A Comparison of Deep Learning Methods for Liver Disease Diagnosis from Exhaled Human Breath

Nilakshi Maruti Mule,[1]* Dipti D. Patil[2]

[1]*PhD Scholar (Corresponding author)*
Department of Computer Engineering,
Smt. KashibaiNavale College of Engineering, Pune, India
nilmule@gmail.com
[2]*Professor*
Department of Information Technology,
MKSSS's Cummins College of Engineering for Women, Pune, India
dipti.patil@cumminscollege.in

KEYWORDS
Deep Learning, Exhaled Breath, Liver Disease, BiLSTM,1D-CNN

ABSTRACT
The liver is a vital organ in the human body. The prevalence of liver problems has surged worldwide at an unprecedented rate as a result of unhealthy lifestyles and excessive alcohol usage. Chronic liver disease is a leading cause of mortality that impacts a significant section of the global population. Obesity, undiscovered hepatitis infection, alcohol misuse, hemoptysis or hematemesis, renal or hepatic failure, jaundice, hepatic encephalopathy, and various other conditions contribute to this condition. Therefore, prompt action is necessary to diagnose the ailment before it becomes critical. The assigned task examines several deep-learning models for gathering data from exhaled breath samples. The model's performance is evaluated based on various criteria, including accuracy, specificity, sensitivity, precision, recall, and F1-Score. These factors are examined using the training dataset to determine the training and testing loss. The proposed work does a comprehensive experimental examination of these parameters, exploring their impact on accuracy and loss function.
Additionally, it evaluates the appropriateness of these models. The deep learning models utilized in the recommended work are BiLSTM, LSTM, GRU, and 1D-CNN. The dataset is divided into 80% for training and 20% for testing, using 24 liver patient samples and 15 healthy person samples. Of the several algorithms, the BiLSTM and 1D-CNN had superior performance in predicting liver illness, achieving accuracies of 0.99 and 0.98, respectively. In addition, these two algorithms demonstrated superior precision, F1-Score, recall, specificity, and sensitivity. Therefore, these two algorithms are regarded as the superior methods for early detection of liver disease..

1 INTRODUCTION

Hepatitis, fibrosis, and cirrhosis are just a few of the many disorders that can impact the body's principal organ, the liver, which controls metabolism and protein absorption [1]. About 2 million people die every year from liver disease, making it one of the leading causes of death worldwide [2]. Limiting the incidence of liver disease and related fatalities requires practical prevention efforts, such as limiting alcohol intake, promoting healthy lifestyles, and vaccination against viral hepatitis [3]. Liver function tests and imaging approaches are also used for liver diagnosis. However, the blood test requires conventional labs and imaging techniques, which have other intrinsic challenges connected to contrast, such as more expensive equipment, operator experience, and other problems [4].

By detecting specific proteins and liver enzymes in a blood sample, a liver function test is the initial diagnostic tool for a range of liver illnesses. Fetor hepaticus, the classic musty breath scent, was once one of the most essential clinical symptoms doctors looked for when diagnosing liver insufficiency [5]. The expelled breath already contained thousands of volatile organic compounds. Exhaled breath contains nitrogen, oxygen, carbon dioxide, and hundreds of volatile organic and inorganic chemicals. These volatile organic compounds might originate from either internal or external sources. The content or profile of volatile organic compounds (VOCs) is regulated by many metabolic causes [6, 7]. Since the levels of VOCs in the breath of healthy individuals and patients with specific diseases differ, these VOCs can be used as biomarkers for certain disorders [8]. Using breath biomarkers has many benefits, one of which is that it is an easy, painless, and repeatable process. Using breath biomarkers to diagnose and monitor several diseases, such as COVID-19, cancer, diabetes, and infectious diseases, has demonstrated encouraging results [9, 10]. Breath provides more rational advantages than blood test procedures and has an unlimited supply. Better patient outcomes, lower healthcare expenditures, and more advanced medical research are all possible outcomes of deep learning's revolutionary potential in the healthcare industry [11]. The purpose of its application in predictive analytics is to enhance the precision of medical diagnoses, new drug development, and the identification of risk factors for the beginning or progression of illness [12]. Breath biomarkers, when used with either a single VOC or a panel of VOCs, can reliably detect liver illness, according to previous studies [13]. Patients preferred the breath test because it was less invasive and took less time than conventional blood testing without sacrificing accuracy. Liver disease changes numerous metabolic pathways, which impacts innumerable VOCs, as the liver is in charge of metabolism. Many different types of VOCs have connections to these pathways [14].

