Graph Convolutional Network for Occupational Disease Prediction with Multiple Dimensional Data

Khanh Nguyen-Trong[1], Tuan Vu-Van[2], Phuong Luong Thi Bich[3]
Faculty of Information Technology, Posts and Telecommunications Institute of Technology, Hanoi, Vietnam[1,2]
Intelligent Computing for Sustainable Development Laboratory (IC4SD),
Posts and Telecommunications Institute of Technology, Hanoi, Vietnam[1]
Faculty of Information Technology, Hanoi University of Architecture, Vietnam[3]

***Abstract*—Occupational diseases present a significant global challenge, affecting a vast number of workers. Accurate prediction of occupational disease incidence is crucial for effective prevention and control measures. Although deep learning methods have recently emerged as promising tools for disease forecasting, existing research often focuses solely on patient body parameters and disease symptoms, potentially overlooking vital diagnostic information. Addressing this gap, our study introduces a Deep Graph Convolutional Neural Network (DGCNN) designed to detect occupational diseases by utilizing demographic information, work environment data, and the intricate relationships between these data points. Experimental results demonstrate that our DGCNN method surpasses other state-of-the-art methods, achieving high performance with an Area Under the Curve (AUC) of 96.2%, an accuracy of 98.7%, and an F1-score of 75.2% on the testing set. This study not only highlights the effectiveness of DGCNNs in occupational disease prediction but also underscores the value of integrating diverse data types for comprehensive disease diagnosis.**

Keywords*—Occupational disease diagnostics; heterogeneous data; imbalanced data; Graph Convolutional Network (GCN); deep graph convolutional neural network

I. Introduction

Occupational diseases has been a major concern for many years, which are caused by harmful working conditions and production processes that affect the health of workers. Before the common era, Hippocrates (460-377 BC) discovered lead poisoning. In the first century, Pliny the Elder discovered the harmful effects of dust on the human body. In the second century, Galen described the diseases that miners suffered from. In the following centuries, mercury poisoning and other occupational diseases were discovered.

The best way to prevent and control occupational diseases is to detect them early. If dangerous occupational diseases are not detected and treated in time, they can cause permanent damage to humans or even death. However, currently, in developing countries, such as Vietnam, the examination and detection of occupational diseases are still limited. Thousands of workers are usually routinely screened in batches to detect disease or the risk of disease. To screen for the risk of occupational diseases, workers are first examined in general through clinical signs, such as questioning, studying medical records, etc. If it is determined that there is a risk of occupational diseases, workers will be prescribed in-depth paraclinical tests, such as chest X-ray, hearing test, FEV1 pulmonary function test, etc. However, due to the small number of occupational disease doctors, the examination of thousands of workers at the same time leads to low efficiency, long waiting time, and expensive costs. Therefore, a solution for early detection of the risks of occupational diseases is necessary.

Owning to the development of machine learning, many methods have been proposed for disease diagnosis, including K-nearest neighbors (KNN), support vector machines (SVM), random forests (RF), and artificial neural networks (ANN), CNN, RNN [1], [2], [3], [4]. Although these studies have achieved promising results in disease diagnosis, they are difficult to apply in practice due to their strict data requirements. The data must be complete and have a common structure for all patients, which is often not the case with medical data. Such data is often incomplete and heterogeneous among patients.

Recently, the rise of Graph Neural Network has made it easier to solve problems related to heterogeneous data like medical data. The network treat each data sample as a graph with nodes representing the relevant features of the sample. The model then uses the data from the nodes and the relationships between them to synthesize the output data and label the sample. The idea of using GCNs for disease diagnosis is similar [5]. Each patient is treated as a graph with nodes representing the patient's features. The nodes are connected to each other based on the relationships between them. The output data is then synthesized based on the nodes and the relationships between them. In GCNs, each graph does not need to be the same as the other graphs. This means that feature selection is not necessary. This means that important features will not be lost. This model increases the flexibility of the model in processing data. We can also expand and upgrade the dataset arbitrarily without fear of the model failing.

In this paper, we propose the use of a deep graph convolutional neural network (DGCNN) for disease diagnosis. DGCNNs are a type of neural network that is designed to work with graphs. Graphs are a natural way to represent data that has relationships between the data points. For example, a graph can be used to represent the relationships between genes in a genome, or the relationships between symptoms in a disease [6].

DGCNNs have been shown to be effective for a variety of tasks that involve graphs, including image classification and natural language processing. In this paper, we show that DGCNNs can also be used for disease diagnosis. We use a DGCNN to learn the relationships between symptoms and diseases, and then use the learned relationships to predict the

disease for a new patient

II. Related Works

The study focuses primarily on the methods of disease diagnosis based on a number of machine learning (ML) algorithms, so in this section, some studies using medical records to diagnose the disease of the subject will be mentioned.

In principle, disease diagnosis is based on a dataset of many patients with relevant information fields related to the disease diagnosis process. The data that affect the diagnosis of the disease, so it needs information related to the patient's health: weight, body mass index, glucose quantification, etc.

Recently, the problem of disease diagnosis is often approached using classical ML algorithms specialized for labeling problems. With the increasingly development of deep learning (DL) algorithms along with their versatility and convenience, these methods are gradually being used in many different types of problems, including disease diagnosis. However, these classical methods all have a common drawback that they are very much affected by the dataset as well as the weaknesses of the dataset. The lack of many important information fields or unbalanced data is very likely to negatively affect the performance of the diagnostic model.

In the past, most studies in the field of disease prediction have been approached using simple modern machine learning methods such as Naive Bayes, K-Nearest Neighbors (KNN), Decision Tree, or more ancient methods such as traditional statistical methods [7]. In statistical methods, the predictor will rely on the statistical parameters and charts of the dataset to make a judgment. This method has a big disadvantage that the result depends on the predictor and the data. If the dataset is not good and the predictor does not have much experience, the result is very likely to be inaccurate.

Naive Bayes is a simple classification model that is easy to install and has a fast processing speed. However, it has a big disadvantage that it requires the input features to be independent, i.e. the information fields do not have a relationship with each other. This is difficult to happen in reality and will reduce the quality of the model.

