A CROSS GENE-CHEMICAL-DISEASE(G-C-D) BASED DOCUMENT CLUSTERING AND CLASSIFICATION MODEL USING DEEP LEARNING FRAMEWORK

JOSE MARY GOLAMARI[1], D. HARITHA[2]

[1]Research Scholar, Department of Computer Science and Engineering, , Koneru Lakshmaiah Education Foundation,Vaddeswaram, Guntur, Andhra Pradesh.

[2]Professor, Department of computer science and engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh

ABSTRACT

In the current biomedical repositories, gene and disease identification and prediction are the essential factors for content clustering and classification models. Since, most of the biomedical databases have heterogeneous features with different levels of gene patterns. Gene identification and clustering of high dimensional patterns in cross biomedical repositories are complex and difficult to process due to noise, uncertain and missing values. In the traditional biomedical repositories, data classification algorithms are used to classify the documents using the MeSH terms or user specific keywords. These models are difficult to find the relational genes and its disease patterns in different biomedical repositories. In the proposed work, a hybrid cross gene-chemical-disease based document clustering and classification model is implemented using the deep learning framework. Experimental results proved that the proposed deep learning-based G-C-D document classification has better optimization than the existing models.

Keywords: *Gene Based Micro-Array Dataset, Feature Selection Measures, Gene Classifiction Disease Prediction.*

1. INTRODUCTION

Biomedical documents contain different types of genes and disease types, so they are difficult to find its similar genes or diseases manually in large databases. Many biomedical prediction techniques have been designed to exploit the gene entity and disease prediction process. Documents that are significant to a particular concept will be assigned in a unique cluster in an optimistic way. An automatic document classification and clustering algorithm in any distributed databases will more easily and accurately determine high quality documents at the peer end. The aim and purpose of the work is to improve the classification, prediction and clustering methods of the TREC document in order to simplify the large documents in the distributed databases. The Internet has allowed access to a vast collection of information worldwide. This facility has stimulated an increasing demand for understanding how different and heterogeneous sources of information can be integrated. The research focuses primarily on the identification and incorporation of relevant information to provide a better knowledge of a particular domain. The combination is mostly useful if it allows communication between dissimilar sources without affecting their autonomy[1].

Neural networks consist of and the link between neurons. Neural network evolution is a significant outcome of the artificial intelligence applied to human-oriented information mining systems. The primary objective of ANN is to model human brain through mathematical computations. The characteristics of ANN based systems are accuracy, low noise, assumption free transmission, ease of maintenance. ANN's success is often well proven with architectures focused on parallel processing. Integrated with deep learning, ANN has developed into many other types which are targeted to different application domains[2].

Document Similarity is examined using one of the document similarity measures, based for instance, the Jaccard measure and the cosine measurement, based on a feature vector or word frequency. Clusters based on these vector spaces use the single word, i.e. only one gramme. The amount of information in various areas is growing rapidly as biomedical repositories are distributed. They don't use any word neighborhood or sentence-based clustering. Pre-processing documents is a reduction of peer documents by selecting important information from the source documents to generate a summary[3]. This however caused the problem of overloading information. The gene and extraction of the document can be used to minimize the intercluster variation to solve this problem. This work takes account of the strategy for extracting functions and the key sentence, classification and discovery approach to the eradication of the redundancy of information derived from several original documents. Data analysis recently became a highly active subject for machine learning due to large numbers of data collected in databases. Most traditional document classification algorithms on the basis of genes are utilised for grouping similar tokens with a limited number of document sets. But the number and number of documents that are considered one of the biggest disadvantages of this algorithm must be specified in advance[4].

2. RELATED WORK

The application of ANN-based techniques in classification algorithms will manage complex pattern recognition in various applications. The classic algorithms, including Naive. ANN is capable of data storage, pattern recognition and pattern retrieval in domain fields. The traditional model based on a graph only takes into account the question of the sentence node and the sentence node relation. In this system, the prediction and the measurement sensitive to query is taken into consideration from sentence to sentence edge. Another model that uses TextRank with certain differences has been presented. This procedure uses a shortest way to generate the TextRank summary. In the first phase, the graph model was created to represent the document and interconnected phrase entities with significant relationships in the graph model[5]. Using a clustering model, a document extraction process was presented. This model is based on the extraction of pre-computed features for single and multiple documents.In a Bayesian network, words are regarded independent of one another, and tf-idf is used to generate individual scores. Because of the growing amount of online textual information, automatic text classification has gotten a lot of attention. In the subject of automatic text categorization, numerous statistical text classification and machine learning approaches such as Naive- Bayes, K-Nearest Neighbor (KNN) Neural Network, and others have been applied in recent years. The centroid based classifier (CBC) is the most used method, however it is very dependent on the data distribution.[6] suggested a gravitation model to tackle the CBC's unbalance classification problem, based on Newton's law of universal gravitation. Text classification and summarization are often regarded as separate research topics in the literature. Both can benefit from one other's assistance. Text summarization problems can benefit from text category information, and text classification issues can benefit from summary information. K- The nearest neighbor (KNN) is an instance-based classification system to support kernel learning in anomaly repositories. The kernel design classification function KNN optimizes the anomaly pattern removal function[7]. Before classification classes are applied, the dataset which is totally skewed towards the majority class is required in the preprocess. In some cases, along with the cases of anomaly, minority classes can be detected. The minority class anomaly is much higher than the majority group. In the case of majority classes, most classification models therefore provide a high level of precision, whereas the minority type is less accurate. A new technique is known as the stratification resampling technique to manage the imbalance (SMOTE). The SMOTE approach that is fundamentally responsible for the over-sampling of examples of minority classes is highlighted here. [8] proposed a reclassification model in biomedical repositories to find the gene for the association of diseases. They predicted the gene ranking score for finding relational attributes to the testing data in the training data.

