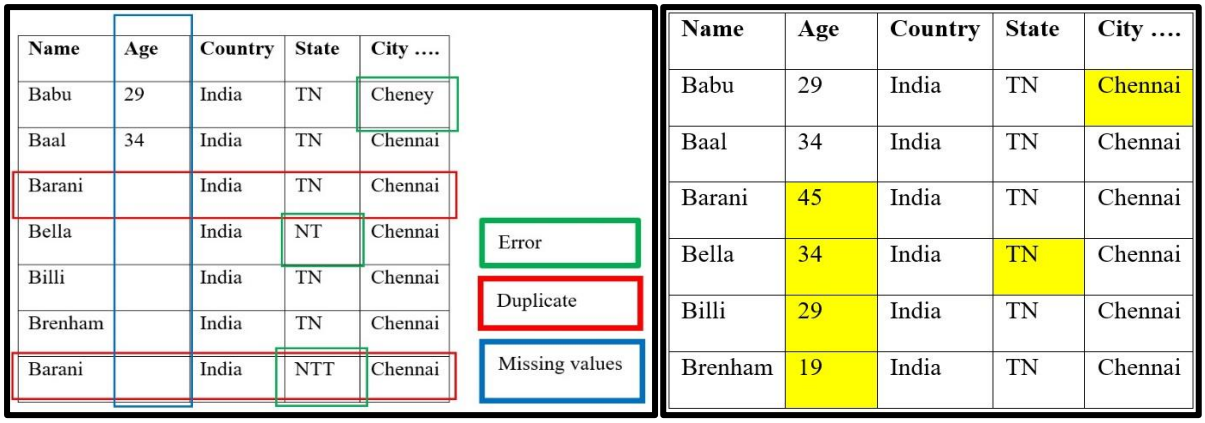
Pre-processing is one of the main processes in data mining, which transforms the real-time raw data into a human-machine understandable format. Real data is always incomplete, has errors, and is inconsistent. Pre-processing is a proven technique that resolves the above-

said issues and prepares the real-time data for further processing—various applications like CRM and rule-based applications, i.e., neural networks. Bigdata analytics is used for massive data in terms of variety. In the data pipeline pre-processing task is considered time-consuming. Error in the data includes errors in data collection, measurement errors, human errors, noise in the data, etc. Pre-processing includes cleaning the data, reducing the data size, transforming and integrating data, exploring the data, and making the data visualized in the table, graph, and figures. Data normalization includes formatting the data in a standard way and converting the data elements into a common unit.



Flow Diagram of Data and Preprocessing

Data pre-proce avoid viewing incorrect output and error generation during report generatio ata mining softwares do not generate output reports and if any data element is missing, they will surely affect the accuracy. Figure-6 shows the flow diagram of the pre-processing task. The output of the pre-processing step provides consistent data and is ready for data processing. Every data element is given in a single column. Each distinct record is arranged in a single row. Various data elements are linked by indexing to work on them—Figure-7 shows clarity of the data prior to and after pre-processing stage. A small portion of the data is taken from the experimental dataset, and pre-processing is applied. The pre-processing task identifies the missing values in the age feature, error (typo) in state and city, and duplicate data in row-3 and row-7. Most of the automatic pre-processing methods fill the missing values with 1 or 0 and mark the duplicate records and error data. Those records will not be considered during the mining process. But this work is particularly designed for public data analysis, 28% of the data needs manual corrections. Manual corrections in the pre-processing task increase the accuracy of the analytics and mining process, which is verified in the experiment. Pre-processing identifies the age value which is not given for all the records from the 3rd to the 7th row, typo errors occur in city name (1st record), state name (4th and 7th records), and data in the 3rd is repeated in row number 7. This kind of error in the data produces compilation and execution errors during the data analytic process. Thus, it is essential to pre-process the data and further increase the quality of data analysis and mining. Figure-7(b) shows the corrected consistent data ready for the data process.



Name	Age	Country	State	City ….
Babu	29	India	TN	Cheney
Baal	34	India	TN	Chennai
Barani		India	TN	Chennai
Bella		India	NT	Chennai
Billi		India	TN	Chennai
Brenham		India	TN	Chennai
Barani		India	NTT	Chennai

Name	Age	Country	State	City ….
Babu	29	India	TN	Chennai
Baal	34	India	TN	Chennai
Barani	45	India	TN	Chennai
Bella	34	India	TN	Chennai
Billi	29	India	TN	Chennai
Brenham	19	India	TN	Chennai

Figure-7. Pre-processing, (a). Identifying Errors, duplicates, and Missing Values, (b). After Correction

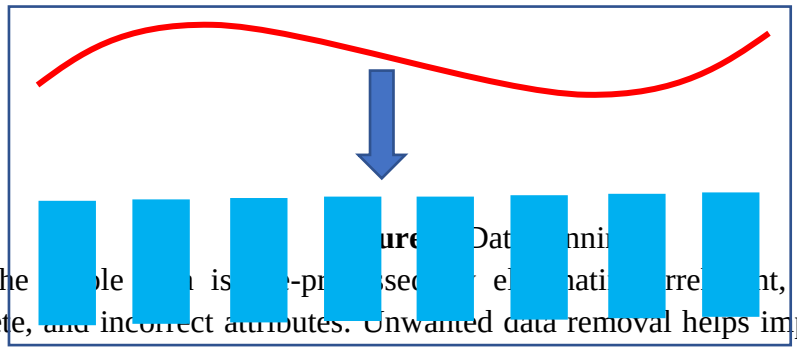
Binning the Data

The Binning process transforms the numerical elements in the dataset into different categorial counterparts. For example, age and financial status are the data elements that are discretized according to the frequency of the data. The Binning process increases the accuracy of the prediction model by removing and eliminating the error, noise, size, and non-linearity. Finally, the binning process detects the outliers, missing values, and invalid numerical data. This process initially divides the data range into $\boldsymbol{M}$ number of equal sizes based on the width and interval boundaries given below.

$$\boldsymbol{width} = (\boldsymbol{maximum} - \boldsymbol{minimum})/\boldsymbol{M}$$

$$\boldsymbol{minimum} + \boldsymbol{width}, \boldsymbol{minimum} + \mathbf{2} \times \boldsymbol{width}, \ldots, \boldsymbol{minimum} + (\boldsymbol{M} - \mathbf{1}) \times \boldsymbol{width}$$

Where the minimum and maximum are the lowest and highest values of the data features, after binning, the entire data quality increases and the process is shown in Figure-8.



ure Dat nni

The ple is e-pr sse el nati rre nt, overlapped, repeated, incomplete, and incorrect attributes. Unwanted data removal helps improve data quality and add attributes to the data to enhance the data representation.