KNN is the simplest and easiest-to-use labeling algorithm. The model uses the K coefficient to identify the K nearest samples to the object and then uses the labels of these samples to proceed with the classification for the object to be predicted. The most obvious advantage of this model is that it does not take time for the training process. However, for large datasets, the algorithm takes more time for the calculation process. KNN is very sensitive to noise when the K coefficient is small. The performance of the model depends largely on the quality of the dataset and the K coefficient.

It was not until Deep Learning, a subfield of machine learning, became popular, that disease diagnosis problems were applied to this method. These types of models have the same installation and operation process, they will all use the input dataset to proceed with the training model, the data is split or repeated several times to improve the model after each training. In other words, Deep Learning models allow it to self-learn and improve its accuracy, hence achieving high accuracy. For example, Mohammed Ismail and colleagues [8] presented a deep learning technology in the diagnosis of heart disease by using an artificial neural network (ANN) model. Junaid Rashid and colleagues [9] last year also proposed the ANN model and compared its efficiency with traditional machine learning models. Or most recently, the paper on the application of advanced deep learning models using two models simultaneously CNN and LSTM also in the problem of heart disease diagnosis of Sudha and Kumar [10] .

GCN models actually appeared early, however, they have not been widely applied due to their complexity and difficulty in installation, requiring users to have a certain understanding in the field of Deep Learning [11], [12], [13]. In 2019, Ping Xuan and colleagues successfully applied a model combining GCN and CNN (Convolutional Neural Network) in the diagnosis of IncRNA disease [14] . Recently, Haohui Lu and Shahadat Uddin also presented on the application of GNN (Graph Neural Networks) in the field of disease diagnosis based on electronic data [15]. These models take advantage of the relationship between objects to build a network of relationships between them, so when predicting a sample, the model not only relies on the information fields of that sample but can also use the information fields of other samples related to the sample to be labeled, unlike the old methods, which can only use the unique attributes of the sample to predict the result of that sample.

GCNs have many advantages in the field of disease prediction, but they also have some limitations. All of these models require the design and structure of a graph of relationships between samples or between attribute fields. This requires users to have a deep understanding of the problem as well as the relationships in the dataset. Poorly constructed relationship graph can also reduce the accuracy of the model by not only not taking advantage of important information but also creating noise. The dataset also needs to have enough samples, if not, it will not take advantage of the strengths of the GCNs model because the relationship diagram is too small, with few relationship edges. A large graph means that the model is more complex, making it difficult for users to visualize or fully understand the model.

Methods using GCNs are showing to be effective in disease diagnosis problems than traditional machine learning methods. These models take advantage of the understanding of the dataset as well as the ability to reuse data in the prediction process. However, current methods require users to design the relationship graph for the entire dataset, requiring a deep understanding of the problem. This makes the model very complex and difficult to control. With the DGCNN [6] model, we consider each sample as a graph with child nodes as attribute fields and edges as relationships between them. Thus, we only need to initialize the relationship diagram frame for each sample without having to design the total link diagram between samples. However, this does not reduce the ability to take advantage of the relationships between samples, on the contrary, it makes the model more clear and easy to understand.

These studies have shown the ability of ML algorithms in diagnosing diseases based on medical records. However, these studies still have some limitations, such as:

The datasets used in these studies are small in scale, so the results of these studies may not be well generalized to larger datasets. These studies primarily use classical ML models, so

these models may be affected by the weaknesses of the dataset. In the future, studies on disease diagnosis based on medical records need to use larger datasets and newer ML models to improve the accuracy of diagnostic models.

III. Material and Methods

In this study, we focus on occupational disease data, which is typically heterogeneous and lacks explicit information. Imputing missing data with arbitrary values can hinder model training performance due to discrepancies between imputed and actual values. In this context, with missing data, traditional methods that often rely on statistical models do not explicitly capture the relationships between different features.

GCNs, on the other hand, are well-suited for modeling complex relationships between data points. This makes them a promising approach for occupational disease forecasting, where the data is inherently complex and interrelated.

This challenge can be effectively addressed by employing a graph-based data structure, such as the Graph Convolutional Network (GCN). GCNs have demonstrated their ability to construct relational graphs from individual health records and transform the data into a format that excludes missing values.

Therefore, we propose a novel approach to occupational disease forecasting using graph convolutional neural networks (GCNN). Our approach synthesizes information related to body parameters, working environments, and disease symptoms to predict the likelihood of a worker developing an occupational disease.

In our approach, we define the relationships between different features in terms of their level of influence and correlation. We then use this information to calculate and adjust the data field values before using them for prediction. This allows us to better capture the complex relationships between features and improve the accuracy of our predictions.

Considering the importance of the working environment in occupational disease prediction, we also combined such information with patient's medical reports to build our GCNN. Inspired by the work of [6], we propose a new deep graph CNN (DGCNN) to deal with such complex data. We have updated the network to increase the number of units in each layer. Besides, the new architecture allow us in better handling the input data, avoiding underfitting and reducing the training cost.

To handle inconsistent and insufficient data, we organize each patient's medical report as an information graph network. After the data runs through the graph network, we concatenate the outputs generated from the last graph layer. These are then passed to two fully connected and dropout layers before diagnosing whether the patient is sick or not.

In the next section, we will present in detail the used features and the architecture of our proposed network.

A. Data Selection and Re-sampling

This study utilizes health data primarily derived from subjects' self-reported information and health measurements compiled into reports. This inherent data structure introduces the potential for missing values due to incomplete reporting or subject uncertainty. While some fields with high missing rates may not directly influence the outcome variable, they could still exhibit subtle relationships with other factors, making traditional data cleaning processes cumbersome.

To address these challenges, we leverage DGCNN architecture. Unlike conventional approaches that establish relationships between subjects, DGCNN treats each data sample as a relational graph composed of the individual data fields associated with that subject. This allows us to utilize data samples regardless of their inherent structure, eliminating the need for extensive data cleaning.

To implement DGCNNs effectively, we define a relational graph for each data point. Since not all data categories share inherent relationships, the only object each category is directly connected to is its corresponding subject (through the sample ID). We further refine the graph by connecting categories that exhibit apparent relationships based on domain knowledge. This approach leverages the inherent structure of the data without requiring pre-defined relationships between subjects, making it particularly well-suited for our heterogeneous dataset.