3. PROPOSED MODEL

Data Preparation Phase

Biomedical document contains the terms and keywords of MeSH which contain essential information for finding the gene and its related diseases. Each document is pre-processed in each biomedical XML document to find the missing genes or diseases. Every XML file is processed to find the terms of biomedicine and the terms of its genes. A novel gene to disease relationships is extracted in the proposed model using the clustering and classification model as shown in figure 1.

Documents are taken together with predefined gene and disease datasets in the proposed document filtering phase, input Medline database (MDB), PubMed database (PDB), and Embase(EDB). Each data instance, in the filtering phase, contains details of the document along with MeSH terms. One million genes are collected to find and extract the gene tags from the input datasets.

Input : MDB,EDB,PDB, Biomedical Disease list, MGenes,PGenes,EGenes

Output: Gene tags extraction, Gene document indexing

Procedure:

```
    Load MGeneDB=MedlinegetGeneNames();
    Load PGeneDB=PubMedgetGeneNames();
    Load EGeneDB=EmbasegetGeneNames();
    Load DiseaseList=getDiseaseList();
    Load Document Sets DS[] from URL(Query)
    For each document D[i]  in DS
    Do
$\text{If}(D[i] \in \text{DiseaseList} \,\&\&\, D[i].\text{type} == \text{PubMed})$
then
$\text{PDocSet}[] = D[i];$
$\text{endif}(D[i] \in \text{DiseaseList} \,\&\&\, D[i].\text{type} == \text{Medline})$
then
$\text{MDocSet}[] = D[i];$
else
$\text{EDocSet}[] = D[i];$

    Done
    For each document PD[i] in PDocSet
    Do
PDTokens[]=Tokenization(PD[]);
SPD[i]=Stemming(PDTokens).
SSPD[i]=Stopword_removal(SPD[i])
PDFR[]=Non-Functional_Remove(SSPD[i]);
        $\text{PDGenes}[] = \text{Extract}(\text{PDFR}[])$

    Done
    // Document filtering and extract Genes from Medline Dataset
    For each document MD[i] in MDocSet
    Do
MDTokens[]=Tokenization(MD[]);
SMD[i]=Stemming(MDTokens).
SSMD[i]=Stopword_removal(SMD[i])
MDFR[]=Non-Functional_Remove(SSMD[i]);
        $\text{MDGenes}[] = \text{Extract}(\text{M DFR}[])$

    Done
    // Document filtering and extract Genes from Embase Dataset
    For each document ED[i] in EDocSet
    Do
EDTokens[]=Tokenization(ED[]);
SED[i]=Stemming(EDTokens).
SSED[i]=Stopword_removal(SED[i])
EDFR[]=Non-Functional_Remove(SSED[i]);
```

```
  EDGenes[] = Extract(E DFR[])
        Done
        CSGenes[]=CombineDocSetsGenes(
PGenes[] , MDGenes[] , EDGenes[] )
        CSDocs[]=CombineDocSets( PDocSet[] ,
MDocSet[] , EDocSet[] )
    For(i=0;i<CSGenes.length;i++)

for(j = 0; j < Gene _DB.length; j+ +)
do
if (CS Genes[i] == Gene _ DB[j])
then
Add(CSSlist < −Map(GeneDB, CSGenes, Sim(U, V))
/ / Combined  Document sets Similarity  list which conta
combined documents with similarity scores.

                end  if
                done
        done
        Add GeneDocs(CSDocs[], CCSlist )
```