Data Pipeline

The input data flow focused on a single streamline is called a pipeline. It reduces computational complexity by integrating the various flow of data, processes, hardware, and other frameworks and manages the data flow. Data generated or collected from multiple applications, humans, or devices must be processed before utilization. The complicated process can be avoided by providing a single data flow. The single data flow helps to monitor the movement of entire data in terms of place, time, and method. The pipeline process includes integrating the data, analyzing the data quality, standardizing the data, avoiding anonymization, and sharing the data with multiple systems through a single channel.

The pipelined data is fed into RNN architecture as input. Each row in the data matrix is fed into RNN, which learns and classifies the data based on the essential features by comparing their range of values. If the feature values lie within the valid range, RNN selects the data. RNN repeats the process of validating the data regarding various feature values. Because of validating the input dataset repeatedly, RNN is used in this paper.

Data Integration

A unique index key ($\boldsymbol{uky_i}$) is generated for integrating multiple data generated for each individual. For each $\boldsymbol{P_i}$, data $\boldsymbol{D_i}$ is generated from various sources at various stages of their life and is integrated using $\boldsymbol{uky_i}$ and they are created by using the location information $\boldsymbol{L}$. These are a person's location information with a five-digit number. The number of digits may increase according to the population. For example, a person living in Chennai, India, is assigned a unique key, "313310303302058", which concatenates the code values assigned for Continent, country, state, district, city/village, area, street, door number, and person number (see Figure-9(a)). The general form of the unique key is shown in Figure-9(b).

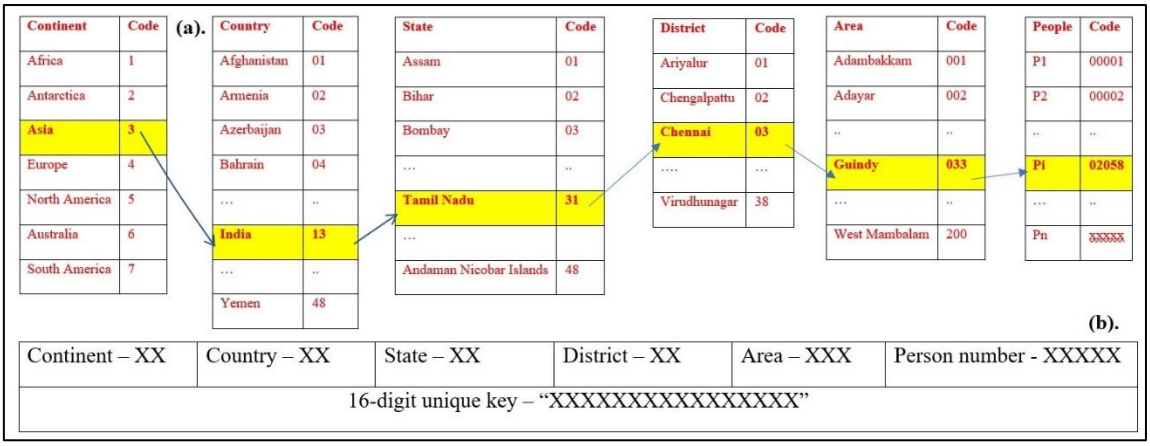


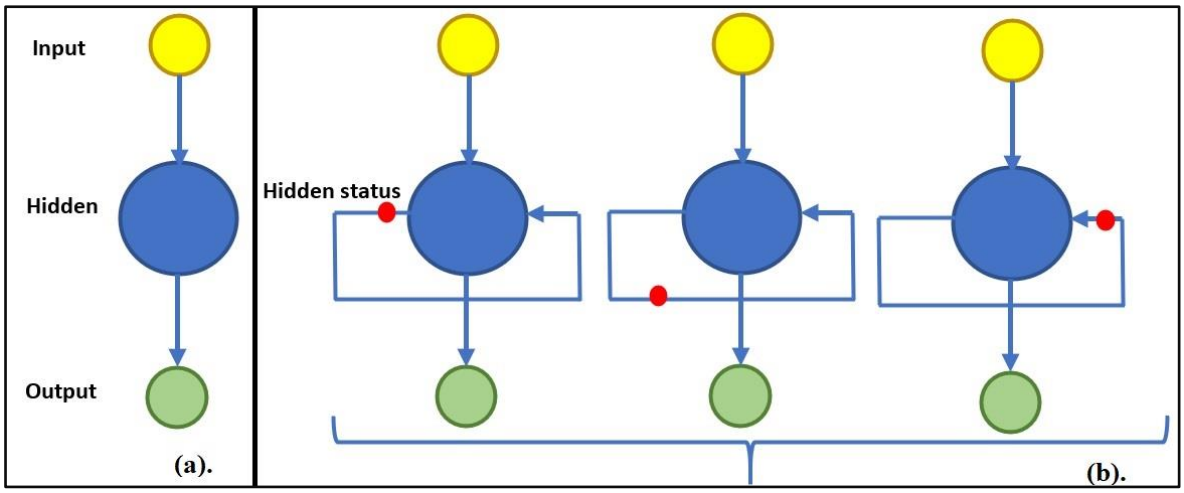
Figure-9. Unique Key Generation for Data Integration

Recurrent Neural Network Model

This work constructs a Recurrent Neural Network architecture for predicting the right people for scholarship grants. The right person is identified based on finance, other assets, and people's density (**FL, Ass, PD**). These are essential features for choosing the right people from the dataset. The actual scholarships provided are based on predictions made by the proposed RNN model that considers the following performance metrics (**FL, Ass, PD**) as an index for measuring success. Scholarship grant selection in common is crucial to predict because of the high dimensional data space. The (**FL, Ass, and PD**) are the most significant metrics used globally to evaluate an individual's financial standing. Thus, it is considered inevitable for developers to study machine learning algorithms and neural networks in massive time-series data analysis.