While DGCNN's data structure allows it to handle imbalanced data to some extent, we further improve model training performance by applying re-sampling techniques to the training dataset. Due to the disparity in data sets and the variable feature shapes across different samples, we cannot apply re-sampling algorithms that rely on the original data to generate new data, like Condensed Nearest Neighbors or SMOTE [16]. Instead, we employ two methods to address such class imbalance, including Random Under Sampler for under-sampling [17] and Random Over Sampler for over-sampling [18] [19].

Consequently, we propose two DGCNN models, each using a different type of re-sampling method: one with Random Over Sampling (DGCNNv1) and another with Random Under Sampling (DGCNNv2). This allows us to compare the impact of different re-sampling approaches on model performance in the context of imbalanced data and identify the most effective strategy for our specific dataset.

B. Deep Graph Convolutional Neural Network for Occupational Disease Detection

The first DGCNN architecture, named DGCNNv1 as illustrated in Fig. 1, employs Random Over-sampling to achieve a balanced class distribution with a 1:10 ratio. This technique retains all samples from the majority class while duplicating instances from the minority class until the desired ratio is reached. DGCNNv1 utilizes a DeepGraphCNN layer as its core component, encompassing four child GCN layers. Each GCN layer has a size of 256 channels, except the final layer, which has only one channel and solely serves a sorting purpose. The output tensor from this DeepGraphCNN layer has 400 rows.

The output of the DeepGraphCNN layer is fed into a convolutional layer with 128 channels. Since this layer primarily synthesizes data from the first layer, its kernel size and stride are set equal to the sum of the DGCNN layer channels. Subsequently, a MaxPool and a Dropout layer are applied. Following the data synthesis from the first layer, a new Conv1D layer is introduced as a feature extractor. The network output is then flattened to a single dimension for processing

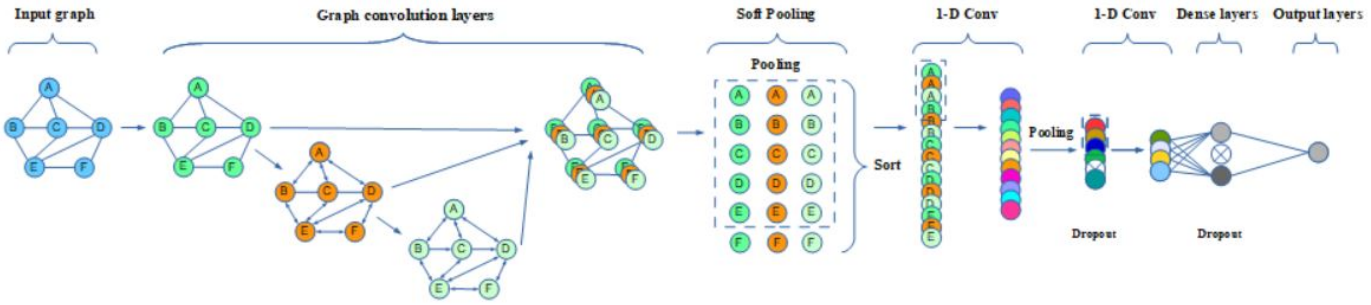


Fig. 1. DGCNN V1 model architecture.

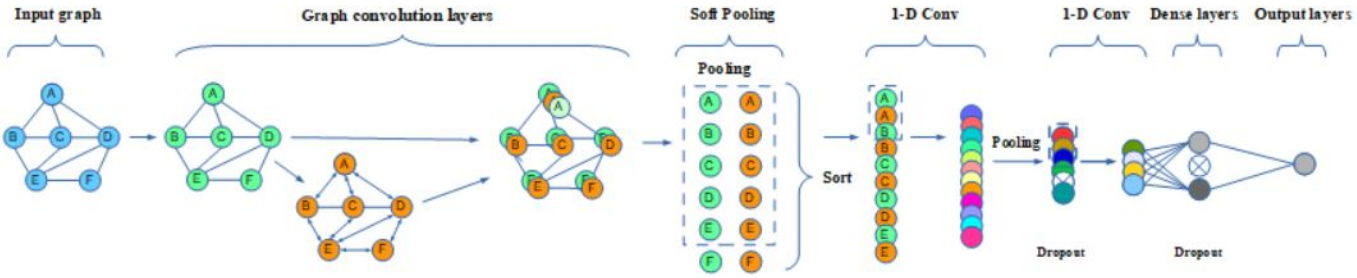


Fig. 2. DGCNN V2 model architecture.

by two consecutive Dense layers. These layers employ ReLU and Sigmoid activation functions, respectively (Table I).

TABLE I. DETAILED DGCNN V1 NETWORK

Deep Graph CNN		
Layer	Configuration	Output
DGCNN	k: 400 layer size: [256, 256, 256, 1] activations: [tanh,tanh,tanh,tanh]	256x256x256
CNN		
Conv1D	kernel: 769 stride: 769 chanel: 128	400x128
MaxPool1D	pool size: 2	200x128
Dropout	rate: 0.1	200x128
Conv1D	kernel: 50 stride: 1 chanel: 256	150x128
Flatten	In: 150x128	19200
Dense	units:512, ReLU	512
Dropout	rate: 0.1	512
Dense	units:1, Sigmoid	1

The second proposed model, DGCNNv2 as presented in Fig. 2, shares a similar structure with DGCNNv1. However, all settings are adjusted to accommodate the training dataset that has been pre-processed with Random Under-sampling to achieve a 5:100 class ratio. Under-sampling serves the same purpose as over-sampling but instead of replicating the minority class, it removes samples from the majority class to achieve the desired ratio (Table II).

Considering the reduced size and significantly higher negative label rate of the under-sampled training data, DGCNNv2 implements several modifications to prevent overfitting and decrease training costs. To mitigate the risk of overfitting, where nearly all predictions become negative, one child GCN layer is removed from the DeepGraphCNN layer. Additionally, the number of output rows in the DeepGraphCNN layer is reduced to 135, and the size of the GCN layers is lowered to 128.