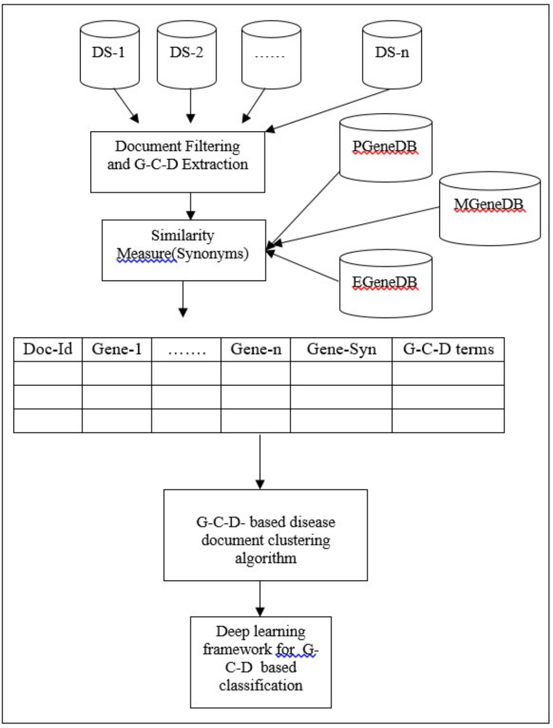
DS-1
DS-2
……
DS-n
Document Filtering and G-C-D Extraction
PGeneDB
MGeneDB
Similarity Measure(Synonyms)
EGeneDB
Doc-Id
Gene-1
…….
Gene-n
Gene-Syn
G-C-D terms
G-C-D- based disease document clustering algorithm
Deep learning framework for G-C-D based classification

Figure 1: Internal G-C-D processing

$$\mathrm{Sim}(G_i, C_i, D_i) \leftarrow \left(\frac{\max|\mu_G(c), \mu_C(c), \mu_D(c)|}{\min|\sigma^2_{GeneDB}(c), \sigma^2_{Clist}(c), \sigma^2_{Dlist}(c)|}\right) \cdot \frac{P(GeneDB[i]/(C\,Docs[i])}{P(GeneDB[i]/D\,Genes[i])}$$

Where c is c biomedical categories(Ex: c=1,2,3..1=PubMed related documents)

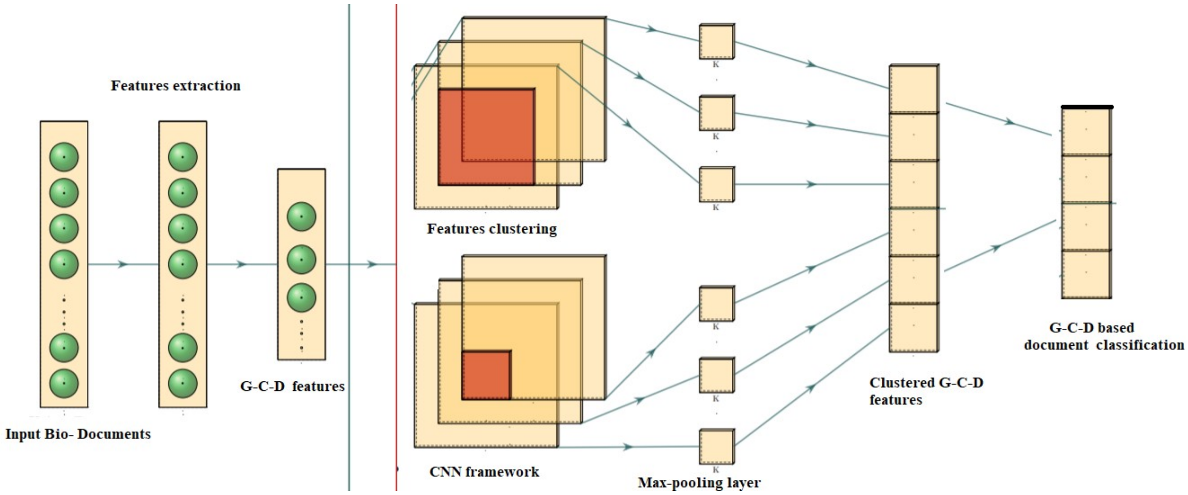

Figure 2: Proposed G-C-D based Document clustering and classification model

2) *Cross Gene-Chemical-Disease based Clustering Algorithm*

In this phase, each document from the G-C-D documents dataset is processed using the clustering algorithm. Here, a cross entity relationships are extracted among the G-C-D terms and the multiple biomedical document repositories such as pubmed,medline,embase document sets.

Input : G-C-D gene synonym database (GeneSyDB), G-C-D document sets.

