BMC Medical Informatics and Decision Making

RESEARCH

Open Access

Classification of lung cancer severity using gene expression data based on deep learning

Ali Bou Nassif[1*], Nour Ayman Abujabal[1] and Aya Alchikh Omar[1]

Abstract

Lung cancer is one of the most prevalent diseases affecting people and is a main factor in the rising death rate. Recently, Machine Learning (ML) and Deep Learning (DL) techniques have been utilized to detect and classify various types of cancer, including lung cancer. In this research, a DL model, specifically a Convolutional Neural Network (CNN), is proposed to classify lung cancer stages for two types of lung cancer (LUAD and LUSC) using a gene dataset. Evaluating and validating the performance of the proposed model required addressing some common challenges in gene datasets, such as class imbalance and overfitting, due to the low number of samples and the high number of features. These issues were mitigated by deeply analyzing the gene dataset and lung cancer stages from a medical perspective, along with extensive research and experiments. As a result, the optimized CNN model using F-test feature selection method, achieved high classification accuracies of approximately 93.94% for LUAD and 88.42% for LUSC.

Keywords Lung cancer, LUAD, LUSC, Gene expression, Classification, Deep learning, Feature selection

Introduction

Lung cancer is defined as the uncontrolled growth and undesired spread of a sample of cells within the lungs. It kills more people than colon cancer, breast cancer, and prostate cancer all combined, making it the leading cancer killer in the United States [1]. Around 20% of lung cancer cases are caused by non-tobacco factors, such as genetics. Recent statistics indicate that less than 20% of patients diagnosed with lung cancer survive for more than five years. There are around 225,000 reported cases, 150,000 deaths, and more than 12 billion dollars in total healthcare costs in the United States alone each year [1]. Globally, 2.1 million cases and 1.8 million deaths were reported in 2018 as lung cancer, which is around 18.4% of all reported cancer cases that year [2, 3].

Lung cancer is classified into two major types, the first is Small Cell Lung Cancer (SCLC), while the other is Non-small Cell Lung Cancer (NSCLC) [4]. A sample of cells where lung cancer is suspected is called a "Lung Biopsy". It is important not to confuse the "stages" of lung cancer with the "types" of lung cancer. Under the umbrella of NSCLC, there are tens of different subtypes of lung cancer. The most common subtypes are Lung Adenocarcinoma (LUAD) and Lung Squamous Cell Carcinoma (LUSC) [5]. However, modern research indicates that these two subtypes are supposed to be treated and classified as two different major types of lung cancer.

Different types of lung cancer are caused by different factors. Among these factors, the most common contributors to developing lung cancer are an unhealthy environment and lifestyle. Air pollution and toxic gases are all forms of unhealthy environment. Given that most people who live in cities breathe unclean air and, in some

*Correspondence:
Ali Bou Nassif
anassif@sharjah.ac.ae
[1]Department of Computer Engineering, College of Computing and Informatics, University of Sharjah, P.O Box: 27272, Sharjah, UAE

areas, breathe carbon emissions from factories in industrial areas, it is reasonable to say that early detection of lung cancer is crucial. However, many people with lung cancer do not exhibit symptoms, which makes it more challenging to detect the cancer at early stages [6, 7]. When symptoms do appear, they may include a persistent cough, unusual wheezing, coughing up blood, difficulty breathing, and weight loss. Over the past decades, the main and most common method for detecting lung cancer has been Computerized Tomography (CT), commonly known as a CT scan. Other diagnostic techniques include Chest Radiography (CXR), Positron Emission Tomography (PET), and Magnetic Resonance Imaging (MRI) [8]. Despite these advanced diagnostic methods, lung cancer is often not detected until it has progressed to an advanced stage [9].

The diagnosis of lung cancer is often performed at a late stage due to poor prediction and absence of symptoms in most cases. The emerging field of high-throughput sequencing has revolutionized the way of approaching lung cancer diagnosis. High-throughput sequencing (gene sequencing) allows researchers to explore the genomic evolution of premalignant conditions through various stages, including tumorigenesis [10]. This insight has led to significant advancements in our understanding of the molecular, immunological, and cellular characteristics of lung cancer at different stages, especially in the early stages. Today, these studies have gone beyond conventional methods, offering more reliable approaches for early detection and potential cures.

The evolution in technology over the past two decades, starting from computer development and medical equipment to the utilization of smart technologies and big data, is playing a key role in improving lung cancer detection [11]. As time passes, more and more techniques appear from the evolution of new technologies, setting up higher standards and raising hope. One of these techniques that have dominated the field of studies and research is the utilization and development of Machine Learning (ML) algorithms. ML algorithms have enabled different approaches to predict lung cancer, including Support Vector Machines (SVM) [12], K-nearest neighbor (KNN) [13], and Naïve Bayes.

Gene expression datasets pose several significant challenges when detecting lung cancer. One common challenge is the size of the dataset. Many gene expression datasets may be relatively small, limiting the diversity and volume of data available for analysis. In addition, high dimensionality is another challenge, as these datasets often contain a vast number of genes or features, making it complex to process and extract meaningful information. Moreover, data imbalance can be an issue, where the number of cancer samples is significantly smaller than non-cancer samples, potentially leading to biased results [14]. Most of the genes are not involved in lung cancer and the goal is to identify, isolate, and eliminate the irrelevant genes. To achieve this, a well-developed ML or DL method that can differentiate relevant from irrelevant genes is required.

The primary contribution of this research is the development of a Deep Learning (DL) model that effectively learns from sequential data and captures complex relationships within datasets. Specifically, the study proposes an optimized Convolutional Neural Network (CNN) model for classifying the severity of lung cancer using gene expression data. A significant aspect of this research is its approach to the common problem of imbalanced class distribution in gene datasets. This issue is addressed through a detailed analysis of the dataset and the stages of lung cancer from a medical standpoint by transforming dataset labels into a binary format and applying a feature selection method to identify the most relevant genes for classification. Additionally, the performance of the developed DL model in predicting lung cancer severity is evaluated and validated by comparing it with different ML models and existing research studies. The workflow of the proposed methodology is depicted in Fig. 1.

The contributions of this paper can be illustrated as follows:

- Developing a DL-based classification method using gene expression data for lung cancer severity classification.
- Identifying what is the most effective model to classify the severity of two types of lung cancer using gene expression data.
- Addressing common problems in gene datasets, such as imbalanced samples and a high number of features, by proposing effective solutions.
- Enhancing the model efficiency by utilizing feature selection methods.
- Tuning the hyperparameters of the proposed DL model to investigate the optimal parameters values to provide the highest classification results.

The research addresses the following research questions:

1. Can the severity of lung cancer be detected by ML using gene expression data?
2. What is the most effective ML or DL model to detect the severity of lung cancer?
3. How does feature selection affect the performance of ML models on genetic datasets?
4. How does tuning the hyperparameters affect the performance of a DL model?

The remaining sections of this research are organized as follows: A brief technological foundation concepts are

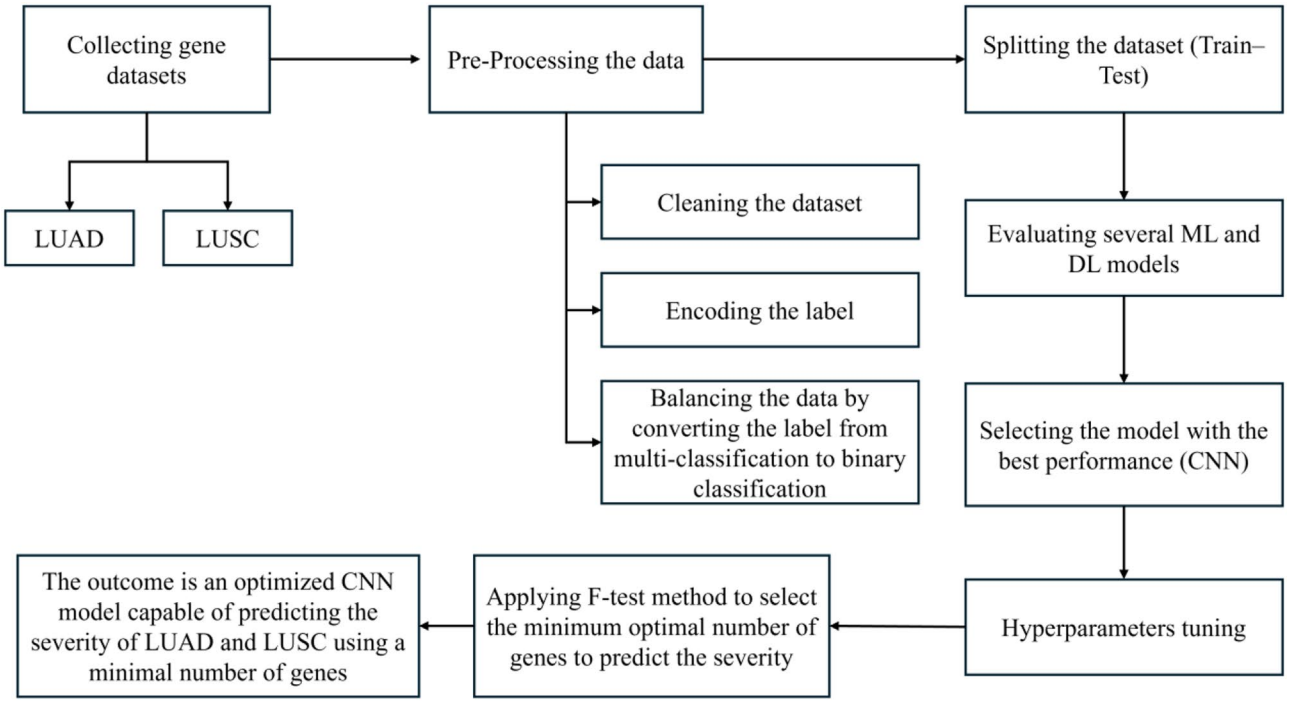

Fig. 1 The proposed workflow diagram for predicting the severity of lung cancer using gene expression data

covered in Section "Technical background", then Section "Literature review" discusses the recent ML methods in lung cancer detection performed by other researchers, while the methods employed in our work are explained in Section "Methodology". After that, the details of the model design are illustrated in Section "Model design", followed by the results and analysis in Section "Results and discussion", and finally, the conclusion and future work in Section "Conclusion".

Technical background

This section discusses the technical concepts that will be used in this research. It will cover a brief background about different ML models that will be used to classify the severity of lung cancer, the CNN model, the F-test feature selection method, and the employed performance metrics.

A) *Adaboost*: It is an example of the ensemble methods. It is a simple procedure for enhancing weak classification models, which improves data classification performance through recurrent training. Learning the training samples produces the initial weak classifier, and then merging the incorrect samples with the untrained data produces a new training sample. Therefore, the second weak classifier is generated. After that, by mixing the incorrect sample with the untrained data, another novel training sample is produced, which may then be trained to generate a third weak classifier. Consequently, by using this process repeatedly for multiple iterations, an improved and robust classifier can be established. The AdaBoost approach uses various sample weights to boost the percentage of correct classifications [15].