Recurrent Neural Network (RNN) is a powerful algorithm in recent real-time applications like speech recognition, image recognition, stock prediction, and language translation. RNN is good at modeling the sequences of data. RNN is the extended version of the FFNN in three different layers: input, hidden, and output. The number of hidden layers is not limited. It varies according to the volume of features to be learned. FFNN uses the previous information with the present data for comparison. By adding a loop that can repeatedly perform the FFNN process, it becomes RNN (see Figure-10). It provides a

significant path for information flow from the previous step to the next step. The hidden state includes information representing the previous inputs. Also, comparing simple or smaller features makes it impossible to suggest that a person is eligible. At the same time, it is easy to come to a better conclusion by analyzing many records and data. The sequence of data about a person is generated from various sources and many forms. Birth, education, occupation, life, and financial status are different forms of data fed into RNN.



(a). Feed Forward Neural Network (b). Recurrent Neural Network

Figure.10. RNN from FFNN

RNN algorithms are usually characterized by gradient vanish or explosion, as experienced in various works. However, in this paper, such issues are minimal because the individual's information is updated at various stages with real-time data that never affect the gradient. The reason for using RNN here is that the data processing and flow direction with the feedback system is achieved automatically.

In RNN, obtaining exact results is possible only at the end as accuracy increases with iteration, so it is initially challenging to predict as the pattern requires training. Updating the gradient is a sequential process, and additions to the data needs added memory. It makes it easier for the algorithm to recognize the pattern. In short, the algorithm keeps receiving, comparing, and updating the data until the end of the process. The data generated from multiple sources $\{\boldsymbol{D_1}$to $\boldsymbol{D_m}\}$ is fed as a sequence of data into RNN. In the first level, the data $\boldsymbol{D_1}$ is fed into the first level of RNN. The kid's level data $\boldsymbol{D_2}$ is fed at the second level of RNN. An appropriated output is generated from each level of RNN and sent back to the previous level. It activates the backpropagation and FFNN process, that can compare the current value with the previous value to fetch the next value, represented as $\{\boldsymbol{P_i(D_1)} \rightarrow \boldsymbol{l_1}, \dots, \boldsymbol{P_i(D_i)} \rightarrow \boldsymbol{l_i}, \dots, \boldsymbol{P_i(D_m)} \rightarrow \boldsymbol{l_m}\}$. Comparing the outputs of each element in the data sequence makes a better prediction since it has enough information to make the decision-making.

The data sequence $\boldsymbol{D_i}$ has different features with different values. The feature value might change based on time. For example, the value of "Age" and "education" increases yearly. Based on education and occupation, the financial status of the people varies. The medical reports of the people also change with age and time. Hence, the RNN is used here to process and learn the data sequence generated at different stages in an individual's life and

fed as input. The recipients are selected based on financial status, assets, and family. Analyzing and processing the dataset can help obtain the details of the people's income, age, and physique. The income of a person is found from $\boldsymbol{D_j}$, and any change in them gets updated in the income column. The person may or may not be affected by any diseases. It can be found from the values in the "medical report" generated at $\boldsymbol{D_k}$. Thus, all the data sequences need to be analyzed to predict people eligible for a scholarship. Crowdsource data analysis using RNN is illustrated in Figure-11.

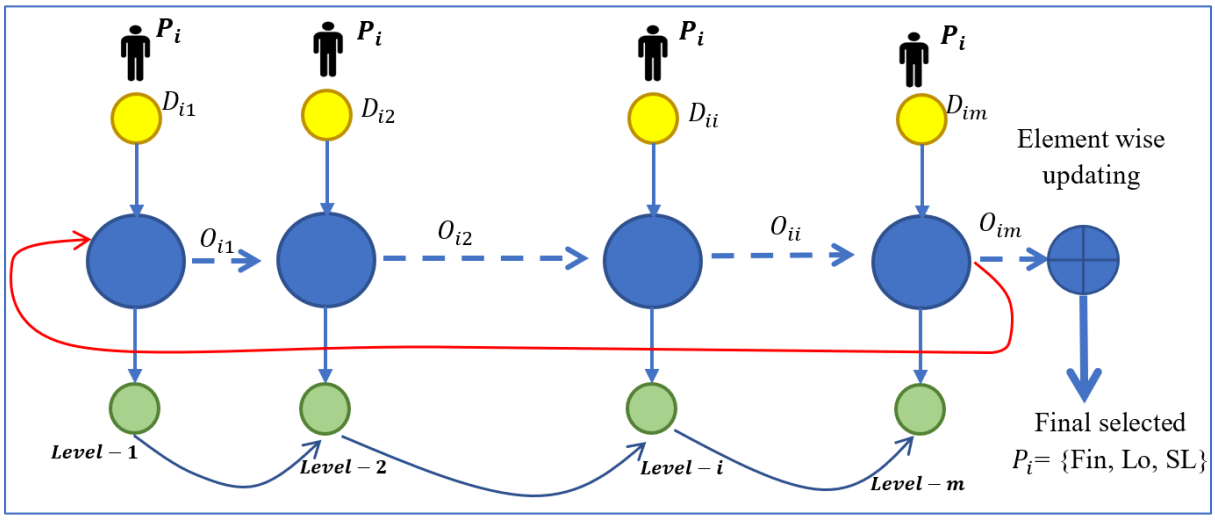


Figure.11. Proposed RNN For People Prediction

The various stages of data collection for every individual are denoted as $\{\boldsymbol{Level_1}, \boldsymbol{Level_2}, \ldots, \boldsymbol{Level_i}, \ldots, \boldsymbol{Level_1}\}$. Initially, the data $\boldsymbol{D_{i1}}$ feeds to the hidden state where the RNN algorithm updates the fields of an individual. After updating, the output $\boldsymbol{O_{i1}}$ is passed to the next hidden state, which is in $\boldsymbol{Level_2}$of RNN, where the RNN encodes $\boldsymbol{D_{i1}}$ with $\boldsymbol{D_{i2}}$ which is already present in the hidden state of $\boldsymbol{Level_2}$ and delivers an output ($\boldsymbol{O_{i2}}$). In $\boldsymbol{Level_2}$ the RNN algorithm processes both the data $\boldsymbol{D_{i1}}$ and $\boldsymbol{D_{i2}}$ and updates the fields with the existing data. It is then sent to the next level as output $\boldsymbol{O_{i2}}$. In the next step, the data $\boldsymbol{O_{i2}}$ feeds to the hidden state of $\boldsymbol{Level_3}$which already consists of data fed by $\boldsymbol{D_{i3}}$. At each level, the RNN carries out the same process of updating the data until we reach the final step or the end of the data sequence at $\boldsymbol{D_{im}}$. The final level of the RNN provides the encoded information obtained from all the data sequences. The final output feeds into the FF layer to classify the data. The FFNN layer provides the predicted data according to the user query. The data sequence $\boldsymbol{D_{ii}}$ is obtained for $\boldsymbol{P_i}$ based on the unique index key, avoiding collision or a mismatch during analytics (see Figure-11).