GC-MS, IMS, and GC-FID are the procedures most often used when testing breath. These procedures necessitate a qualified operator and are time-consuming and costly. An easy, quick, and inexpensive way to get improved accuracy, sensitivity, and specificity results is to use an electronic nose (E-Nose) for breath analysis. Although e-noses are accessible, they can be pricey and complicated to operate. That is why we need to developmore straightforward and more cost-effective systems.

This study aims to assess the precision of various deep learning algorithms in predicting liver disorders, namely LSTM, GRU, Bi-LSTM, and 1D-CNN. It will be achieved by analyzing a self-generated dataset of exhaled breath using an IoT-based analyzer device. The performance of these algorithms will be compared to determine their effectiveness. The primary contributions of the proposed work include conducting experimental analysis on different deep learning models, performing a comparative study of the models based on observations with various parameters, evaluating the strengths and weaknesses of each model through testing, and suggesting the suitability of these models in specific application areas based on the experimental analysis and evaluations.

The work is structured as follows: section 2 outlines the methodology, section 3 introduces the suggested model architecture, sections 4 and 5 showcase the results of several deep learning algorithms, and lastly, the paper concludes in section 5. Table 1 gives an overview of the recent studies on diagnosing liver disease using the exhaled breath analysis method.

2. METHODOLOGY

The flow chart for the comparative study of different deep learning algorithms for disease identification and classification includes several essential components and processes. The system is specifically designed to assist in developing, training, and evaluating deep learning models using highly relevant medical datasets. Here is a breakdown of each component of the proposed system. Here is the flow chart of this study, as depicted in Figure 1.

2.1. Data Set

The dataset is obtained and converted into CSV format, containing the following columns: TGS-2600, TGS-2602, TGS-822, MQ-7, MQ-9, gender, and disease. A value of zero shows the absence of disease, while a value of one indicates the presence of disease.

The dataset comprises 24 liver patient samples, of which 15 are healthy. To train a resilient model, ensure the data is thorough, encompassing a diverse set of values for each characteristic. Figure 2 displays the dataset that has been sampled.

2.2 Data Preprocessing

The gathered data is subjected to preprocessing to guarantee uniformity, purity, and compatibility with deep learning methods. It involves addressing the issue of missing values. Eliminating data points that deviate significantly from the average and standardizing attributes to maintain consistency across datasets. In addition, data augmentation techniques can be used to enhance the diversity and size of datasets, hence improving the resilience of the trained models.

2.3 Dataset Training

The dataset is partitioned into 80% for training and 20% for testing. The models are trained on preprocessed datasets utilizing a subset of the data expressly set aside for training. During training, the models progressively acquire knowledge from the input data, modifying their parameters to reduce the discrepancy between the predicted and actual illness classifications. Validation sets are employed to assess the performance of a model and mitigate the risk of overfitting. Early stopping methods can be used to cease training when the validation performance starts to deteriorate, thus preventing the model from diverging.

Table 1 Summary of the related work

Reference	Methods/Technology Used	Dataset	Accuracy	Sample Collection	Biomarkers
15	Selective Ion Flow Tube Mass Spectrometry (SIFT-MS), Kruskal-Wallis Test, Pearson's chi-square tests, Multivariable logistic regression	61	85%	Mylar Bag, 8H fast	Acetone, Benzene, Carbon Disulfide, Isoprene, Pentane and Ethane
16	GC-MS, ANOVA, Multivariable logistic regression	30- NAFLD with or without cirrhosis, 15-Healthy	98%	Tedlar Bag, Overnight fasting	Isoprene, Acetophenone, Terpinene
17	Thermal Desorption GC-MS, Shapiro-wilk test, Mann Whitney U test, Least square regression	32-Cirrhotic,12-Cirrhotic Patients with HCC, 40-Healthy	78%	ReCIVA Breath Sampler, No dietary restrictions	Limonene
18	E-nose, PLS-DA	39-NCCLD, 65-Liver Cirrhosis	81.3%	Pneumopip, Fasting 12 H	VOCs
19	TD-GC-FAIMS, 1D-CNN, Wilcoxon Rank, Kruskal-Wallis Rank	35-Liver Cirrhosis, 11-Healthy	90%	ReCIVA Breath Sampler, Fasted at least 4H	VOCs
20	GC-tof-MS, Mann-Whitney U-test, Chi-Square test, PLS-DA	87- Chronic Liver Disease(CLD), 34-Compensated Cirrhosis(CIR), 31-Healthy	81%	Tedlar Bag, No dietary restrictions	3-methyl butanal, Octane, Propanoic acid, Terpene, Dimethyl Disulfide
21	Mann-Whitney U test, Kruskal-Wallis test, RF,Naïve Bayes	30-Liver, 33-Healthy	84-94%	Perkin Elmer, No dietary restrictions	Acetone, Alkane, Toluene, Isopropyl Alcohol, Ethyl Acetate, Furan
22	GC-MS, Hierarchical Clustering, Mann-Whitney U-test	46-Cirrhosis, 42-Control	95%	ReCIVA Breath Sampler, No dietary restrictions	Pentene, Limonene, Benzene, Terpepe