B) *Support Vector Machine (SVM)*: It is a machine learning technique that is utilized in classification. It seeks to establish a decision boundary between two different classes such that output may be predicted using one or more feature vectors [16]. The decision boundary, also called the hyperplane, is oriented to be as far as possible away from the closest data points (support vectors) from each class. The simplest SVM model utilizes two hyperplanes to linearly separate the data, with the distance between these two hyperplanes being maximized to reduce error [17].

C) *K-Nearest Neighbor (KNN)*: It is a supervised learning approach that can be utilized to classify data. It is a non-parametric classification technique that divides the dataset's samples into groups (classes) based on whose neighbors' labels are most similar. The size of the classes is determined by K-values. There are two possible ways for classification, either compare the testing sample with its neighbor, or it will depend on how close the testing sample is to its neighbors. The test sample will then be classified depending on most of the samples [18].

D) *Random Forest (RF)*: It is a supervised learning method that can be utilized in data prediction and

classification applications. However, it is mostly employed to solve classification problems. A forest is just a collection of different trees. Consequently, boosting the number of trees will strengthen the forest and make it more robust. Like this, the random forest technique builds decision trees from samples of data, extracts predictions from each, and then asks for a vote on which prediction is the best. Since it averages the results to reduce the overfitting of the data, this approach is viewed as an ensemble method that outperforms the single decision tree method [19].

E) *Convolutional Neural Network (CNN)*: It is a deep learning model that learns from data directly. There are several applications of CNN models, including recognizing objects and categories in images by determining patterns in the images. In addition, a CNN model is utilized to learn the multiple layers of kernel filters along with learning the classifier's weights. The CNN architecture is composed of three main layers: the convolution layer, the pooling layer, and the fully connected layer. The Convolution layer is responsible for feature extraction, while the pooling layer is used for data size reduction. Finally, the fully connected layer is responsible for data classification. Figure 2 presents the architecture of the CNN models [20].

F) *F-test*: F-test is a feature selection method that is used to evaluate the averages of several groups mathematically. Each feature is scored and ranked based on its relationship to the label, with the highest-scoring feature being selected. The number of attributes and the F-ratio can be obtained using ANOVA, where the F-ratio value indicates the degree of class separation. The F-ratio is calculated as the variance between classes divided by the variance within classes. The score of each attribute is determined using Eq. 1, where $\bar{X}_i$ is the mean of the class, $\bar{X}$ is the mean of the attribute, n_i is the frequency with which class i appears in the set, and k is the number of classes [22].

$$\sigma_{cl}^2 = \frac{\sum \left(\bar{X}_i - \bar{X}\right)^2 n_i}{(k-1)} \tag{1}$$

F) Performance Metrics

G) Various performance metrics are employed to evaluate ML models in this research, which are:

i. **Accuracy**: It is the number of positively and negatively classified samples over the total number of samples [23]. The accuracy can be calculated using the formula:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{TN} + \text{FP}} \tag{2}$$

ii. **Confusion matrix:** This is one of the evaluation methods that includes four main values to measure the model performance in classifying negative and positive samples, which are represented by true positive (TP), true negative (TN), false negative (FN), and false positive (FP). After that, these classification results are used in measuring the following metrics [24]:

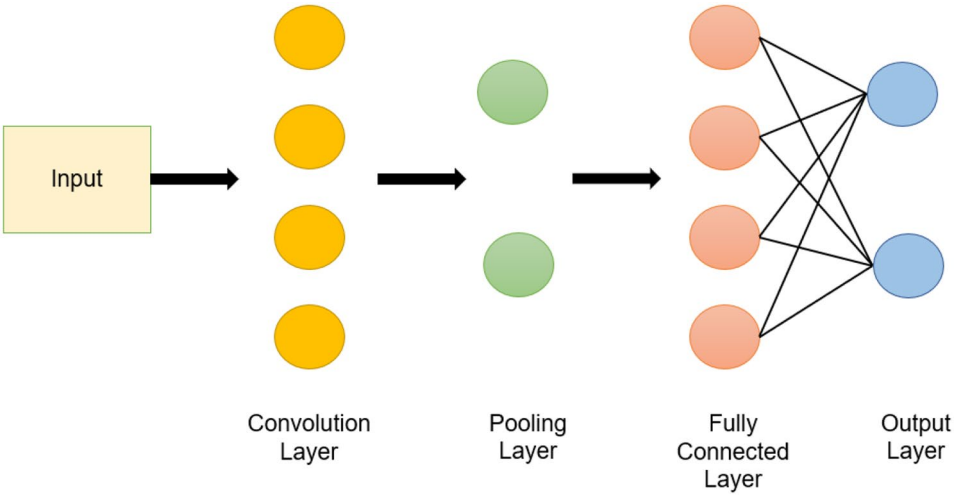

Fig. 2 CNN model architecture [21]

- True Positive Rate (TPR): It refers to the number of samples that are classified as positive correctly over all the number of samples that are actually positive.

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{Actual\,Positvie}} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{3}$$

- False Positive Rate (FPR): It refers to the number of samples that are classified as positive incorrectly over all the number of samples that are actually negative.

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{Actual\,Negative}} = \frac{\mathrm{TP}}{\mathrm{TN} + \mathrm{FP}} \tag{4}$$

- False Negative Rate (FNR): It refers to the number of samples that are classified as negative incorrectly over all the samples that are actually positive.

$$\mathrm{FNR} = \frac{\mathrm{FN}}{\mathrm{FN} + \mathrm{TP}} \tag{5}$$

- True Negative Rate (TNR): It refers to the samples that are classified as negative correctly over all the samples that are actually negative.

$$\mathrm{TNR} = \frac{\mathrm{TP}}{\mathrm{TN} + \mathrm{FP}} \tag{6}$$

iii. **Precision:** It refers to the model's ability to predict and classify the samples positively.

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{7}$$

iv. **Recall:** It refers to the model's ability to detect the class label correctly.

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{8}$$

v. **F-score**: It refers to the weighted average accuracy of the model.

$$\mathrm{F-score} = \frac{2 \times \mathrm{precision} \times \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} \tag{9}$$

vi. **Area Under the Curve (AUC):** It measures a model's ability to distinguish between classes by evaluating the trade-off between the TPR and the FPR across different thresholds.

vii. **Training Time:** It refers to the required time to build and train the model.

Literature review

This section reviews published studies on lung cancer classification using gene expression data. The review has identified two main categories of classification methods: classical ML models and DL models, both of which are examined in the following discussion.

A) *Classical Machine Learning Models*

In [25], the authors developed a DL-based multi-model ensemble method by utilizing k-nearest-neighbor, SVM, decision trees, random forests, and gradient-boosted decision trees in the classification stage. Then, the neural network is applied to ensemble the outputs of the five classical ML models. In addition, feature selection is implemented by using the DESeq method, and the model is designed and built by applying 4-fold cross-validation. The proposed model is evaluated using three gene expression datasets for three types of cancer: LUAD, Stomach Adenocarcinoma (STAD), and Breast Invasive Carcinoma (BRCA). The results of the proposed model achieved high precision values for the three types 99.4%, 99.5%, and 99.6 for LUAD, STAD, and BRCA, respectively.

Authors in [26] used SVM and random forest learning models on a gene dataset that includes two types of lung cancer, LUAD and LUSC. The evaluations of the models are performed using the Monte Carlo feature selection method (MCFS). Basically, the authors focused on using SVM to classify the type of lung cancer as LUAD or LUSC using gene expression data. There are some significant genes that are extensively analyzed for the differentiation between the two types, which are (CSTA, TP63, SERPINB13, CLCA2, BICD2, PERP, FAT2, BNC1, ATP11B, FAM83B, KRT5, PARD6G, PKP1). The performance evaluation is performed with randomly selected features and with informative features using MCFS, where 1100 and 260 features are selected randomly to

evaluate the SVM and random forest models, respectively. The accuracy results of randomly selected features showed 96.7 % for SVM and 87.1% for RF. While using the informative features, only 43 features were selected for both models, and the accuracy results were 75.4% and 77.2% for SVM and RF, respectively.

Authors in [27] implemented the light gradient boosting machine (Light-GBM) model to predict if there exists cancerous tissue in the lungs and then specified the lung cancer subtype as LUAD or LUSC using gene expression quantification (RNA-seq) dataset from The Cancer Genome Atlas (TCGA) project. The dataset consists of 598 LUAD samples and 553 LUSC samples with a total of 20,531 genes as features. The results show the accuracy and average AUC values are around 97.1%.

B) *Deep Learning Models*

Authors in [28] proposed a semi-supervised DL method called the Stacked Sparse Autoencoder (SSAE) to predict three different types of cancer, which are lung, stomach, and breast, based on the RNA sequence datasets. The SSAE consists of multiple single autoencoder layers. The middle layers are the extracted features of the dataset, which have the most significant features and low-dimensional data, and a neural network classifier is used in the output layer. The number of selected features to predict LUAD cancer is 1385 out of 20,532. Moreover, a comparison is conducted between the proposed model and other ML models, such as SVM, RF, NN, and SSAE. The proposed SSAE model outperforms the other ML models in predicting lung cancer (LUAD) with an accuracy of around 99.89%.

In [29], five types of tumor cancer are classified using gene expression data by applying CNN based on binary particle swarm optimization with a decision tree. The CNN model was developed and includes several prepossessing steps, which are converting the dataset to 2D images and data augmentation. The augmentation was used to increase the number of samples from 2086 to 5 times larger, with 971 total number of genes. The five types of cancer are breast invasive carcinoma, kidney renal clear well carcinoma, LUAD, LUSC, and Uterine Corpus Endometrial Carcinoma. The results are compared with other work that has implemented DL techniques without applying any optimization methods, where the overall accuracy is 95.20% with a gene dataset of 10,267 samples and 20,531 genes (features). The overall accuracy of classifying the five types of cancer is 96.20%, which outperforms the other work.

Authors in [30] have developed an immunotherapy-related gene signature to predict the stage of LUAD by analyzing the RNA-seq data and clinical data by implementing four ML classification models, which are SVM, naïve Bayes, random forest, and neural network-based DL, using RNA-seq dataset from TCGA. In addition, the random forest regression method was applied to figure out the association between gene mutations and the immune-related gene signature, where from 610 genes, 271 genes were immune-related genes. Moreover, the risk stratification capacity of each patient is measured by using five GEO validation datasets showing the risk stratification capacity of the immune-related gene signature for disease predictions. All the classification models show high accuracy in discriminating against high-risk patients with early-stage LUAD.