While considering the architecture of RNN, the TensorFlow unit of RNN is called a cell representing the whole layer. Each TensorFlow is connected in recurrent form following the "Feeding-To-Itself" method. A one-roll of RNN cells provides an RNN layer that makes multiple-roll based on the "Number-of-Steps" input fed by the user. From any set of data (sequence of tokens), RNN fetches the input as a sequence of words and provides a set of words in vector form as output. It is well known that RNNs can model short-term dependencies using the hidden state. Because, RNN can retain the data/information from timestep-1 to timestep-2 using the RNN units. The timestep of the current hidden state is calculated from the previous information's hidden state and the timestep of the current input.

Also, a sample timestep allocation is given below to allocate the minimum timestep for maximum iteration.

Input: the total income of Dinakaran is below two lakhs

Timestep:

"the"	–	d(0)d(1),
"total"	–	d(1)d(2),
"income"	–	d(2)d(3),
of "Dinakaran"	–	d(3)d(4),
is "below"	–	d(4)d(5)
"two-lakhs"	–	d(5)d(6)

The process of RNN following FFNN is

$$I = \sigma(Wi \times D) \text{ // } \sigma \text{ is the activation function}$$
$$Y = SoftNMax(Wi \times I)$$
$$I1 = \sigma(Wi1 \times D)$$
$$I2 = \sigma(Wi2 \times I1)$$
$$Y = SoftNMax(Wi \times I2)$$

It says that the output Y of the NN is obtained by processing the inputs D and hidden information I using the weight W. The values of $Wi1$ and $Wi2$ are different. Using this NN process, the RNN is mathematically expressed as:

At each time stamp, **t**:

$$It = \sigma(U \times Dt + W \times It - 1) \text{ // Input}$$
$$Yt = SoftMax(V \times It) \text{ // output}$$
$$J^t(\theta) = -\sum_{j=1}^{|M|} Y_{t,j} \log \overline{Y_{t,J}} \text{ // cost value}$$
$$J(\theta) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{|M|} Y_{t,j} \log \overline{Y_{t,J}} \text{ // cross entropy-based loss value}$$

The set of vocabularies is denoted as M, and the cost function is $J(\theta)$. From the above discussion, the RNN processes are written as a sequence of steps (Figure-12):

1. Compute I_{t-1} from weight vectors U and D
2. Compute Y_{t-1} from I_{t-1} and weight vector V
3. Compute I_{t-1} from U, D, W, and I_{t-1}
4. Compute Y_t from V, and I_t , and so on.

The output generation with respect to the input values fed to RNN is illustrated in Figure-12.

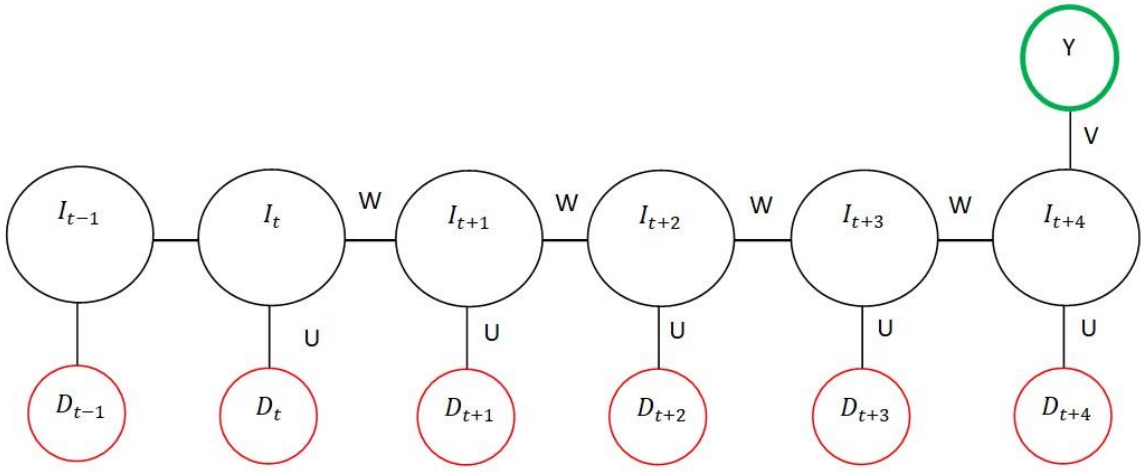


Figure-12. Input- RNN – Output

The pseudocode of RNN is given below, which can be implemented in any programming language, and efficiency is verified.

Pseudocode

{

Step-1:*Load the entire dataset, pre − process, and make the data ready for process.*

Step-2: *Divide the data into two portions as training and testing data.*

Step-3: *Convert the data* $\boldsymbol{D}$ *into integer sequences // tokenizing*

Step-4: *Label each data sequence in the training process* // create a unique index for each data sequence

Step-5:*Build RNN for analysing each token*

Step-6:*Load the trained data, train network for predicting the test data sequences.*

Step-7:*Obtain a new data − abstract collection input to FFNN a seed sequence*

Step-8: *Repeat step* 4 *to* 7 *using trained data labels.*

Step-9:*Change the architecture of the RNN by changing the number hidden layers and validate the performance*

Step-10: *Change the dataset and verify the performance.*

}

The main objective of this paper is to design a new RNN model for predicting the user-required data pattern with increased accuracy and decreased error. The overall logical functions of RNN are given in the form of pseudocode. It can be coded in any computer programming language, implemented, experimented on various datasets, verified, and evaluated the performance. The labelled features are selected based on maximum, minimum, or within a range of values from the overall dataset. For example, very low income and assets maximized poverty scores and depended on age groups. This selection procedure is programmed in the RNN model and is represented in the following manner.