Furthermore, DGCNNv2 adopts a convergent architecture for the data synthesis layer, where the size of each hidden layer progressively decreases. Additionally, the Dropout rate is increased from 10% to 20% to further prevent overfitting.

C. Features and Fusion

We utilize four types of features extracted from patients' medical reports:

- Subject's body parameters: These are mainly single, linear values representing various physiological measurements.
- Workplace information: Categorical data describing the patient's work environment and potential occupational hazards.
- Habits: Categorical data capturing the patient's lifestyle choices and habits.
- Disease symptoms: Both visible and invisible symptoms reported by the patient, classified as categorical data.

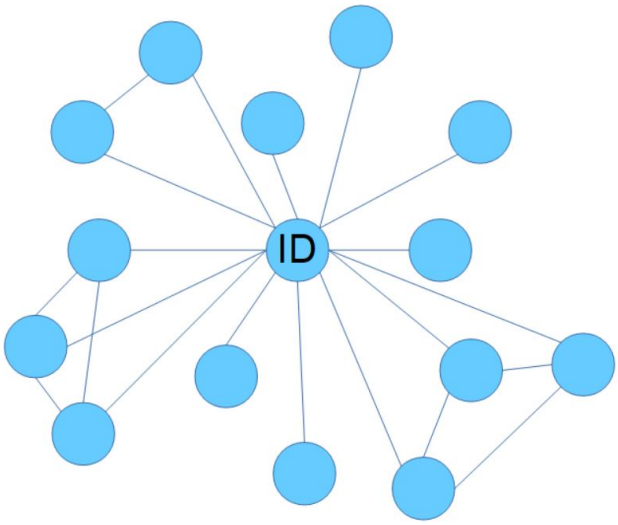


Fig. 3. Relationship graph architecture.

These features are collected from hospital-generated medical reports, ensuring data consistency and quality. While body parameters are primarily numerical, the remaining features are categorized, allowing them to adapt their scope based on the number of unique values encountered.

To effectively capture the relationships between these features, we reorganize the patient data into an adjacency matrix, represented as a relational graph. Each graph comprises nodes and edges corresponding to individual data points and their relationships. The unique patient ID serves as the root node, distinguishing each subject. This root node connects to the four aforementioned feature categories.

Furthermore, we define edges between relevant features to capture intricate relationships. For example, if a patient reports chest pain, we also have information about the pain level, location, duration, and contributing factors. By establishing edges between these nodes, we enable information propagation within the graph, allowing each node to leverage the information contained within its neighbors. The resulting graph structure is similar to the illustration in Fig. 3.

These graphs are then utilized for feature extraction. Spatial graph convolutions are applied to extract vertex features, followed by a SortPooling layer to arrange them in a consistent order. This process generates a sorted graph representation with a fixed size, enabling Convolutional Neural Networks to efficiently process and learn from the data in a consistent manner [16].

TABLE II. DETAILED DGCNN V2 NETWORK

Deep Graph CNN		
Layer	Configuration	Output
DGCNN	k: 135 layer size: [128, 128, 1] activations: [tanh,tanh,tanh]	128x128
CNN		
Conv1D	kernel: 257 stride: 257 chanel: 256	135x256
MaxPool1D	pool size: 2	67x256
Dropout	rate: 0.2	67x256
Conv1D	kernel: 50 stride: 1 chanel: 128	18x128
Flatten	In: 18 x 128	2304
Dense	units:64, ReLU	64
Dropout	rate: 0.2	64
Dense	units:1, Sigmoid	1

TABLE III. DEMOGRAPHIC INFORMATION, PERCEIVED SYMPTOMS SYMPTOMS AND PPE OF STUDIED SUBJECTS

	Healthy	Positive
Demographic information		
Age (Average)	42.6	52.3
Gender (Male/Female)	6379/1440	173/33
Seniority (year)	10.7	20.6
Perceived symptoms		
Cough	1700	149
Sputum	1638	150
Dyspnea	766	144
Chest pain	845	151
Nasal discharge	685	39
Hoarseness	563	36
Wheezing	262	21
Tiredness	982	105
Weight loss	358	30
Personal Protective Equipment (PPE)		
Helmet (Yes/No)	6275/1544	184/27
Boots	6413/1406	167/44
Gauze mask	7553/266	207/4
Gloves	6396/1423	155/56
Goggles	3184/4635	117/94
Employment insurance	6930/889	188/23

IV. EXPERIMENT AND DISCUSSION

A. Dataset

The dataset utilized in this study consists of 8,030 samples. Each sample includes a binary output class indicating whether the subject is healthy or diagnosed with an occupational disease. The dataset exhibits a significant class imbalance, with 7,819 negative (healthy) samples and 211 positive (ill) samples, resulting in an output data ratio of 37:1. To address this imbalance, we employed appropriate data pre-processing techniques for each model, as detailed in the corresponding experiments.

Prior to model training, the dataset was split into two subsets: 70% for training and 30% for validation. The entire original dataset is used for testing to provide a comprehensive evaluation of the models' performance. Table III presents detailed demographic and clinical information about the subjects included in the study.

B. Pre-processing

Given the heterogeneity and imbalance of medical data, thorough pre-processing is crucial for such studies. To address these challenges, we implemented a comprehensive pipeline

focusing on data cleaning, missing value handling, and class imbalance correction.

Firstly, to ensure optimal training performance, we cleaned the dataset by removing 39 empty columns (0.17%) lacking informative value. For remaining fields with missing data, we employed appropriate imputation techniques based on data type and context, preserving valuable information while minimizing bias. Notably, we retained encoded fields with numerous unique values, acknowledging their potential noise but opting for alternative mitigation strategies during training to capitalize on their valuable information.

Secondly, to address the class imbalance (3 positive to 112 negative samples), we employed the Condensed Nearest Neighbors [20], [21], [22] under-sampling technique from Imbalanced-learn. This approach strategically removed redundant majority class samples while preserving all minority class data, resulting in a more balanced 3:7 ratio. This balanced dataset facilitated fair model evaluation and prevented potential bias towards the dominant class, ensuring accurate and reliable predictions for both positive and negative cases.