In [31], a deep neural network model is developed to predict the survival stage of lung cancer patients by predicting the cancer stage based on gene expression data of non-small cell lung cancer. The method used is to merge six independent Gene Expression Omnibus (GEO) datasets with a total of 614 patients to compare the performance of a DNN model with a random forest model. The datasets consist of microarray data and clinical data, where seven common NSCLC biomarkers are used to combine the datasets. The prognosis relevance values (PRV) are used to select eight additional gene biomarkers. So, in the end, 15 biomarkers in addition to the clinical data, are used to predict the patient's survival within 5 years. The evaluation of the model's performance showed that DNN achieves better than RF, with an accuracy of around 75.44 and an AUC value of around 0.816.

In [32], multiple cancer tumor types (kidney renal clear cell carcinoma (KIRC), BRCA, LUSC, LUAD, and uterine corpus endometrial carcinoma (UCEC) are classified using RNA-seq dataset that consists of 2,086 rows and 972 columns. Eight DL models are implemented and compared, which are CNN, Alex-net, Google Net, VGG16, VGG19, ResNet50, ResNet101, and ResNet152. The experimental results included many training and testing methods, such as different splitting percentages and 10-fold validation to select the best method to build the model, which ended up with selecting 70–30 splitting as the best among others. The experimental results show that CNN achieved the best among other techniques with 97% classification accuracy.

In [33], lung cancer prediction was performed using genomics dataset from TCGA, which consisted of 471 samples, and the GEO dataset, which consisted of 197 samples. The prediction results were compared using classical ML models (KNN, SVM, RF, Logistic regression, and MLP) and CNN with a four-input model. The proposed model converts the gene expression data into two types of gene expression images, which are gene functional information and the second kind with KEGG Pathway information. The prediction results of lung cancer are represented by the disease stage. The

highest performance is achieved by CNN, with 71.48% and 72.51% AUC values for the TCGA and GEO datasets, respectively.

Authors in [34] performed different ML and DL models, including random forest, SVM, logistic regression, gradient boosting, X-Gradient boosting, LSTM, and CNN. The evaluation of the classical ML models showed that RF achieved the lowest MSE value of 0.08% and the highest accuracy value of 0.97%. LSTM achieved the highest with an accuracy value of 0.94% and MSE value of 0.30%, while CNN showed the lowest performance, where the highest accuracy was 84%.

Authors in [14] proposed a DL model based on Focal Loss as a loss function and used the K-fold cross-validation method to select the optimal model. The selected K value in implementing the model was 5. The proposed DL model works based on several stages, which are data collection and preprocessing, Kullback-Leibler (KL) divergence gene selection, constructing the deep neural network, and then the validation stage. Two datasets were used, each of which consists of 19,565 genes and a total of 1,135 samples. The performance evaluation is performed on the entire dataset and on the selected dataset using the KL divergence gene selection method. Overall, the result of the proposed method outperforms the entire dataset with an accuracy above 99.8%.

Authors in [21] provided a review of the existing research on different cancer diseases that are predicted using gene expression data and various DL techniques, such as feedforward neural network (FFN), CNN, autoencoder (AE), and recurrent neural network (RNN). The authors compared the published work and illustrated that the highest accuracy of predicting lung cancer (LUAD) was achieved using the FFNN model with an accuracy of 96.67%, and the highest accuracy achieved by CNN to classify LUAD was 84.8%.

C) *Limitations*

Through the review of various research papers, several gaps have been identified in the classification of lung cancer stages using gene expression data. Most existing studies have focused on implementing binary classification to predict lung cancer or multi-class classification. However, limited research has specifically addressed the classification of lung cancer stages, which requires appropriate datasets. A major limitation of existing gene datasets is the small number of samples (patients) available for each stage, particularly for advanced stages (stage III and stage IV). This scarcity arises from the low survival rates of patients in advanced stages, resulting in imbalanced datasets due to their multi-classification output, as illustrated in Fig. 7 and Fig. 8.

Furthermore, applying machine learning methods to these datasets often leads to suboptimal performance or overfitting due to the high dimensionality of gene data. To address this challenge, previous research has employed various feature selection techniques. However, there remains significant potential to explore additional feature selection methods to enhance ML performance in analyzing gene datasets.

Moreover, previous studies often lack critical details about the performance of the applied approaches. Key aspects, such as the time required for the model to learn gene variations, the complexity of the proposed model, and the generality of the method to be applied to other types of cancer, are frequently overlooked. Therefore, there is a need for further investigation into these aspects to comprehensively assess the effectiveness and applicability of the methodologies employed in the literature.

The gaps and limitations in the current literature can be summarized as follows:

- Limitations of the methods used in previous literature:

  1) Insufficient focus on lung cancer stage classification: Limited studies have concentrated on the binary classification of lung cancer severity, leaving a gap in understanding this aspect of classification.
  2) Inadequate evaluation metrics: Many models lack detailed performance evaluations, such as training and testing times and their generalizability to other contexts.

- Limitations of the available datasets:

  1) High dimensionality: The large number of genes in datasets increases the risk of overfitting.
  2) Small sample size: The limited number of samples per stage leads to imbalanced datasets, reducing the reliability of findings and potentially biasing results.

Methodology

A. *Dataset Collection*

The dataset is collected from The Cancer Genome Atlas (TCGA), which is a landmark cancer genomics project that consists of gene expression, DNA, protein expression, and microRNA expression datasets for different cancer types. TCGA was considered a pilot project in 2006 when the National Cancer Institute (NCI) and National Human Genome Research Institute (NHGRI) authorized TCGA for a full production phase for a

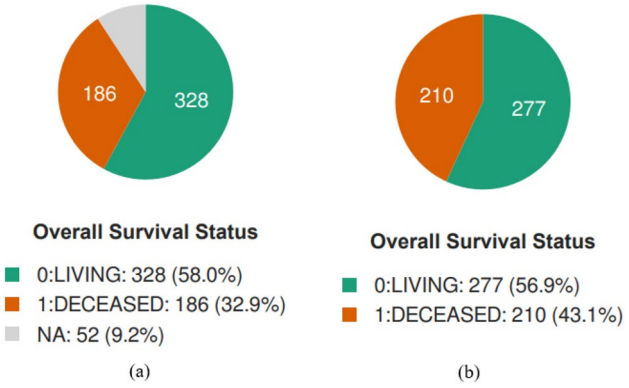

Fig. 3 Overall survival status of: (**a**) LUAD, and (**b**) LUSC datasets [36, 37]

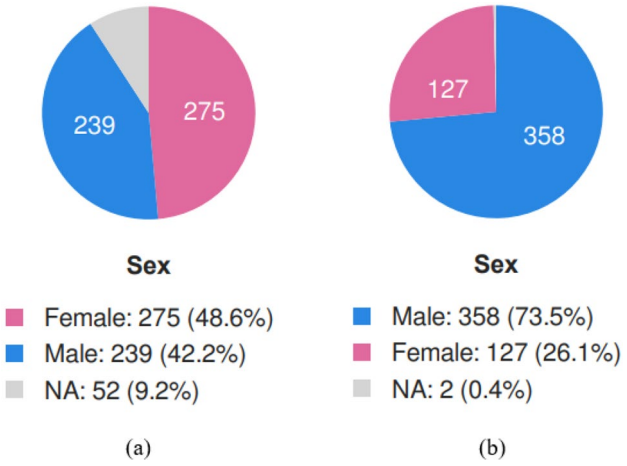

Fig. 4 Sex of the patients in: (**a**) LUAD, and (**b**) LUSC dataset [36, 37]

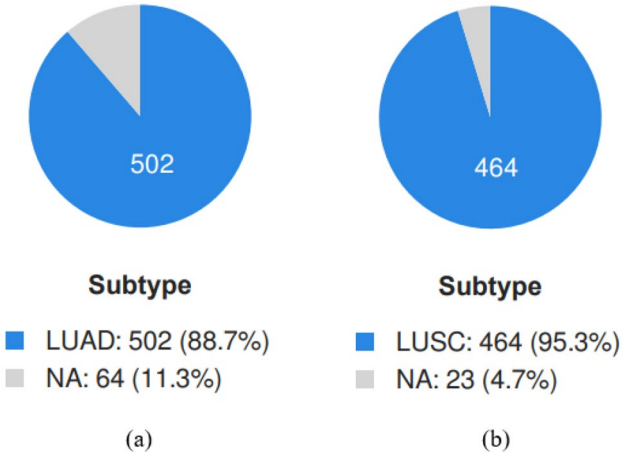

Fig. 5 Lung cancer subtype samples in: (**a**) LUAD, and (**b**) LUSC datasets [36, 37]

complete database of the molecular changes that happen in the tumors [29]. LUAD and LUSC are two types of non-small cell lung cancer, which can be detected using the gene expression that is included in the TCGA database, which is available in cBIoPortal for Cancer Genomics [35]. The gene expression dataset for each cancer type contains values for 20,530 genes, 566 LUAD patients, and 487 LUSC patients without any further details about the output label of the gene's variations for each patient. The patients' information, such as gender, age, race, and cancer tumor stage, are included in a different dataset file. Therefore, combining the gene expression dataset and the stage of the cancer tumor as an output is the first step before proceeding with other steps. In the end, the dataset used to build ML models consists of the gene expression values and the stage of the tumor.

B.*Data exploration*

While dealing with data, it is significant to understand the dataset before using it. Therefore, various types of analytic methodologies will be utilized to learn the various characteristics of the dataset and closely understand the data for more research.

Access to a set of data allows us to test how one sample, category, or subcategory of data affects another. Therefore, it is important to know the characteristics of the dataset samples. Since this work is a binary classification for classifying the stages of two different lung cancer types called LUAD and LUSC, it is important to know the overall survival status in each dataset. As shown in Fig. 3, the total number of patients in the LUAD dataset is 566, where the number of living patients reached 328, representing 58.0% of the dataset. While 186 patients are classified as deceased cases, and the remaining 9.2% were not specified. For the LUSC dataset, 277 patients out of 487 are categorized as living cases, representing 56.9% of the total number of patients. It is observed that the number of living status exceeds the deceased status by 13.8%.

In addition, both datasets will be explored from different features, including the sex of the patient, the age at which a condition was first diagnosed, and the subtype of lung cancer. Figure 4 shows that the LUAD dataset contains 275 female and 239 male patients, which means that the frequency of the data is 48.6% and 42.2% for females and males, respectively, while the remaining 52 patients were not specified. However, the LUSC dataset contains 127 female and 358 male patients, which means that the frequency of the data is 26.1% and 73.5% for females and males, respectively. Only two patients in the LUSC dataset were not applicable to be classified.

Figure 5 represents the appearance of any other categories or subtypes of lung cancer in the dataset. For the LUAD dataset, 502 cases are classified as LUAD cancer, while the remaining samples were categorized as non-applicable. In other words, 88.7% of the dataset is classified as LUAD cancer. However, for the LUSC dataset, 464 cases are classified as LUSC cancer, which is

approximately 95% of the whole dataset, while only 23 samples were classified as non-applicable.