Procedure:

Step 1: Extract synonym gene terms to each document of the biomedical repository as

MGSynDocs=Medline documents genes;
PGSynDocs=Pubmed documents genes;
EGSynDocs=Embase documents genes;

Step 2: Computing feature selection measures between repository documents for document clustering.

$$D_i = CSDocs[]; // i = 1 \text{ to } |G-C-D\,data|$$

$$\sigma(P,E) = r(P,E).\sum_{j=1} \log\left(\frac{Pro\left(\frac{D_i}{PGSynDocs[j]}\right)}{Pro\left(\frac{D_i}{EGSynDocs[j]}\right)}\right).CP(P,E)$$

$$CP(P,E) = (PGSynDocs[j] \cap EGSynDocs[j]);$$

$$r(P,E) = \prod_{j=1}^{|getGSynonym[j]\}} Correlation(D_i, PGSynDocs[j], EGSynDocs[j]);$$

$$\sigma(P,M) = r(P,M).\sum_{j=1} \log\left(\frac{Pro\left(\frac{D_i}{PGSynDocs[j]}\right)}{Pro\left(\frac{D_i}{MGSynDocs[j]}\right)}\right).CP(P,M)$$

$$CP(P,M) = (PGSynDocs[j] \cap MGSynDocs[j]);$$

$$r(P,M) = \prod_{j=1}^{|getGSynonym[j]\}} Correlation(D_i, PGSynDocs[j], MGSynDocs[j]);$$

$$\sigma(E,M) = r(E,M).\sum_{j=1} \log\left(\frac{Pro\left(\frac{D_i}{EGSynDocs[j]}\right)}{Pro\left(\frac{D_i}{MGSynDocs[j]}\right)}\right).CP(E,M)$$

$$CP(E,M) = (EGSynDocs[j] \cap MGSynDocs[j]);$$

$$r(E,M) = \prod_{j=1}^{|getGSynonym[j]\}} Correlation(D_i, EGSynDocs[j], MGSynDocs[j]);$$

$Pro(D_i \cap getGSynonym[j])$ is the common probability of the D_i in the given gene synonym list.

$Pro(D_i \cup getGSynonym[j])$ is the sum of all probability of D_i to the given gene synonym list.

$Pro(D_i)$ is the probability of G-C-D terms that contain in all the document sets.

Step 3: Apply improved Expectation maximization clustering model using the document feature selection measure as weights.

Step 4: Obtain initial number of document clusters as m representative objects. Here representative objects are selected using mean document frequencies (TFIDF).

Step 5: Compute the distance measure of the representative objects to the other document objects using chebyshev distance measure.

Step 6: Repeat steps 4 and 5 until k initial clusters.

Step 7: Initialize the model parameters using the multinomial naive bayes algorithm.

Step 8: In Expectation step, a novel posterior probability computation is used to optimize the class prediction in the gene to disease document sets. Let $\hat{\theta}$ be the model parameter to be estimated using the posterior probability computation.

$Prob(\hat{\theta}/c)$ is the probability of the occurrence of the gene in the given document sets.

$Prob(w/c,\hat{\theta})$ is the probability of occurrence of the disease term in the given document category sets.

Step 9 :

Grouping the gene to disease document sets using the naive parameter estimation computation in proposed EM model as

$$Prob_{P,E}(c/D_i,\hat{\theta}) = Sim(G_i, C_i, D_i).\left\{\frac{Prob_{P,E}(D_i/c,\hat{\theta}).Prob_{P,E}(c/\hat{\theta})}{Prob_{P,E}(D_i/\hat{\theta})}\right\}$$

$$Prob_{P,M}(c/D_i,\hat{\theta}) = \frac{Prob_{P,M}(D_i/c,\hat{\theta}).Prob_{P,M}(c/\hat{\theta})}{Prob_{P,M}(D_i/\hat{\theta})}$$

$$Prob_{M,E}(c/D_i,\hat{\theta}) = \frac{Prob_{M,E}(D_i/c,\hat{\theta}).Prob_{,E}(c/\hat{\theta})}{Prob_{M,E}(D_i/\hat{\theta})}$$

$$If(\,(Prob_{P,E}(c/D_i,\hat{\theta}) > \lambda_{P,E})\,)$$

```
    Then
            Add to cluster gene-disease group
            $C_{P,E}(\sigma(P,E)$ ,PGSynDocs,
            EGSynDocs,
            $\Pr ob_{P,E}(c/D_i,\hat{\theta})$ );
     End If( $(\Pr ob_{P,M}(c/D_i,\hat{\theta}) > \lambda_{P,M})$ )
    Then
            Add to cluster gene-disease group
            $C_{P,M}(\sigma(P,M)$ ,PGSynDocs,
            MGSynDocs,
            $\Pr ob_{P,M}(c/D_i,\hat{\theta})$ );
            End
If( $(\Pr ob_{E,M}(c/D_i,\hat{\theta}) > \lambda_{E,M})$ )
    Then
            Add to cluster gene-disease group
            $C_{E,M}(\sigma(E,M)$ ,EGSynDocs,
            MGSynDocs,
            $\Pr ob_{E,M}(c/D_i,\hat{\theta})$ );
    End
Training cluster data : C={ $C_{P,M}$ , $C_{E,M}$ , $C_{P,E}$ }
```