Moreover, to enable a fair comparison with traditional machine learning models, we adapted our pre-processing pipeline to their specific needs. While DGCNNs handle diverse data formats, traditional models require homogeneous input. We therefore employed additional data cleaning steps, including imputing missing values with context-aware techniques and limiting the data to fields with less than 50% missing data to ensure sufficient information for traditional model training (Table IV).

TABLE IV. DETAILED GRAPH INFORMATION

Graph statistic	
Nodes (max)	147
Nodes (min)	88
Nodes (avg)	107.77
Edges (max)	172
Edges (min)	94
Edges (avg)	119.28
Graphs	8030

C. Experiment Setup

To comprehensively evaluate the proposed method and compare the effectiveness of the DGCNN models against other popular approaches (KNN, SVM, ANN, and LSTM), we conducted six distinct experiments detailed in Table V. Each experiment followed a three-stage pipeline:

- Re-sampling: Recognizing the inherent class imbalance in the dataset, as shown in Fig. 4 and 5, we employed targeted re-sampling techniques to ensure fair model evaluation. For KNN, SVM, ANN, and LSTM models, we utilized Condensed Nearest Neighbor (CNN) under-sampling from Imbalanced-learn, as illstrated in Fig. 7. This technique carefully selected minority class samples and strategically removed redundant majority class data, resulting in a balanced 3:7 ratio. For DGCNNv1 and v2, we opted for Random Under-sampling, maintaining all minority class samples while randomly eliminating a portion of the majority class to achieve a 5:100 ratio. This choice leveraged the DGCNNs' ability to handle imbalanced data more effectively due to their graph-based nature, as shown in Fig. 6.
- Training Model: Each model was trained with the re-sampled dataset using optimized hyperparameters determined through grid search. For DGCNNs, this included configuring graph convolutional layers, activation functions, and learning rates. The goal was to achieve optimal performance with minimal overfitting.
- Evaluating Output Model: We assessed the performance of each model using a set of relevant metrics including precision, recall, F1-score, and balanced accuracy. This provided a comprehensive picture of each model's effectiveness in identifying occupational disease cases, considering both positive and negative predictions.

TABLE V. SIX EXPERIMENTS WITH DIFFERENT INPUTS AND NETWORKS

Exp	Re-sample method	Train/test ratio	Model	Others
1	CNN Condensed Nearest Neighbour	5:5	KNN	k: 5
2	CNN Condensed Nearest Neighbour	5:5	SVM	gamma:1/109
3	CNN Condensed Nearest Neighbour	5:5	ANN	learn rate: 0.001 batch size: 64 epoch: 50
4	CNN Condensed Nearest Neighbour	5:5	LSTM	learn rate: 0.001 batch size: 64 epoch: 50
5	Random Under Sampler Ratio: 0.3	7:3	DGCNN V1	learn rate: 0.0005 batch size: 100 epoch: 100
6	Random Under Sampler Ratio: 0.05	7:3	DGCNN V2	learn rate: 0.001 batch size: 100 epoch: 150

This structured approach, coupled with specific re-sampling strategies tailored to each model type, allowed us to conduct a rigorous and fair evaluation of our proposed method compared to established tools. The results, presented in Table V and further analyzed in subsequent sections, reveal valuable insights into the effectiveness of DGCNNs for analyzing medical data with its inherent complexities..

The experiments leveraged the computational power of a 4 GB NVIDIA Quadro M2200 GPU and an Intel(R) 2.8 GHz Xeon(R) microprocessor, running TensorFlow 2.10.0 and StellarGraph Framework 1.2.1[23] under Python 3.9.12, to implement and train the various models. This framework enabled efficient execution of the DGCNN algorithms, while the powerful GPU-CPU combination facilitated smooth pre-processing and data analysis tasks.

We used the following parameters and techniques for training our models:

- The model was compiled using a binary cross-entropy loss function.
- For optimization, an Adam optimizer was employed with $\beta 1 = 0.9$, $\beta 2 = 0.999$, and $\epsilon =$ 1e-07. The initial learning rate was adjusted to optimize each model.
- The batch size for the ANN and LSTM models was set at 64 to minimize the cost function. For all DGCNN