The age at which a condition or disease was first diagnosed is shown in Fig. 6. As observed in the LUAD dataset, most of the patients are in the range (55, 75). The LUAD dataset contains a few samples under the age of 40 and over the age of 85. However, around 70 and 10 patients were not specified in the LUAD and LUSC datasets, respectively. For the LUSC dataset, the patients' ages range between (60 and 75).

C. *Data Preprocessing*

Building an ML model requires proper exploration and analysis of the dataset. Before building and evaluating the ML models, several preprocessing steps are required, such as data cleaning, handling imbalanced data, and value normalization.

1) Cleaning and Preparing each Dataset This preprocessing workflow aims to integrate two datasets related to LUAD, which are gene expression data and clinical patient information, both featuring the same patients identified by unique patient IDs. The first dataset contains gene expression data, which is initially transposed to facilitate manipulation, followed by removing missing values. Renaming the first column to 'patient_id' ensures consistency with the second dataset (clinical patient dataset). The second dataset, containing clinical information of the patients, retains only relevant columns ('patient_id' and 'tumor stage'). Then, the two datasets are merged based on the 'patient_id' column to create a new data frame. As a result, the final LUAD dataset will contain patient IDs, gene expression data, and tumor stage (label), comprising a total of 508 samples (rows) and 20,520 genes (columns). The same preprocessing steps have been implemented on the LUSC dataset, which also comprises gene expression data and clinical patient information for different patients. Consequently, the resulting LUSC dataset comprises 481 samples (rows) and 20,520 genes (columns).
2) Handling Imbalanced Data The problem tackled with the chosen datasets pertains to the multi-label output represented by the tumor stages. These stages correlate with survival statistics, signifying the extent of cancer cell presence in the body. Moving from stage I to stage IV, the survival probability decreases, as higher stages mean the cancer has spread more in the body [38]. Stage I lung cancer is limited to the lungs and has not spread to the lymph nodes. Stage II lung cancer may have spread to the nearby lymph nodes. Stage III lung cancer has spread to the lymph nodes and other nearby tissues. Stage IV lung cancer has spread from the lungs to other parts of the body. The sub-stages, A and B, are based on the size of the tumor.

As illustrated in Fig. 7 and Fig. 8, the multi-class distribution is imbalanced in terms of the number of samples between the stages, which accordingly results in inaccurate classification of the disease. Therefore, the samples are regrouped into two categories only (severe and non-severe), based on the stage severity and survival probability, which resulted in a balanced dataset, where the sample distribution of the binary classification is demonstrated in Fig. 8 and Fig. 10.

As can be seen in Fig. 9 and Fig. 10, the first step performed in dataset preprocessing is transforming the data from multi-classification output format to binary classification to have a balanced dataset. In order to create a more balanced dataset, multiple experiments have been performed using all the output classes in the multi-classification (9 classes) and by applying (9-n) where n = {1,2,3,4…,7}, where the highest result was achieved when n = 3. Therefore, the three lowest stages samples have been eliminated from the dataset.

Based on the opinion of medical experts, stage one can be considered as non-severe, and everything else can

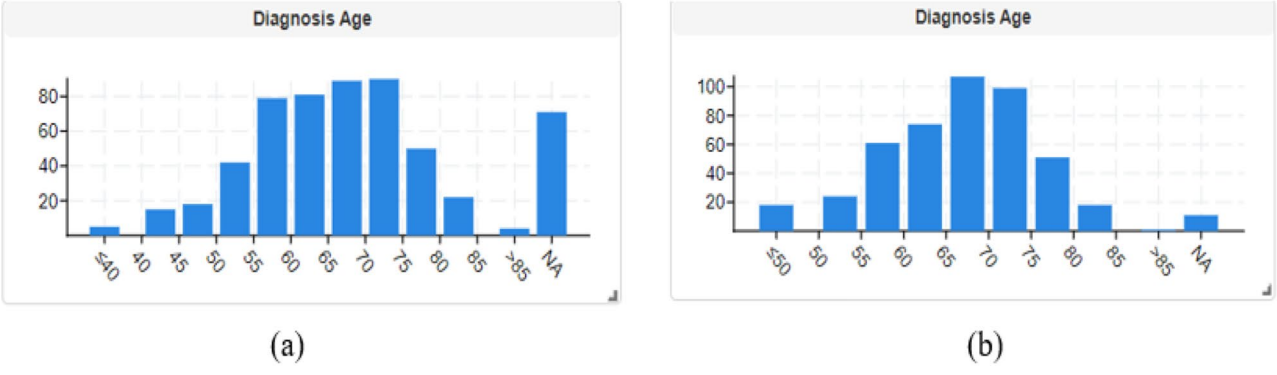

Fig. 6 Diagnosis ages in: (**a**) LUAD, and (**b**) LUSC datasets [36, 37]

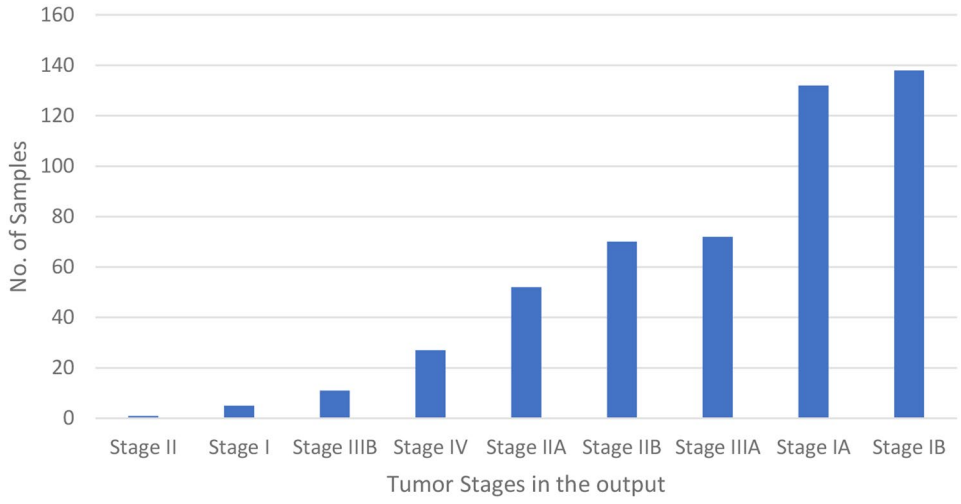

Fig. 7 LUAD-Sample distribution in multi-class classification

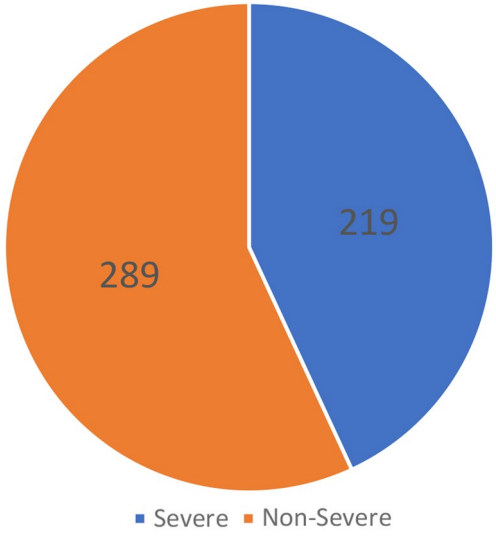

Fig. 8 LUSC-Sample distribution in multi-class classification

be severe, which will improve the accuracy as the dataset will be more balanced. Therefore, the output classes are regrouped into two categories only: severe and non-severe, as shown in Fig. 9 and Fig. 10, and encoded as 0 for non-severe stage and 1 for severe stage. After that, the data values are normalized, achieved by utilizing a standard scaler to rescale the mean of the data to 0 and the standard deviation of the data to 1. Figure 11 summarizes the overall implemented steps to preprocess the data and prepare it to be well fitted into different testing models.

D. *Feature Selection*

Feature selection is the process of filtering the features in the dataset to select the most significant features [39]. Handling the high dimensionality problem in a dataset using feature selection methods is crucial for improving the efficiency and accuracy of data analysis and ML. High-dimensionality datasets are often characterized by a large number of features, which can lead to increased computational complexity and negatively impact model performance. Feature selection techniques aim to mitigate these challenges by identifying and retaining the most informative features while discarding irrelevant or redundant ones.

In this research, the F-test feature selection method has been implemented in the data preprocessing steps to minimize the dataset dimensionality, considering three different percentages of the total feature count: 15%, 25%, and 40%. The F-test is a simple and well-established statistical method that evaluates the variance between groups or classes. It identifies features with statistically significant differences in variance across classes, making it an effective tool for distinguishing between classes within a dataset. It can handle multivariate feature selection, meaning it considers the joint effects of multiple features simultaneously. This is particularly important in cases where the interactions between features are crucial for classification, such as in a gene dataset that contains a huge number of genes.

Model design

A. *Training-testing Method*

Various training and testing methods have been implemented in this research, including k-fold cross-validation and hold-out methods. Through performing and evaluating both methods on classical ML models, it was figured out that the hold-out method achieved higher results compared to k-fold cross-validation. The experiments included evaluating 80–20 and 70–30 data splits and k-fold cross-validation with $k = 5$ and 10. As can be

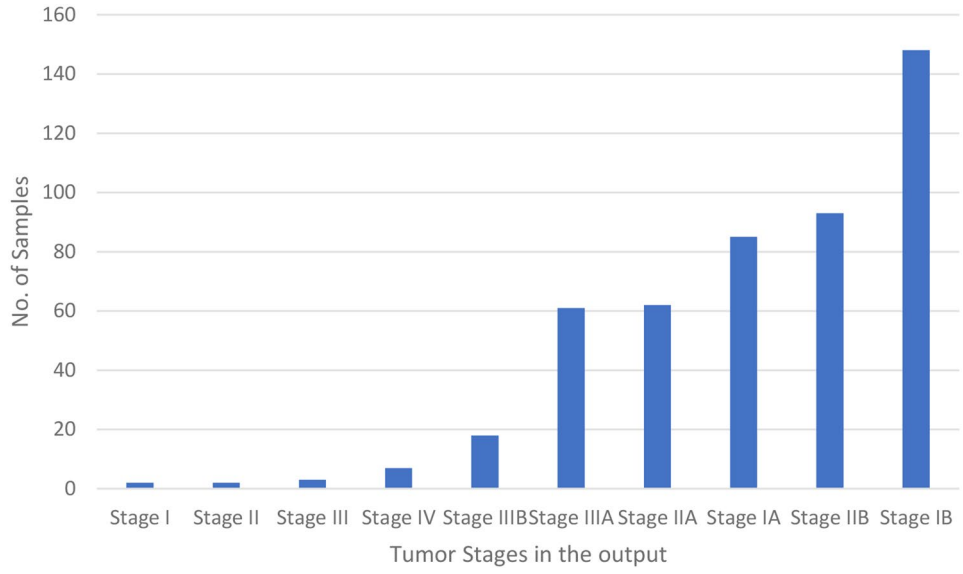

Fig. 9 LUAD-Sample distribution in binary classification

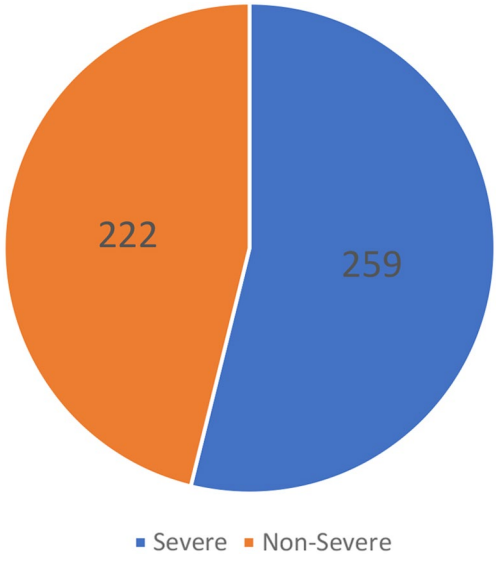

Fig. 10 LUSC-Sample distribution in binary classification

shown in Table 3, 70–30 splitting achieves the highest accuracy without using feature selection in the LUAD dataset, while 80–20 splitting achieves the highest results in the three remaining evaluating cases, which are: (1) LUAD using feature selection, (2) LUSC using feature selection, and (3) without using feature selection.