A function $\boldsymbol{f}$ is defined as $\boldsymbol{f}$: $\boldsymbol{A} \rightarrow \boldsymbol{R}$ from dataset $\boldsymbol{D}$

Sought an element $\boldsymbol{D_0} \in \boldsymbol{D}\ \exists, \boldsymbol{f}(\boldsymbol{D_0}) \leq \boldsymbol{f}(\boldsymbol{D}), \forall\ \boldsymbol{D_i} \in \boldsymbol{A}$ // minimum data

(or)

Sought an element $\boldsymbol{D_0} \in \boldsymbol{D}\ \exists, \boldsymbol{f}(\boldsymbol{D_0}) \geq \boldsymbol{f}(\boldsymbol{D}), \forall\ \boldsymbol{D_i} \in \boldsymbol{A}$ // maximum data

Several real-time applications are validated based on the following equation:

$$\boldsymbol{f}(\boldsymbol{D_0}) \geq \boldsymbol{f}(\boldsymbol{D}) \leftrightarrow \bar{\boldsymbol{f}}(\boldsymbol{D_0}) \leq \bar{\boldsymbol{f}}(\boldsymbol{D})$$

including

$$\bar{\boldsymbol{f}}(\boldsymbol{D}) \coloneqq -\boldsymbol{f}(\boldsymbol{D}), \bar{\boldsymbol{f}} : \boldsymbol{A} \rightarrow \boldsymbol{R}$$

Is more suitable to get minimum or maximum values of the same classes. The above-explained proposed model is experimented with to verify the performance.

Experimental Results and Discussion

The proposed RNN architecture is implemented in Python, and the coding is compiled and executed in the Intel Pentium Core i7 processor with 16GB RAM and a 500GB hard disk. The experiment is conducted with three large-scale data collected from different cities (Chennai, Vellore, Madurai) in India. The dataset size is 23, 45, 847, 85,83,346, and 78,92,484, illustrating the people's personal, occupational, financial, medical, and other

necessary information. The values of the features are highly different, including bank balances, assets, and medical reports. The RNN algorithm implementation has ten main steps, as given above. Initially, all three datasets are combined, loaded, pre-processed, and ready for data analytics. Then the dataset is divided into two portions in the ratio of 80:20 for training and testing. Each part converts the data into integer sequences on each part. Data analytics is carried out from the training dataset piece by amount.

Do tokenizing. Label each data sequence as "nil", "below poor", "poor", "lower middle" or "normal", "middle", "upper middle", "first-class", "super-first class", and "rich" based on the income, current bank balance, assets, a smaller number of people in the family. Other features are also considered for choosing the right people for the scholarship grants. Building RNN using various layers, such as a convolutional layer for learning and extracting data features, fully connected layers for analyzing the weights among the hidden layers, SoftMax layers for classification, and an output layer. Train the RNN model using the trained dataset and do the testing process over the test dataset. Apply feed-forward on the output obtained from the hidden output. Finally, backpropagation is applied for error computation. According to the error, the training and testing process is repeated to reduce the error and improve prediction quality. The prediction percentage in terms of sensitivity, specificity, and accuracy are calculated for a different set of data and comparing the performance of the proposed RNN. The dataset used in the experiment is generated in real-time resources.

Dataset Used in the Experiment

In contrast with other bigdata or crowdsourced data generation, these people's data is not available publicly. In this work, the entire dataset is generated and collected from various sources, such as Government hospitals (13%), primary health centers (37%), schools (14%), colleges (7%), health & family welfare organization (17%), and universities (2%). New-born baby data are recorded in government hospitals, primary health centers, and health & family welfare organizations. The remaining data is collected from schools, colleges, and universities. If needed, the data can be collected from various private and Government financial sectors.

Implementation of RNN

First, the network and hidden state layers are initialized. The hidden state's dimension and shape should reflect the shape and measurement of the RNN. Then the input sequences are passed through the loop, including the invisible state, to the RNN. After encoding each data and hidden state, RNN provides an output with a modified hidden state. Repeat the loop until data reaches Dm and finally pass the output to the FF layer and deliver a prediction. A for loop is assigned to do the forward pass process in RNN. The performance of the RNN can be calculated by training the NN. A forward pass creates a prediction, comparing the prediction with the ground-truth values using $\boldsymbol{LF}$ (Loss Function), which gives an error value that shows the network's poor performance, followed by three steps in the training process. Finally, the error value is used for backpropagation, which estimates the gradients of all the nodes in the network. The following equation shows how to calculate the error.

$$\boldsymbol{E = loss(prediction, ground - truth)}$$

The gradient value calculated in each layer utilized to adjust or modify the network's weight $\boldsymbol{w(i,j)}$ helps to learn the network. If the gradient value is high, the adjusting network is also significant.

Table.1. Data Aggregation in Terms of Age

Age Range	No. of. People (%)
0-3	4.4
3 to 5	3.5
5 to 15	6.3
15 to 18	5.5
18 to 23	6.7
23 to 30	17.6
30 to 45	15.7
45 to 55	15
55 to 65	17.1
65 and above	8.2

The feature aggregation result is obtained initially to evaluate the proposed method's efficiency from the experiment. Some of the essential features aggregated are age, education, gender, salary, and assets are highly necessary to select the person for granting scholarships. Though awarding the scholarship is not restricted to age, it is essential to understand that the scholarship amount and type may change. Table-1 shows the number of people living in a particular location in the age groups from 0 to above 65.