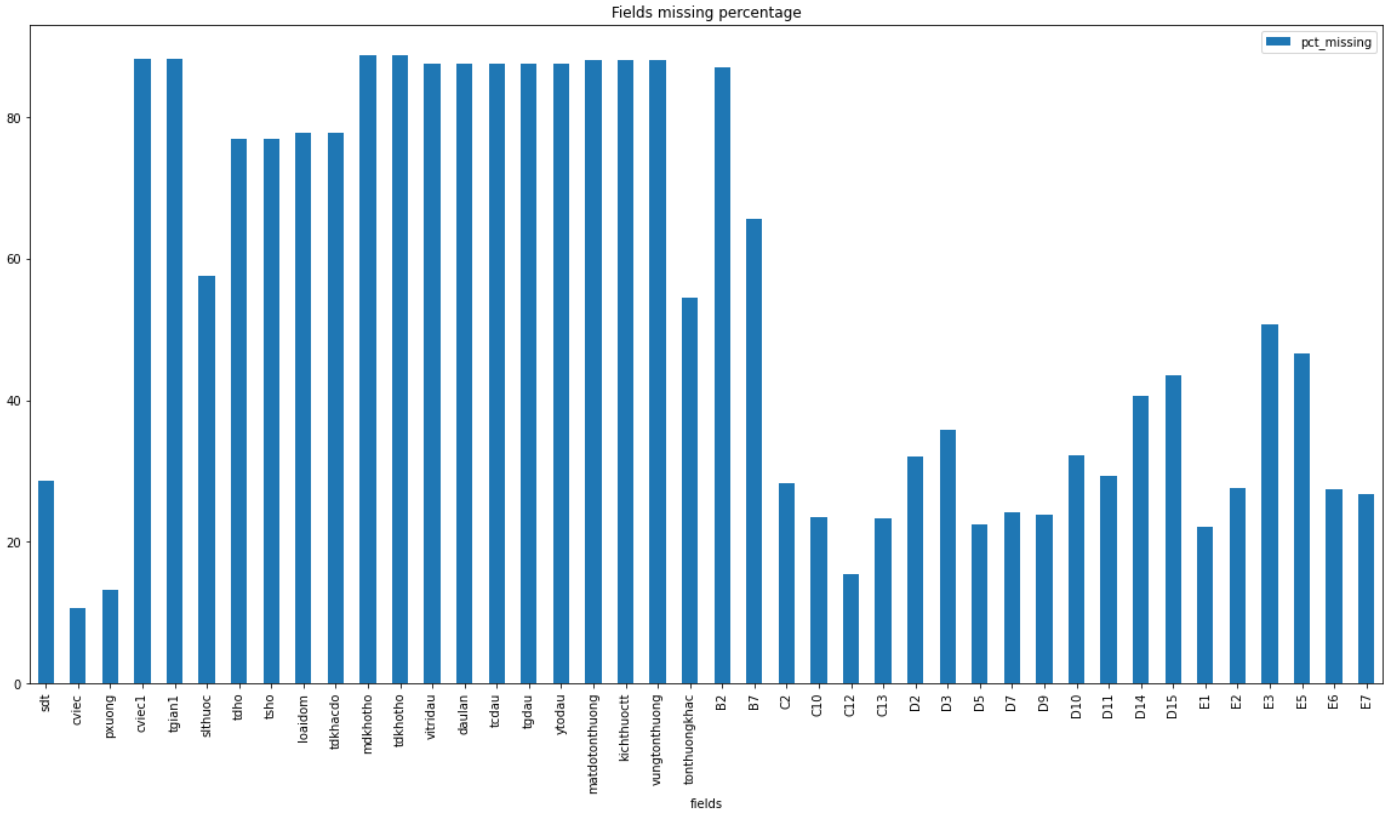


Fig. 4. Missing percentage of data field(s).

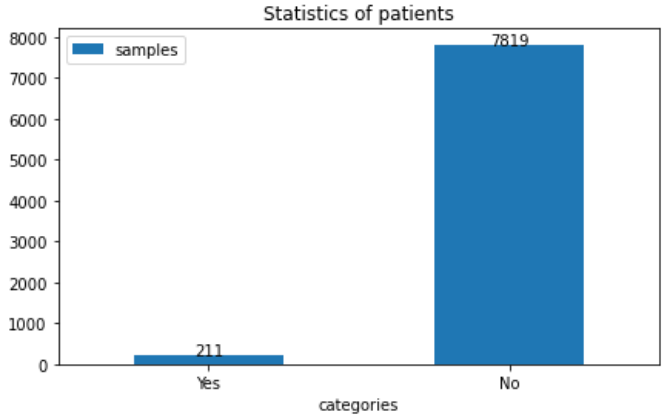


Fig. 5. Statistics of original data set output labels.

networks, a minibatch size of 100 was applied to enhance training performance.

- Since the models were trained without a large number of epochs, the early stopping technique was not implemented in the training process.
- To estimate the efficiency of each model fairly, the prediction results on all 8030 samples from the original dataset were used to calculate the evaluation metrics.

In this study, the performance metrics employed to evaluate the experimental results include accuracy, loss, F1 score, precision, recall, and the confusion matrix. The models will be applied to predict outcomes on the original dataset to ensure a fair evaluation. These metrics are calculated using the following formulas:

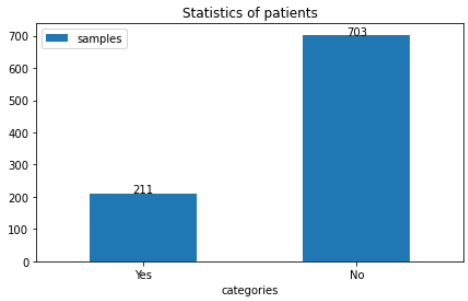
$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

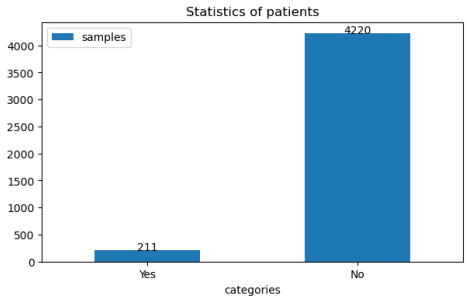
$$Recall = \frac{TP}{TP + FN}$$

$$F_1 = \sum_{i}^{classes} 2 \times \frac{class_i}{totalsamples} \times \frac{Precision_i \times Recall_i}{Precision_i + Recall_i}$$

where TP is the true positive (number of samples *correctly* predicted as "positive"), TN is the true negative (number of samples *correctly* predicted as "negative"), FP is the false positive (number of samples *wrongly* predicted as "positive") and FN is false negative (number of samples *wrongly* predicted as "negative").



(a) DGCNN V1 network.



(b) DGCNN V2 network.

Fig. 6. Under-sampling for DGCNN network.

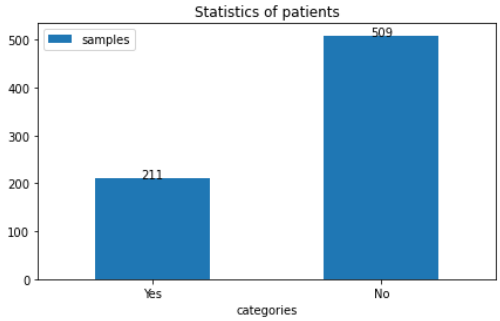


Fig. 7. Condensed nearest neighbors re-sampling.

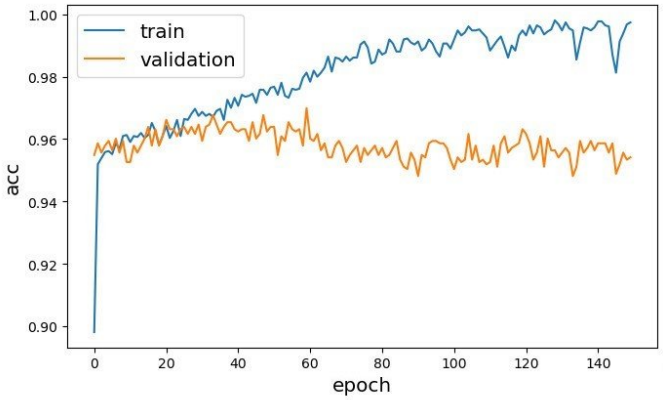


Fig. 8. Training progress.