Convolution neural networks are the neural network used to process data which has a grid like topology known. Processing of natural language tasks using convolution neural networks, use the 1D structure of text data for precise prediction. A convolution neural network provides superior classification accuracy due to the network's non-linearity and the ability to easily incorporate pre-train embedding of terms. Convolution neural networks are networks of convolution and pooling layers, useful for classification tasks such as classification of emotions, etc. Convolution, and pooling architectures are applied to text in convolution neural networks. Natural language processing mainly involves sequence convolutions of one dimension. Convolution is a specialized kind of linear operation and pooling in convolution neural networks is an operation. Different filters with varying window sizes (here, only the height is different) slide over the full E rows in the convolution layer, i.e. the filter width is usually the same as the E width. Each filter conducts E-convolution, generating various function maps. A max-over-time pooling operation is applied to the elements located in the same feature map to extract the most important feature. Max-over-time pooling operation is performed on the PSO features to find the essential key relationships among the gene disease based chemical drug terms as shown in figure 4. In the proposed CNN framework, a multi-objective SVM is proposed to the classify the input tokens for chemical drug prediction based on the gene disease patterns.

Input the CNN features for data classification.

For each feature set do

for each disease in GDP.

do

Apply SVM multi-class optimization models as

$$\min_{W_k,a_k} \frac{1}{2}\|W_k\|_1^2 + \tau_m + \sum_{i=1}^{l} a_i\left(y_i\left[ker < x,y > \cdot w + b\right] - 1 + \xi_i\right) - \sum_{i=1}^{l} \gamma_i \xi_i$$

$$s.t\ ker < x,y > \cdot w + b \geq 1 - \xi_i^n - \tau_m,$$

$$\xi_i^n > 0$$

$$\tau_m > 0; m = 1 \ldots classes$$

Here kernel function ker(x,y) represents the kernel functions defined from gene disease vector space to chemical symbol vector space.

$$Ker < x,y >= e^{-\xi_i^n \log\left(\sum\|x-y\|^2\right)} \quad \text{if x==y}$$

$$= e^{-\xi_i^n \log\left(\sum\|x-y\|^{1/2}\right)} \quad \text{if x<y}$$

$$=$$

$$e^{-\xi_i^n \log\left(\sum\|y\|^2\right)} \quad \text{if x > y}$$

Step 4: Test data is predicted to the class y based on the largest decision values as

$$arg\ max\{W_k^T D_i + b_k\}$$

IV.Experimental Results

Proposed model is simulated in Amazon AWS cloud server with 48GB of RAM. In the AWS server uses a variety of deep learning frameworks to find the best combination of speed and accuracy for large datasets. Using a large AWS instance, a java-based deep learning framework is used to filter out the less important key patterns and classification models in the proposed work. Java deep learning framework is used to simulate the proposed model in order to increase classification speed. Our frameworks are put through their paces on a variety of datasets, including gene, chemical, and biomedical ones. The accuracy of gene chemical prediction is measured using a variety of metrics, including recall, precision, F1 value, and area under the curve (AUC).

Table 1: Gene-Disease Features Extraction Using The Proposed PSO Model

ABC	ACO	Chi-square	Mutual information	Information Gain	Genetic Algorithm	PSO	Proposed PSO
58	59	56	78	63	56	51	41
53	60	54	78	60	52	56	37
55	62	54	74	59	62	69	43
56	63	52	82	57	53	60	46
58	54	55	63	58	60	72	48
56	50	59	85	64	54	52	44
65	61	65	75	58	58	69	42
53	62	65	77	55	62	75	50
64	57	63	79	50	62	56	41
65	64	52	85	59	57	69	36
60	62	71	69	62	56	50	36
56	55	61	70	57	61	70	38
59	61	70	67	65	58	74	42
56	53	67	69	62	60	58	45
54	58	59	69	51	55	60	39
51	54	72	74	60	57	72	45

53	59	64	61	57	53	60	46
60	61	59	85	50	58	65	44
56	59	62	76	62	61	69	38
58	52	62	71	51	60	54	38

Table1 shows the results of gene-disease feature extraction using the proposed method on large datasets. Table1 clearly demonstrates that the current feature extraction method has a much higher filtering rate than previous methods.

Table 2:Chemical-Disease Features Extraction Using The Proposed PSO Model

ABC	ACO	Chisquare	Mutual information	Information Gain	Genetic Algorithm	PSO	Proposed PSO
64	55	63	66	60	56	60	48
64	61	52	77	62	59	66	47
57	54	58	80	56	56	53	48
62	59	50	72	64	58	66	38
55	64	63	65	51	58	65	35
60	64	53	77	54	54	60	47
60	65	68	71	54	63	72	46
63	63	73	66	64	51	68	40
53	51	54	78	54	58	52	46
53	64	74	71	57	52	52	40
61	64	73	80	61	62	59	46
58	55	55	75	56	57	58	48
57	63	56	64	54	58	51	44
58	58	70	74	60	59	72	42
64	59	60	81	50	65	61	42
55	58	66	65	55	52	69	36
61	61	69	60	59	65	50	48
52	60	62	84	56	53	58	47
54	65	51	64	63	53	72	44
64	54	57	66	50	50	63	37

Table2 shows how well the proposed PSO approach performs on large datasets when extracting chemical-disease features. According to table2, this feature extraction method has a higher filtering rate than the other approaches.