B.*Parameters of the Models*

Each model has specific parameters that must be predefined during the building and training processes. Moreover, some parameters are used to validate the model over multiple runs of the code. The details of these parameters are explained in this section, and the proposed architecture of the CNN model is illustrated in Fig. 12.

i. Training and testing splitting parameters: the training and testing splitting parameters are explained in Table 1.
ii. Model parameters: The parameters of the model are initially selected, as illustrated in Table 2. Some of the parameters are selected based on the nature of the dataset, such as setting the activation of the output layer as sigmoid as the label is binary. However, other parameters are tuned based on the performance of the model, where several parameters are evaluated and updated to improve the model's performance. The optimal parameters are determined based on performance metrics, which are assessed through the dataset-splitting method. Table 2 summarizes the parameters used in developing the models.

Results and discussion

This section presents the experimental results. Figure 13 illustrates the experiments conducted. The results are divided into two main parts. In the first part, preprocessing steps were implemented on the LUAD and LUSC datasets, followed by the construction of five models without the use of feature selection. The four classical ML models (KNN, SVM, RF, AdaBoost) models were then compared with the proposed CNN model, which is a DL approach. In the second part, the F-test feature selection method was applied to each dataset, and the results were compared using varying numbers of features. The results obtained using feature selection outperformed those without the feature selection method, highlighting the impact of feature selection on model accuracy.

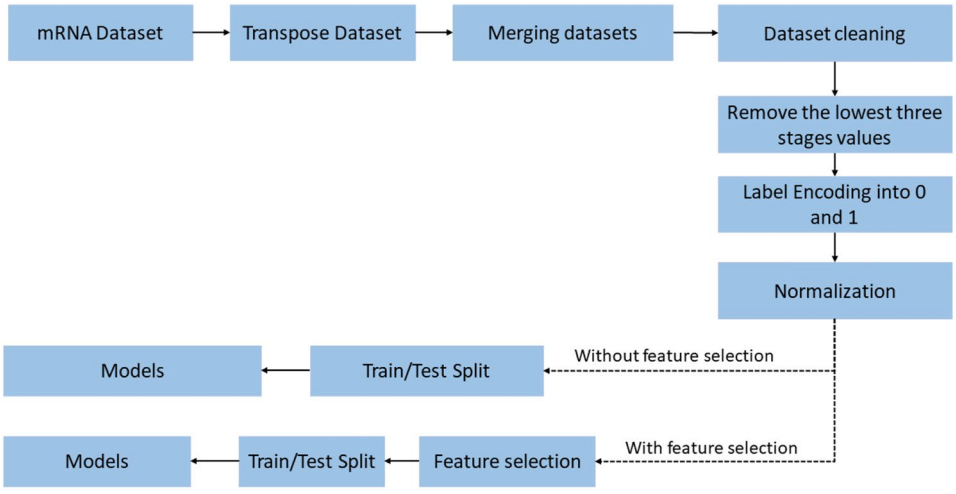

Fig. 11 Detailed preprocessing and steps of the proposed work

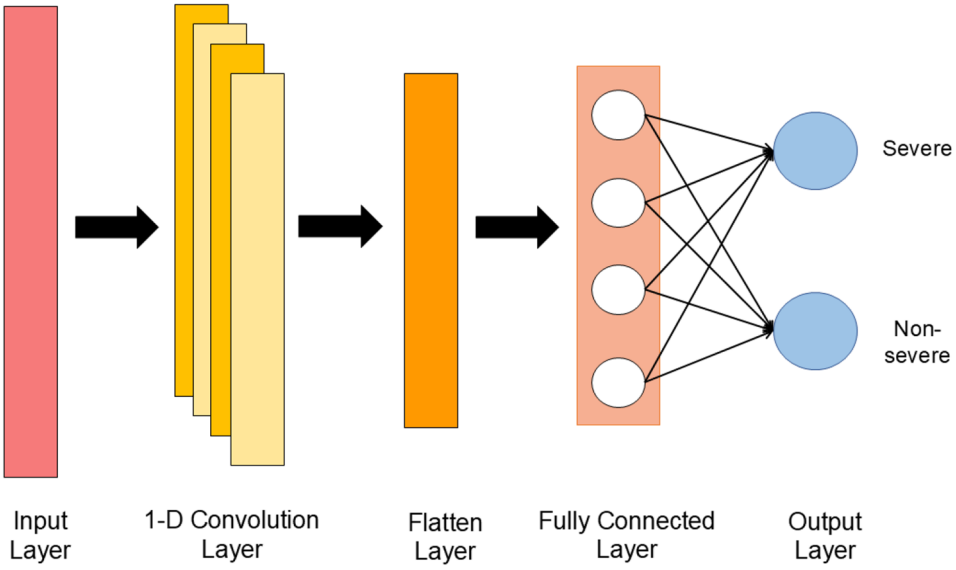

Fig. 12 CNN model architecture of the proposed work

Table 1 Training and splitting parameters

Parameter	Value	Meaning
Test size	0.2 and 0.3	Refers to the total number of test samples.
Shuffle	True	Shuffling the samples before splitting.
Stratify	None	It helps in retaining the same proportion of classes in the training and testing dataset.
Random state	2	It controls how the data is shuffled before splitting. It helps for getting the same output over multiple runs.

Furthermore, various evaluation metrics were used to compare and evaluate the proposed model.

In this research, several experiments are performed to select the best and optimal model. Firstly, we will start by building the ML models and training them using all the features in the datasets. Different training and testing methods are applied, such as k-fold cross-validation and hold-out methods. Table 3 illustrates the accuracy results of SVM, RF, KNN, and Adaboost using 5-fold and 10-fold cross-validation, as well as 80–20 and 70–30 data splitting methods. The results show that the 70–30 data splitting achieved the highest accuracy in the LUAD dataset without employing feature selection. In contrast, the 80–20 data splitting produced the highest results in three other evaluation scenarios: LUAD with feature selection, LUSC with feature selection, and without feature selection.

Table 2 The parameters of CNN model

Parameter	Value	Meaning
Random Seeds	7	To fix the reproducibility of the results over multiple runs of the code.
No. of epochs	50	The number of iterations of all the training dataset.
Activation function of the hidden layer	RELU	The default activation function for the hidden layers in CNN.
Activation function of the output layer	Sigmoid	It is a binary classification activation function of the output layer.
Patience value	5	It refers to the number of epochs the model can stop the training process, if there are no improvements.
Optimizer	Adam	It refers to Adaptive movement estimation algorithm that is utilized to update the network weight iterative based in training data.
Loss function	Binary cross entropy	It is used to compare the actual label with the predicted output.
Filter size	16	The number of channels in the output of the convolutional layer.
Kernel size	5	The size of the convolutional filters.

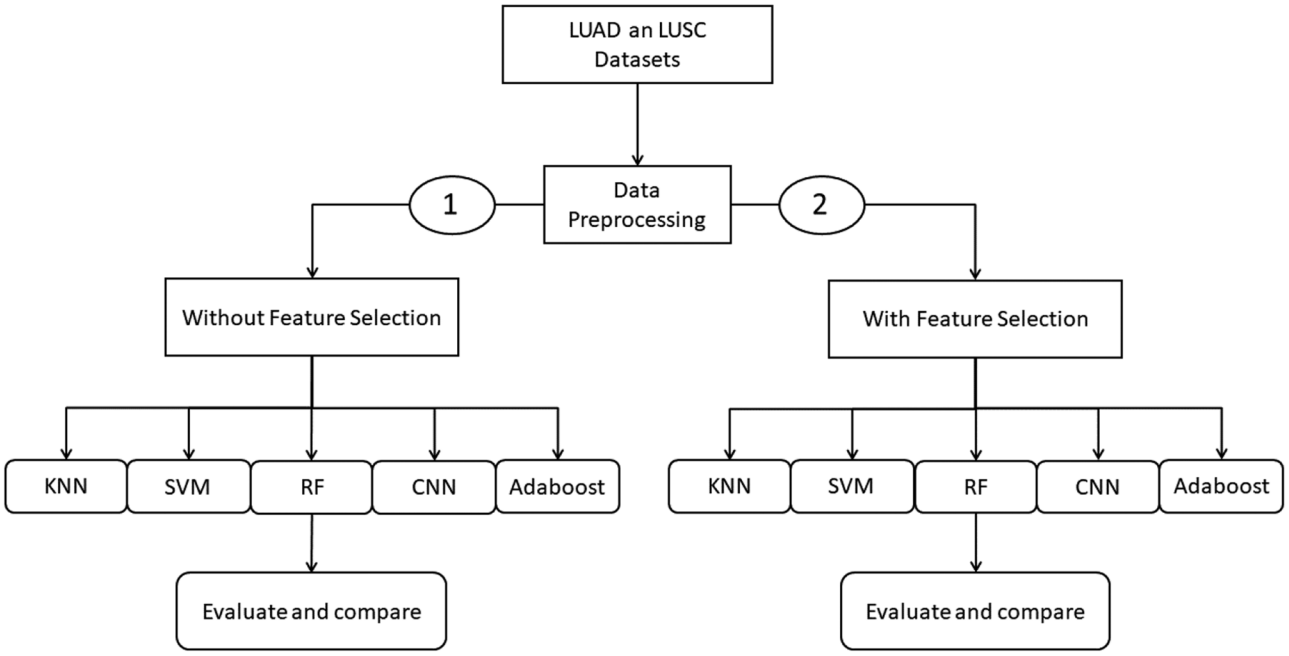

Fig. 13 Flow diagram of the results

Table 3 Evaluating classical ML models' accuracy using different training/testing methods without feature selection

	Training – testing method	SVM	RF	KNN	Adaboost
LUAD	10-fold cross-validation	0.57	0.54	0.58	0.54
	5-fold cross-validation	0.57	0.57	0.58	0.56
	80–20 splitting	0.61	0.58	0.57	0.60
	70–30 splitting	**0.63**	**0.60**	**0.59**	**0.56**
LUSC	10-fold cross-validation	0.56	0.53	0.50	0.52
	5-fold cross-validation	0.55	0.52	0.51	0.55
	80–20 splitting	0.61	0.59	0.59	0.57
	70–30 splitting	**0.58**	**0.57**	**0.54**	**0.50**

A. *Results without Feature Selection*

The first part of the experimental results, as shown in Fig. 13, involves preprocessing the two datasets and employing different models to classify the severity of lung cancer using all the features within these datasets. Notably, hold-out splitting methods consistently yield superior results compared to k-fold cross-validation in classical ML models. Consequently, the accuracy results of the CNN model on the LUAD and LUSC datasets were evaluated using both 70–30 and 80–20 data splits. The results illustrated that 70–30 data splitting obtained better accuracy for LUAD, while 80–20 data splitting was more effective for LUSC. Table 4 presents the accuracy results of the CNN model in classifying samples as severe or non-severe. The results show the performance of the CNN for both LUAD and LUSC datasets using 80–20 and 70–30 train-test splits, with the highest accuracy values reaching 61.49% and 57%, respectively.