2.4 Deep Learning Classification Model

Deep learning, a branch of machine learning, transforms decision-making by emulating the neural networks seen in the human brain. The system employs an artificial neural network (ANN) of interconnected neurons that manipulate and convert incoming data over multiple layers. In contrast to traditional programming, deep learning acquires knowledge independently by analyzing extensive datasets, revealing intricate patterns and connections. Convolutional neural networks (CNNs) are highly proficient at

Recognizing images, whereas recurrent neural networks (RNNs) are specialized in processing sequential data. The success of deep learning in several fields is attributed to its capability to learn hierarchical data representations. It has led to breakthroughs in artificial intelligence and has opened up new possibilities for innovation, especially with the rise in processing power and data availability[27].

2.4.1. Long Short-Term Memory (LSTM)

LSTM networks, a specific variant of RNNs, perform exceptionally in processing sequential data that contains extensive dependencies across vast distances. In contrast to conventional RNNs, LSTM networks address the issue of disappearing gradients by including a memory cell, which allows them to preserve information for longer durations. LSTMs, equipped with three gates that control the flow of information, can choose to retain, modify, or reject data. This makes them particularly well-suited for natural language processing (NLP) tasks and time seriesforecasting. Their capacity to comprehend complex patterns in sequential data, particularly in medical contexts involving temporal relationships, highlights their importance. In addition, LSTM models can be improved by incorporating attention mechanisms, which can enhance their ability to comprehend intricate sequential material in several areas[27].

$$i_t = \sigma\left(W_i\left[h_{t-1}, x_t\right] + b_i\right) \tag{1}$$

$$f_t = \sigma\left(W_f[h_{t-1}, x_t] + b_f\right) \tag{2}$$

$$o_t = \sigma\left(W_o[h_{t-1}, x_t] + b_o\right) \tag{3}$$

Where i_t = represents the input gate, f_t = forget gate, o_t = represents the output gate, σ = represents a sigmoid functionw_x = weight for the respective gate (x) neurons, h_{t-1} = output of the previous LSTM block (at timestamp t-1), x_t = input at the current timestamp, b_x = biases for the respective gates (x). The equation for the cell state, candidate cell state, and the final output are as follows.

$$C_t = tanh\left(W_c\left[h_{t-1}, x_t\right] + b_c\right) \tag{4}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{5}$$

$$h_t = o_t * tanh\left(C^t\right) \tag{6}$$

Where Ct = Cell state (memory) at timestamp (t)
$\widetilde{C_t}$ = Represents candidate for cell state at timestamp (t).

2.4.2. Gated Recurrent Unit (GRU)

GRU, a type of RNN, tackles issues such as the vanishing gradient problem and difficulties in capturing long-term relationships in sequential data. GRU, a simplified version of the LSTM network, merges the forget and input gates into a single update gate. This consolidation reduces computing complexity while maintaining the ability to propagate information over time. GRUs are highly effective at acquiring knowledge from sequential data, making them well-suited for natural language processing and medical data analysis applications. GRU is well-suited for applications with limited computational resources or severe latency requirements due to quicker training timeframes and lower parameters than LSTMs. Despite these advantages, GRU performs competitively in modeling temporal dependencies and capturing complicated patterns[23].

$$r_t = \sigma\left(W_r\left[h_{t-1}, x_t\right]\right) \quad (7)$$

$$z_t = \sigma\left(W_z\left[h_{t-1}, x_t\right]\right) \quad (8)$$

$$h_t = (1 - Z_t) * h_{t-1} + Z_t * h_t \quad (9)$$

Where σ = represents the sigmoid activation function, X_t= present input, H_{t-1}= = previous hidden state, R_t = Reset gate, Z_t = Update gate, W_r, W_z= Weights for the rest and update gate, H_t= present hidden state output of the present cell, $\hat{h}$= New Cell state.