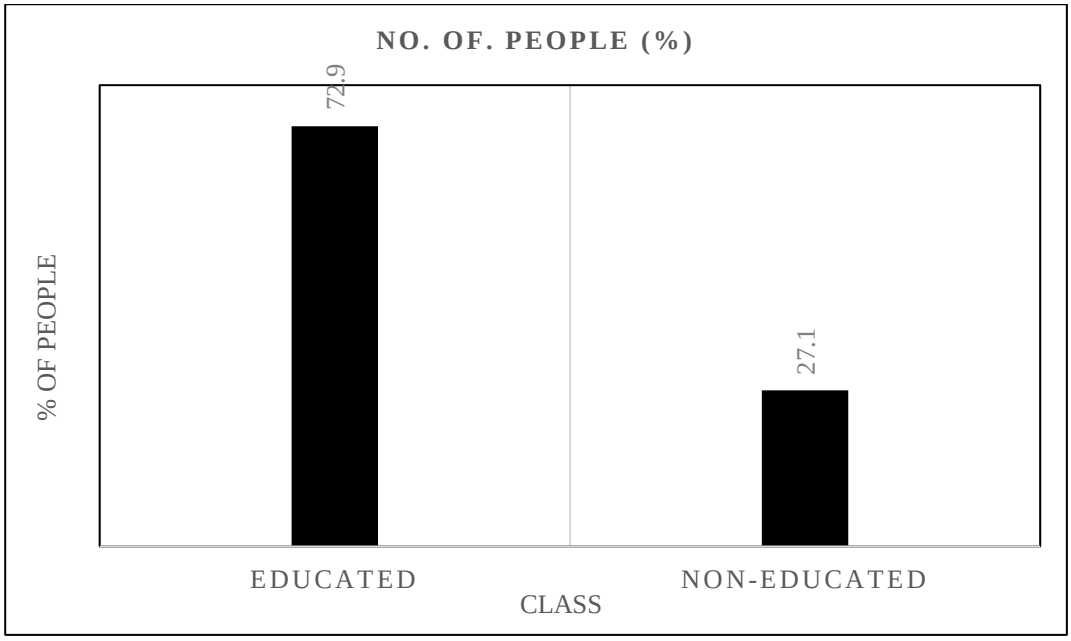


Figure-13. Data Aggregation in Terms of Education

Educated people can get a high pack of scholarships based on their present or further educational requirements. It doesn't mean that all educated people require a scholarship. 72.9% of people are identified as educated from the total number of people, and others are not educated. Also, it is obtained from the experiment and shown in Figure-13. The range of scholarships gets changed even for gender variation. So, the feature engineering process is

applied to feature space, and gender-based data aggregation is obtained. The experimental result in Figure-14 shows that 53% of the people are female, and the remaining 47% are male.

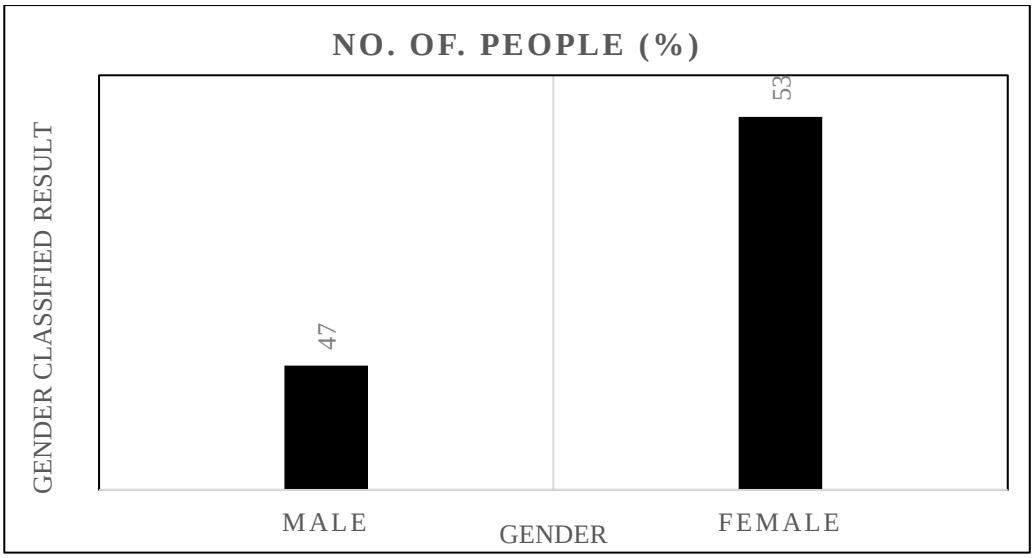


Figure-14. Data Aggregation in Terms of Gender

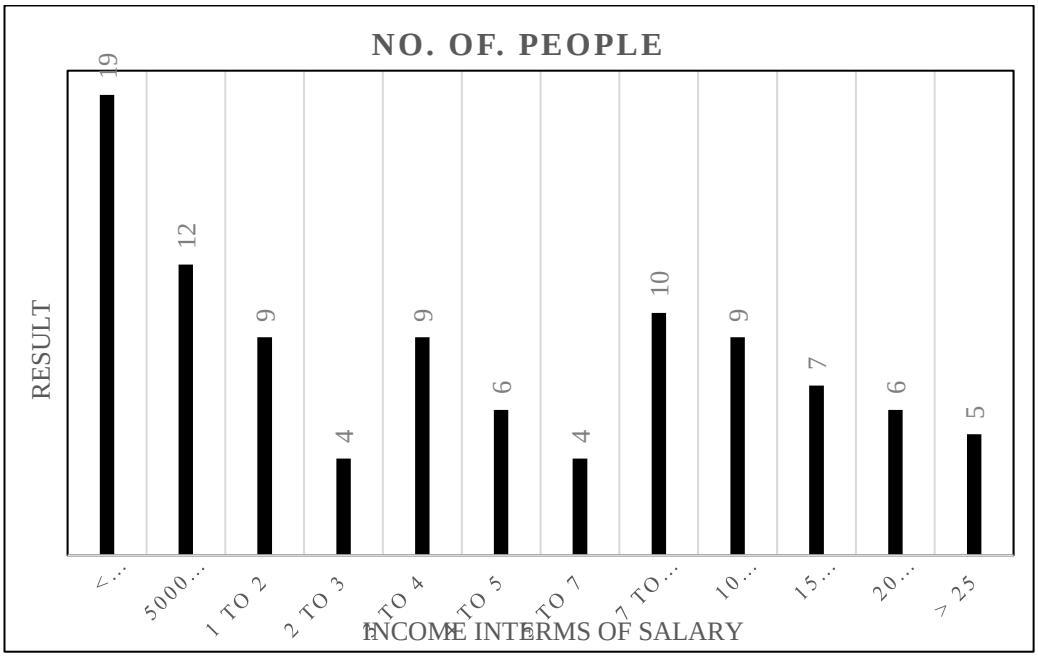


Figure-15. Data Aggregation in Terms of Salary

The income of the people positively influences granting of the scholarship. It looks for people with significantly less income who cannot run the family. Thus, income-based data aggregation is carried out, and the relevant result is shown in Figure-15. People's income is decided by occupation. If the salary is high, but the work is low, it is the most severe need to watch the person, whether he is doing any illegal activities against the government rules. The data aggregation in terms of occupation is shown in Figure-16. Of the total, 77% of the people are employed, and others are not used.

Figure-16. Data Aggregation in Terms of Occupation

People may have more money and are rich if they have their ancestral properties or receive income from their assets. People who may have more assets cannot get any scholarships.

Figure-17. Asset Value Vs. Number of People

Thus, data aggregation in terms of assets is shown in Figure-17. It shows that people below the poverty line (i.e., 59%) need financial aid from the Government during disaster periods or regularly. After data aggregation, the RNN model classifies and predicts the data based on the user queries.