Furthermore, the CNN model has been built with different architectures and parameter values. Two CNN architectures are mainly evaluated in this research. The first network architecture consists of an input convolutional layer, one flattening layer, and one dense layer that

Table 4 Evaluating CNN using 80–20 and 70–30 splitting methods without feature selection

Dataset	Training – testing method	Convolutional layers	Max-pooling layer	Max-pooling layer size	Dense layer	Learning rate	Batch size	Accuracy
LUAD	70–30 splitting	1	NA	NA	1	0.01	128	**61.49%**
	80–20 splitting							60.61%
LUSC	70–30 splitting	1	NA	NA	1	0.01	128	55%
	80–20 splitting							**57%**

Table 5 Detailed performance comparison between all the models without feature selection for LUAD dataset using 70–30 splitting method

Performance Metric	KNN (K = 7)	SVM	RF	AdaBoost	CNN
Accuracy	0.59	0.63	0.60	0.56	0.61
Recall	0.50	0.51	0.55	0.56	0.57
Precision	0.49	0.65	0.56	0.56	0.58
F-score	0.46	0.42	0.55	0.55	0.57
AUC	0.49	0.51	0.55	0.55	0.56
TPR	0.36	0.66	0.46	0.43	0.48
TNR	0.62	0.62	0.65	0.67	0.66
FPR	0.37	0.37	0.34	0.32	0.33
FNR	0.63	0.33	0.53	0.56	0.51
Training Time (Seconds)	0.09	0.87	1.31	22.60	30.09

Table 6 Detailed performance comparison between all the models without feature selection for LUSC dataset using 80–20 splitting method

Performance Metric	KNN (K = 7)	SVM	RF	AdaBoost	CNN
Accuracy	0.59	0.61	0.59	0.57	0.57
Recall	0.56	0.56	0.56	0.55	0.55
Precision	0.56	0.58	0.56	0.55	0.55
F-score	0.56	0.56	0.56	0.55	0.55
AUC	0.56	0.56	0.55	0.55	0.54
TPR	0.65	0.65	0.65	0.65	0.64
TNR	0.47	0.5	0.46	0.45	0.44
FPR	0.52	0.5	0.53	0.55	0.55
FNR	0.34	0.34	0.34	0.34	0.35
Training Time (Seconds)	0.20	1.13	1.53	23.98	35.86

includes two output neurons (severe and non-severe) without adding max-pooling layers. On the other hand, the second evaluated architecture consists of an input convolutional layer, two hidden convolutional layers, three max-pooling layers, one flattening layer, and three dense layers that include 128, 64, and 2 neurons, respectively. In addition, hyperparameters were optimized to ensure the models yield the best results. Table 5 and Table 6 describe a detailed performance comparison between the models by selecting the highest training and testing method: 70–30 splitting for the LUAD dataset and 80–20 splitting for the LUSC dataset, respectively.

B.*Results with Feature Selection*

Carrying on from the first part, a feature selection method was implemented before fitting the dataset to the ML models. Three different percentages of the total number of features have been evaluated, which are 15% (8000 genes), 25% (5000 genes), and 40% (3000 genes). After selecting the features, the same ML models were used to classify the severity of lung cancer. The classical ML models were evaluated through k-fold cross-validation and hold-out splitting methods, employing the three selected feature quantities for both the LUAD and LUSC datasets. The objective is to identify the most effective training approach and the optimal number of features.

Table 7 Accuracy evaluation of classical ML models using different training-testing methods with 3000 features

Dataset	Training – testing method	SVM	RF	KNN	AdaBoost
LUAD	10-fold cross-validation	0.71	0.58	0.58	0.58
	5-fold cross-validation	0.70	0.61	0.59	0.58
	80–20 splitting	0.79	0.63	0.61	0.67
	70–30 splitting	0.78	0.63	0.61	0.61
LUSC	10-fold cross-validation	0.711	0.63	0.60	0.58
	5-fold cross-validation	0.719	0.59	0.60	0.59
	80–20 splitting	**0.76**	**0.67**	**0.67**	**0.62**
	70–30 splitting	0.69	0.66	0.61	0.58

Table 8 Accuracy evaluation of classical ML models using different training-testing methods with 5000 features

Dataset	Training – testing method	SVM	RF	KNN	AdaBoost
LUAD	10-fold cross-validation	0.68	0.56	0.56	0.58
	5-fold cross-validation	0.67	0.58	0.56	0.58
	80–20 splitting	**0.79**	**0.65**	**0.62**	**0.66**
	70–30 splitting	0.73	0.69	0.59	0.62
LUSC	10-fold cross-validation	0.698	0.594	0.559	0.58
	5-fold cross-validation	0.696	0.596	0.554	0.567
	80–20 splitting	0.74	0.65	0.62	0.59
	70–30 splitting	0.69	0.62	0.60	0.51

For the LUAD dataset, the highest accuracy results were achieved using 80–20 data splitting and selecting 25% of the features, equating to approximately 5000 features. In contrast, for the LUSC dataset, the highest accuracy results were achieved using an 80–20 data splitting and selecting 15% of the features, which amounts to around 3000 features. Table 7, Table 8, and Table 9 illustrate the accuracy results for ML models obtained using 15%, 25%, and 40% feature percentages, respectively. The findings indicate that selecting 5000 features from the LUAD dataset outperforms the other feature amount, while in the case of the LUSC dataset, selecting 3000 features yields the best results. Consequently, the CNN

Table 9 Accuracy evaluation of classical ML models using different training-testing methods with 8000 features

Dataset	Training – testing method	SVM	RF	KNN	AdaBoost
LUAD	10-fold cross-validation	0.64	0.58	0.57	0.52
	5-fold cross-validation	0.62	0.57	0.58	0.56
	80–20 splitting	0.75	0.64	0.60	0.65
	70–30 splitting	0.72	0.64	0.60	0.57
LUSC	10-fold cross-validation	0.68	0.597	0.58	0.559
	5-fold cross-validation	0.66	0.592	0.57	0.552
	80–20 splitting	0.72	0.62	0.61	0.49
	70–30 splitting	0.68	0.58	0.57	0.54

model was evaluated using 5000 features for LUAD and 3000 features for LUSC.

It can be observed from Table 7, Table 8, and Table 9 that 80–20 data splitting consistently leads to the highest overall accuracy in both the LUAD and LUSC datasets. For instance, SVM achieved the highest accuracy values of 79% in LUAD and 76% in LUSC using 5000 and 3000 features, respectively. However, CNN outperformed the classical ML models by achieving accuracy rates of 93.94% and 88.42% in LUAD and LUSC datasets, respectively. As demonstrated in Table 10, in the case of the LUSC dataset, two different batch sizes (128 and 256) resulted in identical accuracy. Therefore, the training time was considered for both batch sizes to select the best parameter.

Moreover, for optimal selection for the number of features, the CNN is evaluated using a different number of features, as illustrated in Table 11. The results show that for LUAD severity classification, 5000 features are achieving the best result, and 3000 features are required to classify LUSC severity to achieve the best results. Tables 12 and 13 describe the detailed performance comparison between the classical ML models and the CNN model using feature selection for LUAD and LUSC datasets, respectively. Overall, in LUAD and LUSC datasets, the CNN model outperforms the other classical ML models in terms of classification accuracy, but it requires more time to be trained.

Table 11 The accuracy results of CNN for different number of features

	No. of selected features	Accuracy
LUAD	3000	91.92%
	5000	**93.94%**
	8000	88.89%
LUSC	**3000**	**88.42%**
	5000	85.26%
	8000	82.11%

Table 12 Detailed performance comparison between all the models using F-test for LUAD dataset using 5000 features

Performance Metric	KNN (K = 7)	SVM	RF	AdaBoost	CNN
Accuracy	0.62	0.79	0.65	0.66	**0.94**
Recall	0.55	0.75	0.64	0.64	**0.94**
Precision	0.60	0.83	0.64	0.64	**0.94**
F-score	0.50	0.76	0.64	0.64	**0.94**
AUC	0.54	0.74	0.64	0.64	**0.93**
TPR	0.58	0.91	0.57	0.57	**0.92**
TNR	0.62	0.75	0.71	0.71	**0.94**
FPR	0.37	0.25	0.28	0.28	**0.05**
FNR	0.41	0.08	0.42	0.42	**0.07**
Training Time (Seconds)	0.02	0.35	6.49	6.49	10.63

Table 10 Evaluating CNN with feature selection using 80–20 splitting method using different parameters values

Dataset	Convolutional layers	Max-pooling layer	Max-pooling layer size	Dense layer	Learning rate	Batch size	Accuracy
LUAD using 5000 features	1	NA	NA	1	0.001	32	88.89%
					0.01	32	89.90%
						16	89.90%
						64	89.90%
						128	**93.94%**
						256	91.92%
	3	3	1	3	0.01	128	91.92%
			3		0.01		75.76%
			1		0.001		81.82%
LUSC using 3000 features	1	NA	NA	1	0.001	32	85.26%
					0.01	32	82.11%
						16	85.26%
						64	84.21%
						128	**88.42%** (training time = 5.7 Sec)
						256	**88.42%** (training time = 5.6 Sec)
	3	3	1	3	0.01	128	75%
			2		0.01	128	78%
			1		0.001	128	75%

Table 13 Detailed performance comparison between all the models using F-test for LUSC dataset using 3000 features

Performance Metric	KNN (K = 7)	SVM	RF	AdaBoost	CNN
Accuracy	0.67	0.76	0.67	0.62	**0.88**
Recall	0.63	0.74	0.65	0.60	**0.87**
Precision	0.65	0.75	0.65	0.60	**0.89**
F-score	0.64	0.74	0.65	0.60	**0.87**
AUC	0.63	0.73	0.64	0.60	**0.86**
TPR	0.70	0.78	0.72	0.68	**0.87**
TNR	0.60	0.70	0.58	0.51	**0.90**
FPR	0.39	0.29	0.41	0.48	**0.09**
FNR	0.29	0.21	0.27	0.31	**0.12**
Training Time (Seconds)	0.02	0.11	0.60	3.72	5.6

Finally, it is crucial to mention that DL-based models are computationally expensive. To address this issue, feature selection methods are employed during data preprocessing. This practice significantly decreases model training and running time and facilitates model convergence. Generally, feature selection is becoming more and more relevant in discovering and studying significant genes.