2.4.3. Bidirectional Long-Short-term Memory (BiLSTM)

BiLSTM networks improve upon typical LSTM architectures by capturing preceding and subsequent context in sequential data. BiLSTM, unlike normal LSTM, employs two distinct LSTM layers that process the input sequence in both the forward and reverse orientations. BiLSTMs enhance their capacity to grasp long-range dependencies and subtle patterns by integrating the outputs from both layers, thus thoroughly comprehending the input. The Bidirectional technique employed by BiLSTM is particularly well-suited for tasks such as NLP and time series forecasting, where the inclusion of context from both directions is essential. BiLSTMs outperform unidirectional LSTMs in understanding sequential data with intricate temporal dynamics and dependencies[26].

$$i_t = \sigma\left(W_i x_t + U_i h_{t-1} + b_i\right) \quad (10)$$

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right) \quad (11)$$

$$o_t = \sigma\left(W_o x_t + U_o h_{t-1} + b_o\right) \quad (12)$$

$$\tilde{C} = f_t \odot C_{t-1} \quad (13)$$

$$h_t = o_t \odot tanh\left(c_t\right) \quad (14)$$

The variables x_t, σ, and c_t represent the input sample at time t, the sigmoid activation function, and the memory unit, respectively. The abbreviations (b_i, b_f, b_o) and (w_i, w_f, w_o) represent each gate's bias and weight matrices, respectively. The symbol ⊙ represents the multiplication operation between the elements. Initially, the h_{t-1}, c_{t-1}, and xt transmit the input data to the LSTM unit. The forward layer $\left(h_t^f\right)$ and the backward layer $\left(h_t^b\right)$ of the BiLSTM model produces the following output:

$$h_t = \alpha h_t^f + \beta h_t^b \quad (15)$$

$$y_t = \sigma\left(h_t\right) \quad (16)$$

Where α and σ are the numerical factors respecting the equality α+σ=1.

2.4.4. One-Dimensional Convolutional Neural Networks (1D-CNN)

1D-CNN is designed to handle sequential data such as time series or text sequences. Contrary to conventional CNNs, these operate on one-dimensional input, extracting local patterns by sliding filters along the sequence. 1D-CNNs capture meaningful patterns at different levels of abstraction by utilizing convolutional processes and activation functions to learn hierarchical representations. Their expertise lies in speech recognition, NLP, and biological signal processing, where analyzing local patterns is paramount. 1D-CNNs are commonly employed in practical scenarios that require sequential data processing due to their effectiveness and ability to handle large-scale datasets[24].

$$y_j = b_j + \sum_{c=0}^{n_c-1} \sum_{k=-p}^{p} x_{c,}\ j-k\ w_{c,}\ \ k \qquad (17)$$

$$\frac{\partial \alpha}{\partial_{x_{c,i}}} = \sum_{k=-p}^{p} \frac{\partial \alpha}{\partial_{y_{i+k}}} w_{c,k} \qquad (18)$$

$$\frac{\partial \alpha}{\partial_{w_{c,k}}} = \sum_{j=0}^{m-1} \frac{\partial \alpha}{\partial_{y_j}} x_{c,j-k} \qquad (19)$$

$$\frac{\partial \alpha}{\partial_{b\,j}} = \frac{\partial \alpha}{\partial_{y_j}} \qquad (20)$$

2.5 EVALUATION METRICS

Four distinct metrics are considered to assess the efficacy of the deep learning model: Accuracy, Precision, Recall, and F1-Score. The ratio of correct predictions to the total number of forecasts. The accuracy metric measures the ratio of accurate predictions to the total number of projections. Accuracy is the measure used to evaluate classification effectiveness [25,26].

$$Accuracy\ (ACC) = (TP+TN)/\ (TP+FP+FN+TN) \qquad (21)$$

Precision refers to the measure of accurate forecasts that determine the quality of predictions. The positive predictions can be determined using the following formula.

$$Precision(P) = TP/(TP+FP) \qquad (22)$$

The recall refers to the proportion of actual examples successfully retrieved from the total number of examples available.

$$Recall(R) = TP/(TP+FN) \qquad (23)$$

The F1 score is a suitable metric to leverage the occurrence. Analogously, the F1 score can be used to observe the relationship between precision and recall, and the issue of jagged class propagation persists.

$$F1-Score = 2*(P*R)/(P+R) \qquad (24)$$

More precisely, when α = 1, the formula for the F1-Score becomes less complex. These formulas allow us to calculate accuracy, precision, recall, and the F1-Score, commonly used metrics for evaluating classification performance[25].

2.6 Compare Performance

The algorithms BiLSTM, LSTM, GRU, and 1D-CNN were utilized, and their performance was compared to determine the most satisfying one.