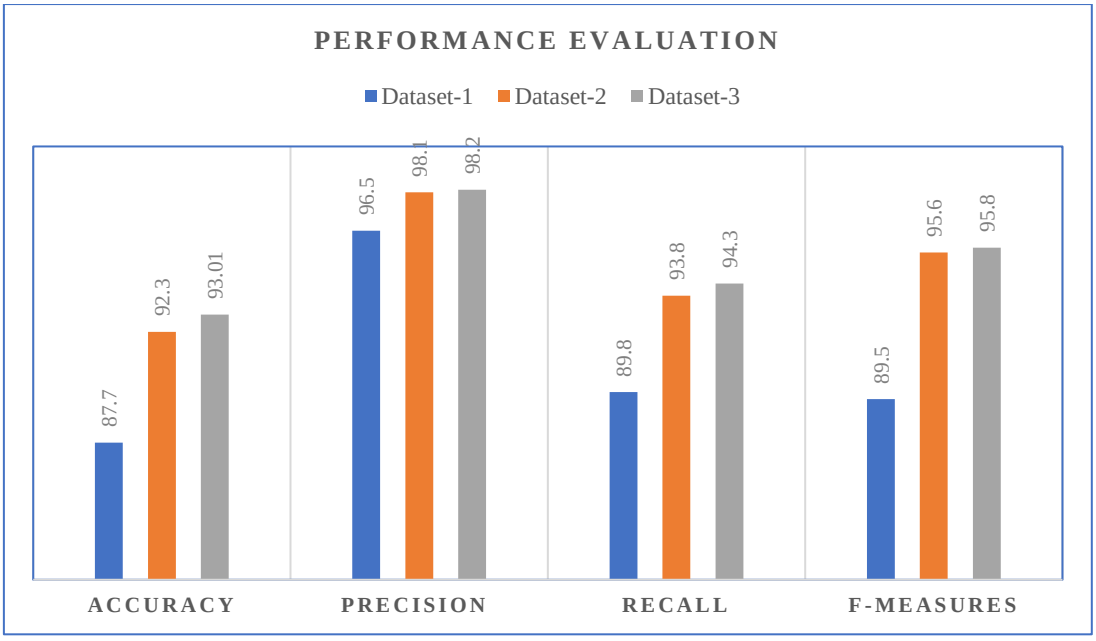


Figure-18. Performance Comparison

The experiment calculates precision, recall, and accuracy measures to evaluate the prediction performance. Finally, the performance of the proposed method is evaluated by calculating the performance measures such as precision, recall, accuracy, and F-measures. Three different datasets are collected from three other states: Tamil Nadu, Kerala, and Andhra Pradesh assumed as Dataset-1, Dataset-2, and Dataset-3, which are given in Figure-18. The parameter precision represents the percentage of correct answers associated with and retrieved from the set of all related results. The recall is the measure that represents the number of solutions retrieved from the dataset. The harmonic mean of precision and recall are calculated as F-measure. Based on these parameters, the prediction accuracy is calculated to answer the queries triggered by the users. From the comparison, for Dataset-3, the accuracy is high compared to the other two datasets. The overall performance shows that the proposed method performs better for the real-time public dataset and predicts the eligible people based on the query over financial and medical status.

Conclusion

A hybrid deep learning model is crafted to achieve the aims of this paper, leveraging a Recurrent Neural Network (RNN) algorithm for feature engineering on a crowdsourced dataset procured from the public. The dataset encompasses contributions from various sources including government hospitals, family welfare organizations, educational institutions, quasi-governmental bodies, and private enterprises. Pre-processing of this data involves techniques such as binning, pipeline structuring, and interconnection. Binning operations are employed to rectify missing values and outliers, while pipeline structures facilitate the sequential alignment of data. Interconnection ensures coherence among features through a unified key. RNN is deployed for comprehensive data analysis, with experimentation conducted across datasets sourced from three distinct Indian states. The RNN model orchestrates feature interconnections across multiple data inputs, culminating in the classification of the entire dataset into vectors, thereby enabling precise predictions in response to user queries. These predictions guide governmental actions, including the

implementation of welfare schemes such as scholarships. The experimental findings showcase the RNN model's robust performance, yielding an accuracy of 93.01%, precision of 98.2%, recall of 94.3%, and an F1-Score of 95.8%. These results underscore the applicability and efficacy of the proposed methodology in real-time societal contexts, offering valuable assistance to governmental initiatives.

Future Work

In regard to hardware, it's imperative to boost both the processor speed and dynamic memory capacity. This is essential as we've observed RNN encountering memory constraints while processing and safeguarding sequences of data and sensitive information. Given the escalating data volumes in real-time applications, transitioning to LSTM for RNN extension is warranted, with performance validation being paramount. Our study encountered challenges in sourcing data on a state-by-state basis. The reluctance of citizens to divulge personal information to government officials posed a significant obstacle to data collection. Additionally, the process was protracted and laborious, leading to considerable delays in completion.

References

1. Patel, K., Fogarty, J., Landay, J. A., & Harrison, B. (2008, April). Investigating statistical machine learning as a tool for software development. In *Proceedings of the SIGCHI* Conference on Human Factors in Computing Systems (pp. 667-676).
2. Dai, P., & Weld, D. S. (2010, July). Decision-theoretic control of crowdsourced workflows. In Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (pp. 1168-1174).
3. Kamar, E., Kapoor, A., & Horvitz, E. (2013, June). Lifelong learning for acquiring the wisdom of the crowd. In Twenty-Third International Joint Conference on Artificial Intelligence.
4. Rzeszotarski, J. M., &Kittur, A. (2011, October). Instrumenting the crowd: using implicit behavioral measures to predict task performance. Proceedings of the 24th annual ACM symposium on User interface software and technology (pp. 13-22).
5. Brabham, D. C. (2013). Crowdsourcing. Massachusetts Institute of Technology.
6. Law, E., & von Ahn, L. (2011). Human computation: Synthesis lectures on artificial intelligence and machine learning. Morgan and Claypool, 13.
7. Michelucci, P. (Ed.). (, 2013). Handbook of human computation (Vol. 2, No. 3). New York: Springer.
8. Franklin, M. J., Kossmann, D., Kraska, T., Ramesh, S., & Xin, R. (2011, June). CrowdDB: answering queries with crowdsourcing. In Proceedings of the 2011 ACM SIGMOD International Conference on data management (pp. 61-72).
9. R. Xu and D. C. W. II, "Survey of clustering algorithms," IEEE Trans. Neural Networks, vol. 16, no. 3, pp. 645–678, 2005.
10. P. G. Ipeirotis, "Analyzing the amazon mechanical Turk marketplace," ACM Crossroads, vol. 17, no. 2, pp. 16–21, 2010.
11. Stantchev, V., Colomo-Palacios, R., Soto-Acosta, P., &Misra, S. (2014). Learning management systems and cloud file hosting services: A study on students' acceptance. Computers in Human Behavior, 31, 612–619.