Figure 14 illustrates all the model's accuracy results, both with and without feature selection. Remarkably, the CNN model achieved the best accuracy in classifying the severity of LUAD and LUSC when feature selection was applied. On the other hand, Fig. 15 shows the training time taken by each model to learn the gene variations, clearly illustrating that CNN requires the highest training time. Therefore, in future work, a hyperparameter optimization technique will be used to reduce it.

C. *Comparison with Previous Work*

Overall, the CNN model shows effective results by testing it on two different gene datasets for two types of lung cancer disease. The CNN model is chosen due to its prevalent use in the literature review for managing

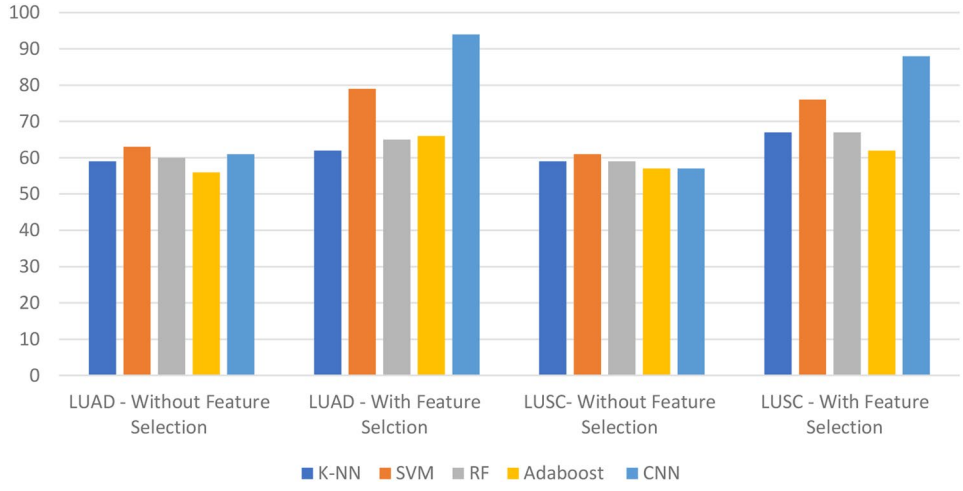

Fig. 14 Models' accuracy comparison

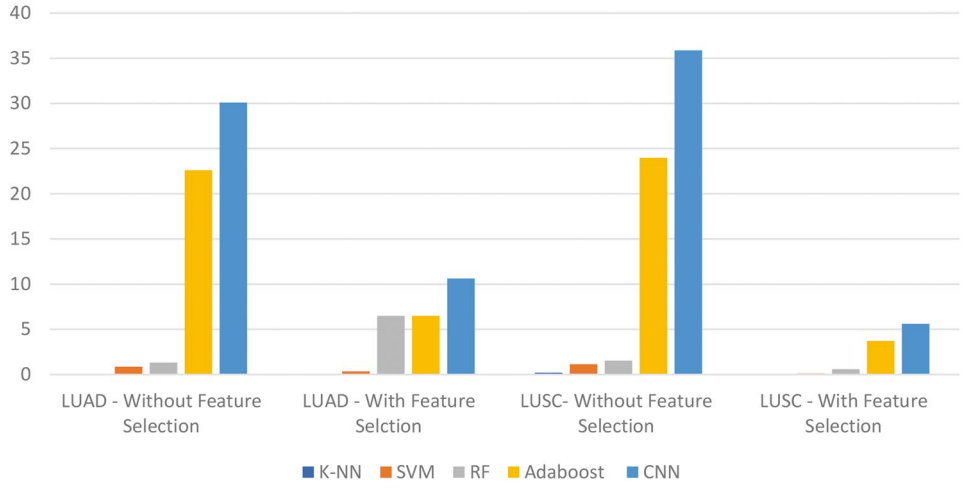

Fig. 15 Models' training time comparison

Table 14 Comparison of models' performance between our work and related works

Paper	Application	Model	Accuracy
[21]	Lung cancer prediction	CNN	84.8%
[26]	Lung cancer subtypes prediction	Random Forest	77.2%
[31]	Lung cancer survival prediction	Deep Neural Network	75.44%
[33]	Lung cancer long-term survival prediction	CNN	72.51%
[40]	Lung cancer subtypes prediction	Naive Bayes	90%
Our work	Lung cancer severity prediction	Optimized CNN	94%

high-dimensional data, especially in genomic studies. Its effectiveness is in its ability to extract features and reduce data dimensionality through convolutional, which improves both efficiency and helps to minimize overfitting. Therefore, a comparative analysis has been done in Table 14 to compare the performance of our proposed model with previous works in terms of detection accuracy. Our study demonstrates a significant improvement in predicting lung cancer severity using an optimized CNN model. For instance, the study in [21] utilized a CNN model for lung cancer prediction and achieved an accuracy of 84.8%, while the study in [26] applied a random forest model for predicting lung cancer subtypes with a 77.2% accuracy. Additionally, the authors in [31] used a deep neural network model for lung cancer survival prediction, resulting in a 75.44% accuracy, and the authors in [33] employed a CNN model for lung cancer long-term survival prediction, reaching 72.51% accuracy. Notably, the study in [40] achieved a 90% accuracy in lung cancer subtype prediction using Naive Bayes model. Our work surpasses these results, achieving an impressive 94% accuracy in predicting lung cancer severity, highlighting the efficacy of our optimized CNN model in this application.

Conclusion

Globally, cancer is a serious health issue. To classify the severity of lung cancer, we have compared classical ML models with the CNN model in this research. We specifically examined gene expression data from two kinds of lung cancer (LUAD and LUSC). The output is converted from multi-class classification to binary classification based on the tumor spread to address the imbalanced dataset. Additionally, the F-test feature selection approach is used to minimize the dimensionality of the datasets to prevent overfitting in classification. The findings demonstrate that the F-test method is essential for reducing data dimensionality and choosing useful information, enhancing prediction accuracy, and substantially reducing computational time.

To sum up, this research applied several classical ML models, including SVM, KNN, RF, and Adaboost, to detect two different lung cancer types (LUAD and LUSC) using gene expression data. However, since the obtained performance of the model has demonstrated very low results using all features, the F-test feature selection method has been utilized to improve the performance of the models. Furthermore, these classical ML models have been compared with the proposed CNN model. Experimental results showed that the proposed CNN model outperformed the other ML models and obtained the highest performance in detecting lung cancer with an accuracy of 93.94% for the LUAD dataset and 88.42% for the LUSC dataset using the feature selection method.

Using the F-test feature selection method, we identified 3000 and 5000 significant genes out of more than 20,000 genes for LUSC and LAUD, respectively. Subsequently, through analysis of TCGA data from cancer patient samples, these genes hold the potential for clinical utilization by physicians in classifying the severity of lung cancer. We are confident that if doctors check these genes for their patients, they will be able to determine the stage of lung cancer, especially since the used evaluation datasets are highly variable in terms of the collected samples, thereby enhancing the diagnostic process and enabling more personalized treatment strategies. Eventually, this research has proven the power of the feature selection method in improving the classification accuracy using all the models by evaluating the models using different performance metrics to ensure the efficiency and reliability of the proposed model.

A. *Answers to Research Questions*

To conclude the analysis of the results of this research, we will address the research questions outlined in Section "Introduction".

Research Question 1: Can the severity of lung cancer be detected by ML using gene expression data?

The experimental results confirm that ML models can effectively detect the severity of lung cancer using gene expression data. The classification performance varies depending on the model, dataset, and feature selection method applied. In Table 5 and Table 6, models trained without feature selection on the LUAD and LUSC datasets achieved moderate accuracy, with SVM achieving the highest accuracy (0.63) for LUAD and (0.61) for LUSC. However, when the F-test feature selection method was applied, the accuracy of all models significantly improved, as can be seen in Table 12 and Table 13. CNN achieved the best performance, reaching 0.94 accuracy for LUAD and 0.88 for LUSC, indicating that DL models

can leverage high-dimensional gene expression data more effectively.

Research Question 2: What is the most effective ML or DL model to detect the severity of lung cancer?

From the performance comparisons, CNN demonstrated the highest classification performance in both LUAD and LUSC datasets when feature selection was applied. In Table 12 and Table 13, CNN achieved an accuracy of 0.94 and 0.88 for LUAD and LUSC, respectively. Traditional ML models such as SVM and RF performed well, with SVM reaching 0.79 accuracy for LUAD and 0.76 for LUSC. However, deep learning models outperformed ML models in handling gene expression data, making CNN the most effective model for lung cancer severity detection.

Research Question 3: How does feature selection affect the performance of ML models on genetic datasets?

Feature selection significantly improves the performance of ML models by reducing the dataset dimensionality and enhancing classification accuracy. When models were trained without feature selection (Table 5 and Table 6), their accuracy ranged between 0.56 and 0.63. However, after applying the F-test feature selection, the accuracy of all models improved significantly (Table 12 and Table 13). For instance, CNN's accuracy increased from 0.61 (LUAD) and 0.57 (LUSC) without feature selection to 0.94 (LUAD) and 0.88 (LUSC) after selecting relevant features. Similarly, SVM's accuracy improved from 0.63 (LUAD) and 0.61 (LUSC) to 0.79 (LUAD) and 0.76 (LUSC). These findings demonstrate that selecting the most relevant features enhances model efficiency and accuracy while reducing computational costs.

Research Question 4: How does tuning the hyperparameters affect the performance of a DL model?

Hyperparameter tuning plays a crucial role in optimizing deep learning models for better classification performance. Table 10 highlights the impact of different hyperparameter settings on CNN performance for LUAD and LUSC datasets. For LUAD using 5000 features, increasing the batch size from 32 to 128 resulted in a significant accuracy improvement from 88.89% to 93.94%. Similarly, adjusting the number of convolutional layers and max-pooling settings influenced performance, with a three-layer configuration achieving 91.92% accuracy when using a batch size of 128. For LUSC using 3000 features, batch size adjustments also played a key role, with 128 achieving an accuracy of 88.42% while maintaining reasonable training time (5.7 seconds). A learning rate of 0.01 produced variable results, highlighting the importance of selecting an optimal value to balance model convergence and generalization. These results highlight the crucial role of hyperparameter tuning, including batch size, learning rate, and the number of convolutional layers, in optimizing CNN performance. Choosing optimal hyperparameters improves both accuracy and efficiency.