Table 3: Performance Analysis Of Computational Runtime(Ms) With Different Traditional Feature Selection Models

Features Size	ABC	ACO	Chisquare	Mutual information	Information Gain	Genetic Algorithm	PSO	Proposed_PSO
GeneDisease-100	5417	7117	6297	7230	7024	6638	6481	4747
GeneDisease-200	5365	6071	6529	6957	5917	6253	6516	3965
GeneDisease-300	6200	5753	6609	5959	6882	7099	6469	3474
GeneDisease-400	6122	6801	5514	6055	5729	7430	5584	3495
GeneDisease-500	6727	5488	5676	7488	6685	6103	6753	4809
GeneDisease-600	5774	5592	6089	6823	6273	5996	6597	4292
GeneDisease-700	7205	6524	6045	6060	7449	7448	6319	3888
GeneDisease-800	6976	5859	7340	6036	6544	6855	5864	4723
GeneDisease-900	7080	6097	5710	6692	5661	6866	7365	4488
GeneDisease-1000	5854	6731	7118	7087	5916	5541	5552	4368
GeneDisease-1100	6431	7548	7224	5597	7359	7022	6253	4470
GeneDisease-1200	6595	5404	5767	5989	7086	5950	7552	3995
GeneDisease-1300	6077	6919	7018	5357	6242	5645	6886	4335
GeneDisease-1400	7350	5536	6953	5711	6927	7133	7061	4439
GeneDisease-1500	7536	7139	5695	5414	6211	6909	6438	4303
GeneDisease-1600	7557	7237	7425	6421	6729	6922	5686	3618
GeneDisease-1700	7032	7402	5520	6334	5614	5942	5840	4524
GeneDisease-1800	7556	5435	7183	5417	5700	7236	5630	4734
GeneDisease-1900	5669	6159	5589	5708	7281	5900	6588	4568
GeneDisease-2000	6318	7119	6253	5603	6852	5925	7090	4178

On large datasets, the proposed PSO approach's computational runtime (ms) for gene-disease feature extraction is shown in Table3. When compared to other feature extraction methods, the one used in this study has a shorter computation runtime (see table3).

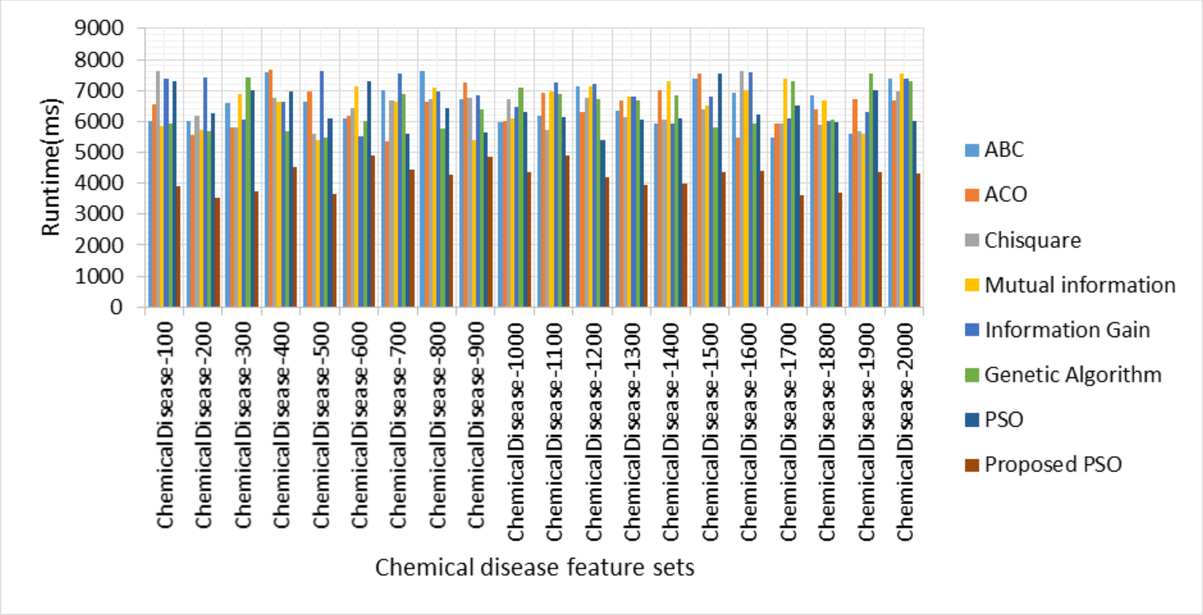

Table3, describes the performance of computational runtime(ms) of gene-disease feature extraction using the proposed PSO approach on large datasets. From the table3, it is clearly shown that the present feature extraction procedure has low computation runtime as compared to the existing approaches.