TABLE VI. EXPERIMENT RESULTS

Exp	Method	Precision	Recall	F1 Score	ACC
1	KNN	18.8%	70.14%	29.66%	91.26%
2	SVM	39.4%	82%	53.23%	96.21%
3	ANN	58.14%	83%	68.36%	97.98%
4	LSTM	62.4%	78.67%	69.6%	98.19%
5	DGCNN V1	43.6%	35.8%	39.3%	96.46%
6	**DGCNN V2**	**78%**	**72.89%**	**75.23%**	**98.66%**

D. Results and Discussion

The results of the six methods that we have discussed are presented in Table VI. This table illustrates that all six models are capable of detecting occupational diseases using their respective inputs and networks. Among these, the sixth experiment exhibits the best performance, achieving an accuracy of 98.66%, a loss of 1.65%, a recall of 72.89%, a precision of 78%, and an F1 score of 75.23%. The F1 score, precision, and recall are not as high as the accuracy, primarily due to noise arising from elements in the dataset that contain multiple classification values. The amount of noise is proportional to the size of the input data. Furthermore, the input samples used for prediction are imbalanced; therefore, a high rate of correct predictions does not necessarily indicate that every class has a similar rate of correct prediction. The fact that recall, precision, and F1 score are almost equal suggests that our model's predictions are more balanced and has accurately diagnosed many patients.

The progress of training and Mean Receiver Operating Characteristic (ROC) curves for the sixth model are displayed in Fig. 9 and 8, respectively. They indicate that our DGCNN V2 model architecture, as detailed in Table II, achieved a high accracy and Area Under the Curve (AUC) of 96.22%. This demonstrates the model's strong performance in classifying negative and positive samples.

Table VII showcases a comprehensive classification comparison between sick patients and healthy individuals from our fourth experiment. In this experiment, the LSTM model outperformed other conventional methods, demonstrating effective detection of diseased patients within the overall patient population in the dataset. Yet, our DGCNN V2 model, as depicted in Table VIII, exhibits even greater effectiveness, particularly in the context of the sixth experiment. This model excels in handling heterogeneous datasets. For the negative class, precision, recall, and F1 scores are uniformly high at approximately 99.33%. In contrast, the positive class yields scores of 77.73% for precision, 72.9% for recall, and 75.23% for the F1 score. The macro averages are calculated as 88.47% for precision, 86.14% for recall, and 87.27% for the F1 score, with the weighted averages hovering around 98.63%. Overall, our model attained an impressive 98.66% accuracy across the entire dataset. Support numbers stand at 225 for the occupational diseases category and 7805 for the healthy category, contributing to a total of 8030 for each accuracy, macro average, and weighted average metric. A comparison

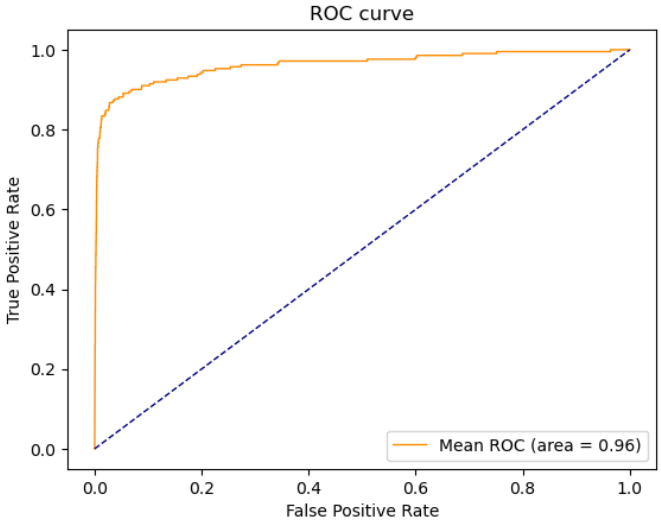


Fig. 9. Mean ROC curves for the classifiers on the test set.

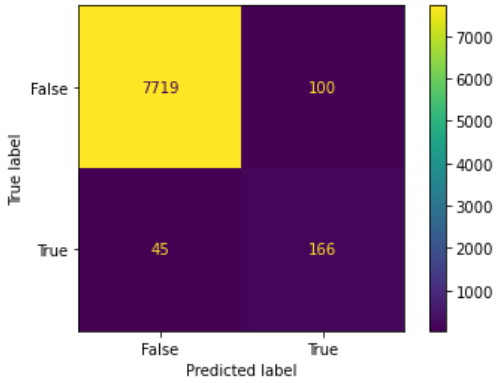


Fig. 10. LSTM model prediction confusion matrix.

of the confusion matrices from the LSTM outputs and the DGCNN prediction results, as shown in Fig. 11 clearly demonstrates the superior performance of our network over traditional methodologies (Fig. 10).

TABLE VII. EXPERIMENT 4 - CLASSIFICATION PERFORMANCE

	Precision	Recall	F1 Score	Support
class No	99%	99%	99%	7764
class Yes	79%	62%	70%	266
Accuracy			98%	8030
Macro avg	89%	81%	84%	8030
Weighted avg	98%	98%	97%	8030

V. CONCLUSION

In this study, we sought to enhance occupational disease detection performance. Our proposed approach utilizes a relationship graph to store and analyze body indicators alongside information about patients' working environments and the interrelations of these parameters. We empirically validated our method on a collected dataset, demonstrating its superior

TABLE VIII. EXPERIMENT 6 - CLASSIFICATION PERFORMANCE

	Precision	Recall	F1 Score	Support
class No	99%	99%	99%	7805
class Yes	78%	73%	75%	225
Accuracy			99%	8030
Macro avg	88%	86%	87%	8030
Weighted avg	99%	99%	99%	8030

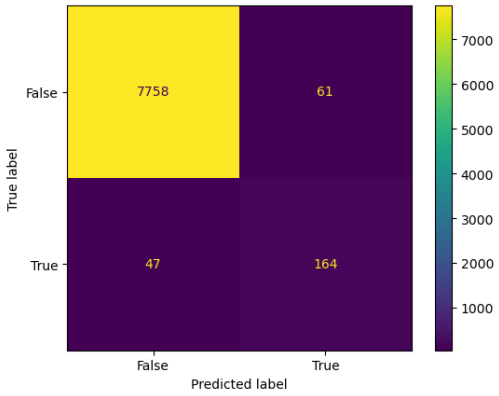


Fig. 11. DGCNN model prediction confusion matrix.