B. *Future Work and Suggestions*

Through reviewing the literature and addressing the research problems, we have investigated some areas of improvement that can be implemented in the future, such as:

- Due to the insufficient samples in the used datasets, merging the LUAD and LUSC datasets together as they share the same features (genes) and creating a binary classification problem to classify the two subtypes would lead to better classification results.
- Develop other deep learning models.
- Test and evaluate various feature selection methods that have been used in the literature to identify the most suitable approach for this problem.

Acknowledgements
The authors would like to thank University of Sharjah for the financial support.

Author contributions
AN, NA and AO wrote the manuscript; AN, NA revised the manuscript; AN, NA and AO conducted experiments.

Funding
University of Sharjah.

Data availability
All datasets used in this research are publicly available and are discussed in Section "Methodology". It is available in the following repositories [35, 36], and it can be accessed from the following references [37, 41–43].

Declarations

Ethics approval and consent to participate
Not applicable as the research is done on publicly available datasets.

Consent for publication
Not applicable.

Informed consent
This study does not involve any experiments on animals.

Competing interests
The authors declare no competing interests.

Received: 13 July 2024 / Accepted: 23 April 2025
Published online: 14 May 2025

References

1. Alakwaa W, Nassef M, Badr A. Lung Cancer Detection and Classification with 3D Convolutional Neural Network (3D-CNN). Int J Adv Comput Sci Appl. 2017;8. https://doi.org/10.14569/IJACSA.2017.080853.
2. Cancer NL, et al. Deep learning for the classification of small-cell and non-small-cell lung cancer. 2020.
3. Rajagopalan K, Babu S. 2020. The detection of lung cancer using massive artificial neural network based on soft tissue technique. BMC Med Inf Decis Mak. 20(1):1–13. https://doi.org/10.1186/s12911-020-01220-z
4. Zhang H, Hu D, Duan H, Li S, Wu N, Lu X. 2021. A novel deep learning approach to extract Chinese clinical entities for lung cancer screening and staging. BMC Med Inf Decis Mak. 21(2):1–13. https://doi.org/10.1186/s12911-021-01575-x
5. Chen JW, Dhahbi J. 2021. Lung adenocarcinoma and lung squamous cell carcinoma cancer classification, biomarker identification, and gene expression analysis using overlapping feature selection methods. Sci Rep. 11(1):13323. https://doi.org/10.1038/s41598-021-92725-8
6. Guan X, et al. Construction of the XGBoost model for early lung cancer prediction based on metabolic indices. BMC Med Inf Decis Mak. 2023;23(1):1–16. https://doi.org/10.1186/s12911-023-02171-x.
7. Musthafa MM, Manimozhi I, Mahesh TR, Guluwadi S. Optimizing double-layered convolutional neural networks for efficient lung cancer classification through hyperparameter optimization and advanced image pre-processing techniques. BMC Med Inf Decis Mak. 2024;24(1):1–21. https://doi.org/10.1186/s12911-024-02553-9
8. Thakur SK, Singh DP, Choudhary J. Lung cancer identification: a review on detection and classification. Cancer Metastasis Rev. 2020;39(3):989–98. https://doi.org/10.1007/s10555-020-09901-x
9. Belinsky SA, et al. Gene promoter methylation in plasma and sputum increases with lung cancer risk. Clin Cancer Res. 2005;11:6505–11.
10. Salehi-Rad R, Li R, Paul MK, Dubinett SM, Liu B. The biology of lung cancer: development of more effective methods for prevention, diagnosis, and treatment. Clin Chest Med. 2020;41(1):25–38.
11. Hu D, Zhang H, Li S, Duan H, Wu N, Lu X. An ensemble learning with active sampling to predict the prognosis of postoperative non-small cell lung cancer patients. BMC Med Inf Decis Mak. 2022;22(1):1–12. https://doi.org/10.1186/s12911-022-01960-0
12. Huang S, Cai N, Pacheco PP, Narrandes S, Wang Y, Xu W. Applications of support vector machine (SVM) learning in cancer genomics. Cancer genomics \& proteomics. 2018;15(1):41–51.
13. Taylor P, Altman NS, Altman NS. An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression, no. December. 2014. 2012;37–41. https://doi.org/10.1080/00031305.1992.10475879.
14. Liu S, Yao W. Prediction of lung cancer using gene expression and deep learning with KL divergence gene selection. BMC Bioinform. 2022;23(1):1–11. https://doi.org/10.1186/s12859-022-04689-9
15. Wang R. AdaBoost for feature selection, classification and its relation with SVM, a review. Phys Procedia. 2012;25:800–07.
16. Suthaharan S, Suthaharan S. Support vector machine, Mach. Learn. Model. algorithms big data Classif. Think with Examples Eff Learn. 2016;207–35.
17. Chang V, Bhavani VR, Xu AQ, Hossain MA. An artificial intelligence model for heart disease detection using machine learning algorithms. Healthc Anal. 2022;2:100016.
18. Omar AA-C, Nassif AB, Lung Cancer Prediction using Machine Learning based Feature Selection: a comparative Study, In 2023 Advances in Science and Engineering Technology International Conferences (ASET). 2023:1–6.
19. Abujabal NA, Nassif AB, Enhanced Heart Failure Prediction Using Feature Selection-based Machine Learning Models. In 2023 Advances in Science and Engineering Technology International Conferences (ASET). 2023:1–6.
20. Nassif AB, Al-Chikh Omar A. A Comprehensive Study on Machine Learning in Breast Cancer Detection and Classification, Proceedings of the 6th International Conference on Advances in Artificial Intelligence. 2022:81–87.
21. Ravindran U, Gunavathi C. A survey on gene expression data analysis using deep learning methods for cancer diagnosis, Prog. Biophys Mol Biol. 2022 August, 2023;177:1–13. https://doi.org/10.1016/j.pbiomolbio.2022.08.004.
22. Siraj MJ, Ahmad T, Ijtihadie RM. Analyzing ANOVA F-test and Sequential Feature Selection for Intrusion Detection Systems. Int J Adv Soft Comput \& Its Appl. 2022;14(2).
23. Javaheri SH, Sepehri MM, Teimourpour B. Chapter 6 - Response Modeling in Direct Marketing: a Data Mining-Based Approach for Target Selection. Data Mining Applications with *R*. Zhao Y, Cen Y. Eds. Boston: Academic Press; 2014. p. 153–80. https://doi.org/10.1016/B978-0-12-411511-8.00006-2.
24. Obeidat AA. Hybrid approach for botnet detection using K-means and K-medoids with hopfield neural network. Int J Commun Networks Inf Secur. 2017;9(3):305–13.
25. Xiao Y, Wu J, Lin Z, Zhao X. A deep learning-based multi-model ensemble method for cancer prediction, Comput. Methods Programs Biomed. 2018;153:1–9. https://doi.org/10.1016/j.cmpb.2017.09.005.
26. Yuan F, Lu L, Zou Q. Analysis of gene expression profiles of lung cancer subtypes with machine learning algorithms. Biochim Biophys Acta (BBA)-Molecular Basis Dis. 2020;1866(8):165822.
27. Ramos B, Pereira T, Moranguinho J, Morgado J, Costa JL, Oliveira HP. An Interpretable Approach for Lung Cancer Prediction and Subtype Classification using Gene Expression. Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS. 2021:1707–10. https://doi.org/10.1109/EMBC46164.2021.9630775.
28. Xiao Y, Wu J, Lin Z, Zhao X. A semi-supervised deep learning method based on stacked sparse auto-encoder for cancer prediction using RNA-seq data, Comput. Methods Programs Biomed. 2018;166:99–105. https://doi.org/10.1016/j.cmpb.2018.10.004.
29. Khalifa NEM, Taha MHN, Ali DE, Slowik A, Hassanien AE. Artificial intelligence technique for gene expression by tumor RNA-Seq data: a novel optimized deep learning approach. IEEE Access. 2020;8:22874–83.
30. Bao X, Shi R, Zhao T, Wang Y. Immune landscape and a novel immunotherapy-related gene signature associated with clinical outcome in early-stage lung adenocarcinoma. J Mol Med. 2020;98(6):805–18.
31. Lai Y-H, Chen W-N, Hsu T-C, Lin C, Tsao Y, Wu S. Overall survival prediction of non-small cell lung cancer by integrating microarray and clinical data with deep learning. Sci Rep. 2020;10(1):4679.
32. Rukhsar L, Bangyal WH, Ali Khan MS, Ag Ibrahim AA, Nisar K, Rawat DB. Analyzing RNA-seq gene expression data using deep learning approaches for cancer classification. Appl Sci. 2022;12(4):1850.
33. Wang S, Zhang H, Liu Z, Liu Y. 2022. A Novel Deep Learning Method to Predict Lung Cancer Long-Term Survival With Biological Knowledge Incorporated Gene Expression Images and Clinical Data. Front Genet. 13(March):1–13. https://doi.org/10.3389/fgene.2022.800853
34. Thakur T, Batra I, Malik A. Performance evaluation of various machine learning and deep learning models for gene expression. J Phys Conf Ser. 2022;2327(1):12034.
35. cBioPortal. https://www.cbioportal.org
36. Nathional Cancer Institute–. GDC Data Portal. https://portal.gdc.cancer.gov/repository.
37. Lung Squamous Cell Carcinoma (TCGA, PanCancer Atlas). https://portal.gdc.cancer.gov/projects/TCGA-LUSC
38. Society AC. Non-Small Cell Lung Cancer Stages. https://www.cancer.org/
39. Abujabal NA, Nassif AB. Meta-heuristic algorithms-based feature selection for breast cancer diagnosis: a systematic review. In 2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME). 2022:1–6. https://doi.org/10.1109/ICECCME55909.2022.9988285.
40. Pineda AL, et al. On Predicting lung cancer subtypes using 'omic'data from tumor and tumor-adjacent histologically-normal tissue. BMC Cancer. 2016;16:1–11.
41. Adenocarcinoma L (TCGA, PanCancer Atlas). https://portal.gdc.cancer.gov/projects/TCGA-LUAD
42. cBioPortal (Lung Adenocarcinoma (TCGA, PanCancer Atlas)). https://www.cbioportal.org/study/summary?id=luad_tcga_pan_can_atlas_2018
43. cBioPortal (Lung Squamous Cell Carcinoma (TCGA, PanCancer Atlas)). https://www.cbioportal.org/study/summary?id=lusc_tcga_pan_can_atlas_2018

Publisher's Note