Recall

Table 4: Performance Analysis Of Recall Using Different Traditional Deep Learning Frameworks.

Features_Size	SVM	Random_Forest	CNN	RNN	HNN	Proposed-CNN-ISVM
GCDisease-100	0.83	0.8	0.82	0.86	0.83	0.98
GCDisease-200	0.87	0.87	0.85	0.92	0.8	0.96
GCDisease-300	0.84	0.84	0.89	0.89	0.82	0.98
GCDisease-400	0.89	0.76	0.9	0.91	0.86	0.96
GCDisease-500	0.78	0.79	0.76	0.81	0.84	0.96
GCDisease-600	0.83	0.74	0.71	0.82	0.83	0.97
GCDisease-700	0.85	0.78	0.79	0.83	0.82	0.97
GCDisease-800	0.74	0.74	0.82	0.89	0.92	0.98
GCDisease-900	0.86	0.8	0.81	0.81	0.8	0.96
GCDisease-1000	0.84	0.75	0.82	0.9	0.92	0.97
GCDisease-1100	0.75	0.8	0.78	0.88	0.92	0.98

GCDisease-1200	0.74	0.77	0.83	0.88	0.84	0.97
GCDisease-1300	0.83	0.86	0.79	0.82	0.85	0.98
GCDisease-1400	0.78	0.77	0.82	0.84	0.9	0.97
GCDisease-1500	0.89	0.86	0.73	0.8	0.84	0.96
GCDisease-1600	0.86	0.78	0.7	0.91	0.81	0.97
GCDisease-1700	0.71	0.77	0.86	0.92	0.86	0.97
GCDisease-1800	0.82	0.74	0.84	0.9	0.81	0.96
GCDisease-1900	0.71	0.83	0.83	0.83	0.81	0.98
GCDisease-2000	0.87	0.86	0.71	0.91	0.85	0.97

Table 4 shows the accuracy of the proposed deep learning framework for chemical-gene-disease classification on large datasets. There is a clear comparison between the current framework and the existing frameworks in table4.

Precision

Table 5: Performance Analysis Of Precision Using Different Traditional Deep Learning Frameworks.

Features_Size	SVM	Random_Forest	CNN	RNN	HNN	Proposed-CNN-ISVM
GCDisease-100	0.75	0.85	0.76	0.8	0.8	0.96
GCDisease-200	0.78	0.88	0.83	0.92	0.8	0.97
GCDisease-300	0.88	0.77	0.89	0.86	0.9	0.97
GCDisease-400	0.83	0.78	0.9	0.91	0.82	0.97
GCDisease-500	0.7	0.84	0.88	0.81	0.84	0.96
GCDisease-600	0.75	0.87	0.83	0.86	0.93	0.98
GCDisease-700	0.84	0.78	0.74	0.87	0.93	0.97
GCDisease-800	0.73	0.73	0.85	0.89	0.87	0.98
GCDisease-900	0.85	0.78	0.85	0.84	0.82	0.97
GCDisease-1000	0.72	0.76	0.76	0.93	0.92	0.98
GCDisease-1100	0.75	0.74	0.75	0.87	0.84	0.98
GCDisease-1200	0.72	0.8	0.87	0.85	0.84	0.98
GCDisease-1300	0.8	0.77	0.78	0.93	0.83	0.96
GCDisease-1400	0.74	0.8	0.73	0.9	0.89	0.97
GCDisease-1500	0.76	0.87	0.8	0.91	0.84	0.97
GCDisease-1600	0.86	0.89	0.83	0.84	0.82	0.96
GCDisease-1700	0.84	0.72	0.89	0.86	0.83	0.97
GCDisease-1800	0.88	0.72	0.74	0.8	0.87	0.97
GCDisease-1900	0.82	0.83	0.8	0.88	0.82	0.96
GCDisease-2000	0.76	0.87	0.79	0.9	0.85	0.96

Deep learning-based chemical-gene-disease classification performed well when applied to large datasets, as shown in Table 5. Comparing the current framework to other existing frameworks (table5), it is clear that the current one has higher computational precision.

F1-Measure

Table 6: Performance Analysis Of F1-Measure Using Different Traditional Deep Learning Frameworks.