efficiency with an accuracy of 98.66%, an F1 Score of 75.23%, and a ROC (Receiver Operating Characteristic) of 96.22%. Additionally, when applied to a commonly used stroke prediction dataset from Kaggle, our method achieved remarkable results: an accuracy of 99.69%, an F1 Score of 96.9%, and a perfect ROC of 100%. These outcomes not only outperformed other state-of-the-art methods but also surpassed previous solutions as indicated in various studies [24], [4], [25]. The results affirm the Deep Graph CNN network's suitability for handling heterogeneous data, which is crucial for accurately diagnosing diseases.

Looking ahead, our future work will focus on developing an API that integrates this proposed method. This will enable medical websites to utilize our approach for diagnosing occupational diseases, leveraging user-provided occupational information.

REFERENCES

[1] M. M. Islam, M. R. Haque, H. Iqbal, M. M. Hasan, M. Hasan, and M. N. Kabir, "Breast cancer prediction: a comparative study using machine learning techniques," *SN Computer Science*, vol. 1, pp. 1–14, 2020.

[2] V. Ramalingam, A. Dandapath, and M. K. Raja, "Heart disease prediction using machine learning techniques: a survey," *International Journal of Engineering & Technology*, vol. 7, no. 2.8, pp. 684–687, 2018.

[3] K. Pingale, S. Surwase, V. Kulkarni, S. Sarage, and A. Karve, "Disease prediction using machine learning," *International Research Journal of Engineering and Technology (IRJET)*, vol. 6, no. 12, pp. 831–833, 2019.

[4] G. Sailasya and G. L. A. Kumari, "Analyzing the performance of stroke prediction using ml classification algorithms," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 6, 2021.

[5] S. Parisot, S. I. Ktena, E. Ferrante, M. Lee, R. G. Moreno, B. Glocker, and D. Rueckert, "Spectral graph convolutions for population-based disease prediction," in *Medical Image Computing and Computer Assisted Intervention- MICCAI 2017: 20th International Conference, Quebec City, QC, Canada, September 11-13, 2017, Proceedings, Part III 20*. Springer, 2017, pp. 177–185.

[6] M. Zhang, Z. Cui, M. Neumann, and Y. Chen, "An end-to-end deep learning architecture for graph classification," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[7] A. M. Barhoom, A. Almasri, B. S. Abu-Nasser, and S. S. Abu-Naser, "Prediction of heart disease using a collection of machine and deep learning algorithms," 2022.

[8] M. Ismail, V. H. Vardhan, V. A. Mounika, and K. S. Padmini, "An effective heart disease prediction method using artificial neural network," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 8, pp. 1529–1532, 2019.

[9] J. Rashid, S. Batool, J. Kim, M. Wasif Nisar, A. Hussain, S. Juneja, and R. Kushwaha, "An augmented artificial intelligence approach for chronic diseases prediction," *Frontiers in Public Health*, vol. 10, p. 860396, 2022.

[10] V. Sudha and D. Kumar, "Hybrid cnn and lstm network for heart disease prediction," *SN Computer Science*, vol. 4, no. 2, p. 172, 2023.

[11] S. Zhang, H. Tong, J. Xu, and R. Maciejewski, "Graph convolutional networks: a comprehensive review," *Computational Social Networks*, vol. 6, no. 1, pp. 1–23, 2019.

[12] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger, "Simplifying graph convolutional networks," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 6861–6871. [Online]. Available: https://proceedings.mlr.press/v97/wu19e.html

[13] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[14] P. Xuan, S. Pan, T. Zhang, Y. Liu, and H. Sun, "Graph convolutional network and convolutional neural network based method for predicting lncrna-disease associations," *Cells*, vol. 8, no. 9, p. 1012, 2019.

[15] H. Lu and S. Uddin, "Disease prediction using graph machine learning based on electronic health data: A review of approaches and trends," in *Healthcare*, vol. 11, no. 7. MDPI, 2023, p. 1031.

[16] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[17] J. Prusa, T. M. Khoshgoftaar, D. J. Dittman, and A. Napolitano, "Using random undersampling to alleviate class imbalance on tweet sentiment data," in *2015 IEEE international conference on information reuse and integration*. IEEE, 2015, pp. 197–202.

[18] Z. Zheng, Y. Cai, and Y. Li, "Oversampling method for imbalanced classification," *Computing and Informatics*, vol. 34, no. 5, pp. 1017–1037, 2015.

[19] M. Khushi, K. Shaukat, T. M. Alam, I. A. Hameed, S. Uddin, S. Luo, X. Yang, and M. C. Reyes, "A comparative performance analysis of data resampling methods on imbalance medical data," *IEEE Access*, vol. 9, pp. 109 960–109 975, 2021.

[20] G. Lemaître, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 559–563, 2017.

[21] K. C. Gowda and T. Ravi, "A modified condensed nearest neighbour rule using the symbolic approach," in *Proceedings of ANZIIS'94-Australian New Zealnd Intelligent Information Systems Conference*. IEEE, 1994, pp. 174–178.

[22] N. G. Siddappa and T. Kampalappa, "Adaptive condensed nearest neighbor for imbalance data classification," *International Journal of Intelligent Engineering and Systems*, vol. 12, no. 2, pp. 104–113, 2019.

[23] C. Data61, "Stellargraph machine learning library," https://github.com/stellargraph/stellargraph, 2018.

[24] N. Biswas, K. M. M. Uddin, S. T. Rikta, and S. K. Dey, "A comparative analysis of machine learning classifiers for stroke prediction: A predictive analytics approach," *Healthcare Analytics*, vol. 2, p. 100116, 2022.

[25] M. Ashrafuzzaman, S. Saha, and K. Nur, "Prediction of stroke disease using deep cnn based approach," *Journal of Advances in Information Technology Vol*, vol. 13, no. 6, 2022.