12. Haoqi Zhang, Edith Law, Rob Miller, Krzysztof Gajos, David Parkes, and Eric Horvitz. 2012. Human computation tasks with global constraints. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM. 217–226.
13. Aniket Kittur, Boris Smus, SusheelKhamkar, and Robert E Kraut. 2011. Crowdforge: Crowd-sourcing complex work. In Proceedings of the 24th annual ACM symposium on User interface software and technology. ACM, 43–52.
14. Aniket Kittur, Jeffrey V Nickerson, Michael Bernstein, Elizabeth Gerber, Aaron Shaw, John Zimmerman, Matt Lease, and John Horton. 2013. The future of crowd work. In Proceedings of the 2013 conference on computer-supported cooperative work. ACM, 1301–1318.
15. Justin Chang, Jaime Teevan, Shamsi T Iqbal, and Michael S Bernstein. 2015. Break it down: A comparison of macro-and microtasks, in Proceedings of CHI.
16. Michael S Bernstein, Greg Little, Robert C Miller, Bjorn ¨ Hartmann, Mark S Ackerman, David R Karger, David Crowell, and Katrina Panovich. 2010. Soylent: a word processor with a crowd inside. In Proceedings of the 23rd annual ACM symposium on User interface software and technology. ACM. 313–322.
17. Aniket Kittur, Boris Smus, SusheelKhamkar, and Robert E Kraut. 2011. Crowdforge: Crowd-sourcing complex work. In Proceedings of the 24th annual ACM symposium on User interface software and technology. ACM, 43–52.
18. Cheng, J., & Bernstein, M. S. (2015, February). Flock: Hybrid crowd-machine learning classifiers. Proceedings of the 18th ACM conference on computer-supported cooperative work & social computing (pp. 600-611).
19. Hartmann, B., Abdulla, L., Mittal, M., and Klemmer, S. R. Authoring sensor-based interactions by demonstration with direct manipulation and pattern recognition. In Proc. CHI (2007).
20. Fails, J. A., and Olsen Jr, D. R. Interactive machine learning. In Proc. IUI (2003).
21. Fogarty, J., Tan, D., Kapoor, A., and Winder, S. Cueflik: interactive concept learning in image search. In Proc. CHI (2008).
22. Settles, B. Closing the loop: Fast, interactive semi-supervised annotation with queries on features and instances. In Proc. EMNLP (2011).
23. Dai, P., Weld, D. S., et al. Decision-theoretic control of crowdsourced workflows. In Proc. AAAI (2010).
24. Kamar, E., Kapoor, A., and Horvitz, E. Lifelong learning for acquiring the crowd's wisdom. In Proc. JCAI (2013).
25. Whitehill, J., Ruvolo, P., Wu, T., Bergsma, J., and Movellan, J. R. Whose vote should count more: Optimal integration of labels from labelers of new expertise. In Proc. NIPS (2009).
26. Rzeszotarski, J. M., and Kittur, A. Instrumenting the crowd: using implicit behavioral measures to predict task performance. In Proc. UIST (2011).
27. Rzeszotarski, J., and Kittur, A. Crowdscape: interactively visualizing user behavior and output. In Proc. UIST (2012).

28. Khalaf, O. I., & Abdulsahib, G. M. (2021). Design and Performance Analysis of Wireless IPv6 for Data Exchange. Journal of Information Science and Engineering 37, 1335-1340 (2021). DOI: 10.6688/JISE.202111_37(6).0008
29. U. Srilakshmi, N. Veeraiah, Y. Alotaibi, S. Alghamdi, O. I. Khalaf, and B. V. Subbayamma, "An Improved Hybrid Secure Multipath Routing Protocol for MANET," in IEEE Access, DOI: 10.1109/ACCESS.2021.3133882
30. Rajendran, S., Khalaf, O.I., Alotaibi, Y. et al. MapReduce-based big data classification model using feature subset selection and hyper-parameter tuned deep belief network. Sci Rep 11, 24138 (2021). https://doi.org/10.1038/s41598-021-03019-y
31. Khalaf, O.I., Abdulsahib, G.M. Optimized dynamic storage of data (ODSD) in IoT based on blockchain for wireless sensor networks. Peer-to-Peer Netw. Appl. (2021). https://doi.org/10.1007/s12083-021-01115-4
32. Raghupathy, V., Khalaf, O. I., Andrés, C., Sengan, S., Sharma, D. K. (2022). Interactive Middleware Services for Heterogeneous Systems. Computer Systems Science and Engineering, 41(3), 1241–1253
33. Rajalakshmi, M., Saravanan, V., Arunprasad, V., A., C., Khalaf, O. I. et al. (2022). Machine Learning for Modelling and Control of Industrial Clarifier Process. Intelligent Automation & Soft Computing, 32(1), 339–359.
34. Surendran, R., Khalaf, O. I., Andres, C. (2022). Deep Learning-Based Intelligent Industrial Fault Diagnosis Model. CMC-Computers, Materials & Continua, 70(3), 6323–6338.
35. Alsubari, S. N., Deshmukh, S. N., Alqarni, A. A., Alsharif, N., H., T. et al. (2022). Data Analytics for the Identification of Fake Reviews Using Supervised Learning. CMC-Computers, Materials & Continua, 70(2), 3189–3204.
36. Palanisamy, S.; Thangaraju, B.; Khalaf, O.I.; Alotaibi, Y.; Alghamdi, S. Design and Synthesis of Multi-Mode Bandpass Filter for Wireless Applications. Electronics 2021, 10, 2853. https://doi.org/10.3390/electronics10222853
37. Rout, R.; Parida, P.; Alotaibi, Y.; Alghamdi, S.; Khalaf, O.I. Skin Lesion Extraction Using Multiscale Morphological Local Variance Reconstruction Based Watershed Transform and Fast Fuzzy C-Means Clustering. Symmetry 2021, 13, 2085. https://doi.org/10.3390/sym13112085
38. Nathalia Ospina García. et al. (2021). Remote Academic Platforms in Times of a Pandemic. International Journal of Emerging Technologies in Learning, 16(21), 121–131.