Features_Size	SVM	Random_Forest	CNN	RNN	HNN	Proposed-CNN-ISVM
GCDisease-100	0.82	0.77	0.76	0.82	0.92	0.97
GCDisease-200	0.81	0.83	0.83	0.92	0.88	0.97
GCDisease-300	0.74	0.8	0.73	0.83	0.86	0.96
GCDisease-400	0.78	0.78	0.72	0.91	0.85	0.97
GCDisease-500	0.81	0.71	0.83	0.8	0.93	0.96
GCDisease-600	0.88	0.86	0.89	0.83	0.88	0.98
GCDisease-700	0.88	0.72	0.87	0.88	0.83	0.97
GCDisease-800	0.87	0.79	0.73	0.88	0.88	0.96
GCDisease-900	0.7	0.77	0.76	0.92	0.87	0.98
GCDisease-1000	0.86	0.74	0.79	0.86	0.93	0.96
GCDisease-1100	0.74	0.74	0.76	0.86	0.81	0.96
GCDisease-1200	0.82	0.87	0.78	0.8	0.93	0.97
GCDisease-1300	0.86	0.79	0.84	0.84	0.86	0.98
GCDisease-1400	0.79	0.78	0.82	0.84	0.91	0.96
GCDisease-1500	0.82	0.86	0.71	0.93	0.88	0.98
GCDisease-1600	0.73	0.81	0.86	0.89	0.82	0.97
GCDisease-1700	0.83	0.88	0.8	0.91	0.85	0.97
GCDisease-1800	0.71	0.79	0.86	0.92	0.91	0.98
GCDisease-1900	0.75	0.77	0.86	0.84	0.94	0.96
GCDisease-2000	0.82	0.8	0.8	0.91	0.91	0.97

Table6 shows the results of the proposed deep learning framework on large datasets for the F1-measure of chemical-gene-disease classification. According to table6, this framework has a higher computational F1-measure than any of the other existing ones.

Improvement on Traditional literature work:

A weighted charting method for extraction of document features has been proposed using a new approach that includes ranking both phrases and phrases[9]. A three-phase method was presented. In the initial step, the document structure can be represented as a graph undirected for each document in the set document. Document sentences play an important role in the formation of sentences in the graph model. In the second step, the ranking technique is used to calculate each phrase in the document. Finally, for creating the relevant summary, the maximum marginal relevance technique is used. The number of clusters, the centroid of the cluster and the type of domain or application will depend on the cluster. Most of the cluster model consists of pre-processing, clustering and extraction functions[10].

5 CONCLUSION

In this paper, a novel gene-disease prediction on multiple biomedical document repositories is predicted using the Hadoop framework. The uniqueness of the proposed approach lies in two ways. One is that our model is efficient against noisy data. The other one is the prediction rate of the gene and its related disease patterns on large biomedical databases. An efficient clustering based classification model is developed for gene to disease clustering for document classification model. In this model, an improved clustering is implemented to find the gene to disease prediction in multiple biomedical repositories for classification. Experimental results proved that the presented model has high computational relationship and accuracy for gene-disease clustering and classification compared to the traditional models.

This work is extended to real-time time series gene datasets for the distributed applications. In order to handle distributed applications, a novel parallel biomedical gene disease prediction approach is required to filter the patterns on large candidate sets.

REFERENCES

[1] R. Xu and D. C. Wunsch, "Clustering Algorithms in Biomedical Research: A Review," in IEEE Reviews in Biomedical Engineering, vol. 3, pp. 120-154, 2010.

[2] J. Chiang, C. C. H. Liu, Y. H. Tsai and A. Kumar, "Discovering Latent Semantics in Web Documents Using Fuzzy Clustering," in IEEE Transactions on Fuzzy Systems, vol. 23, no. 6, pp. 2122-2134, Dec. 2015.

[3] J. Xuan, J. Lu, G. Zhang, R. Y. D. Xu and X. Luo, "Bayesian Nonparametric Relational Topic Model through Dependent Gamma Processes," in IEEE Transactions on Knowledge and Data Engineering, vol. 29, no. 7, pp. 1357-1369, July 1 2017.

[4] Mark, Wren, J., Herschkowitz, J., Perou, C. and Garner, H. (2007). Clustering microarray-derived gene lists through implicit literature relationships. Bioinformatics, 23(15), pp.1995-2003.

[5] Homayouni, R., Heinrich, K., Wei, L. and Berry, M. (2004). Gene clustering by Latent Semantic Indexing of MEDLINE abstracts. Bioinformatics, 21(1), pp.104-115.

[6] Hu, H. (2010). Mining patterns in disease classification forests. Journal of Biomedical Informatics, 43(5), pp.820-827.

[7] Sezin Ata Kircali,Disease gene classification with metagraph representations. (2017). Methods, 131, pp.83-92.

[8] Hong-Dong Li,A phase diagram for gene selection and disease classification,Chemometrics and Intelligent Laboratory Systems 167 (2017) 208–213.

[9] Andrea Mesa,Hidden Markov models for gene sequence classification,Pattern Anal Applic,2015,23-45.

[10] Erica,Classification of Genes: Standardized Clinical Validity Assessment of Gene-Disease Associations Aids Diagnostic Exome Analysis and Reclassifications,Clinical Validity and Reclassification