3. PROPOSED MODEL ARCHITECTURE

Figure 3 displays the layered architecture model consisting of BiLSTM, LSTM, and GRU. The input layer comprises the unprocessed sensor data in the BiLSTM, LSTM, and GRU models. The dropout layer is a regularization approach employed to mitigate overfitting by randomly deactivating a portion of input units to zero during the training process. By implementing dropout, the neural network enhances its resilience and ability to generalize to unfamiliar input, enhancing its performance on real-world tasks. The dense layer receives the concatenated output from the layers and converts it into the appropriate output shape. It is commonly employed for classification or regression tasks. The ultimate layer in the network generates the ultimate predictions or classifications relying on the processed sequential input.

The Bi-LSTM layer incorporates input and output gates that regulate the movement of input and output data within the LSTM cell. The input gate is responsible for selecting and integrating new information into the cell state, while the output gate controls the flow of information to the subsequent layer. The forget gate determines whether information should be excluded from the cell state, assisting the model in disregarding unimportant input. The cell state in a recurrent neural network retains important information over multiple time steps and is modified by gates that consider both current inputs and previous states. Bidirectional processing involves processing each sequence in both the forward and backward directions. The outputs from both directions are then concatenated, allowing for capturing context from past and future data. Within the BiLSTM model, dropout can be implemented on the input and recurrent connections to the LSTM units, reducing interdependencies among neurons. The dropout rates vary between 0.2 and 0.5, indicating the number of units removed during training. The dense layer establishes connections between each neuron and every neuron in the preceding layer, facilitating a thorough integration of the data acquired by the BiLSTM. An activation function, such as ReLU, sigmoid, or softmax, introduces non-linearity and maps the output to the desired range.

LSTM layers employ three gates, namely the input gate, the forget gate, and the output gate, to control the flow of information. These gates ensure that relevant information is retained while irrelevant information is rejected. The memory cell state facilitates the transmission of the information at different time intervals, allowing the network to sustain long-term connections. At each time step, the hidden state and the input are combined to generate an output that can be utilized for subsequent processing or predictions. The sigmoid and hyperbolic tangent functions are used in the gates and memory cells to control and modify the flow of information efficiently. The dropout rate parameter in dropout layers determines the proportion of input units that are randomly dropped during training. This parameter usually ranges from 0.1 to 0.5, with larger values indicating a more aggressive dropout strategy. The dense layer comprises several neurons (units) that are entirely coupled to the output of the LSTM layer. Every individual neuron within the thick layer performs a calculation where it multiplies the output of the LSTM layer by a set of weights and then applies an activation function such as softmax for classification or linear activation for regression.

The GRU layer integrates the forget and input gates into a unified update gate, resulting in decreased computational complexity compared to LSTM. The update gate regulates the quantity of previous information to preserve. The reset gate determines the extent to which previous information should be disregarded. The memory content is refreshed by employing candidate activation and the reset gate, effectively managing long-term dependencies. Dropout is implemented on the inputs, recurrent connections, or both, using a predetermined dropout rate P (e.g., 0.2), indicating the likelihood of a unit being assigned a zero value. By omitting units during training, the network's sensitivity to the individual weights of neurons is reduced, which enhances its ability to handle many situations and make generalizations. Dropout is exclusively implemented during the training phase and not during the testing phase, guaranteeing that the model's performance is not negatively impacted. In a dense layer, each neuron receives input from every neuron in the preceding layer, allowing it to grasp intricate linkages and interactions within the data. The weight matrices and bias vectors are acquired during training to optimize the model's performance on the specified task. Figure 4 (a)-(d) displays the model summary of BiLSTM, LSTM, GRU, and 1D-CNN.

The conv1D layer employs a collection of filters (or kernels) that traverse the input sequence to extract localized patterns. Every filter is implemented over the time dimension of the incoming data. The kernel size determines the extent of the filter by indicating the number of time steps it spans. It affects the extent of the receptive field for the convolutional process. Strides determine the magnitude of the step taken by the filter as it traverses the input sequence. It regulates the amount of displacement that occurs in the filter following each convolutional operation. An activation function, such as ReLU, Sigmoid, or tanh, is usually applied after each convolutional operation to create non-linearity. It allows the network to learn more intricate patterns. Padding is a technique that adds zeros to the input sequence at the boundaries. It can be applied in two ways: the same padding, which maintains the output length, and valid padding, which reduces the output length.

The Maxpooling 1D layer decreases the dimensionality of the input sequence by selecting the highest value within a defined window. This process aids in downsampling the data and decreasing computing complexity. The algorithm moves a window of a predetermined size along the input sequence and returns the highest value within each window. The extent to which the input sequence is shortened is governed by the pool size and the stride. The dropout layer functions by selectively excluding specific nodes from the network during each training iteration, thereby compelling the model to acquire more resilient features. The dropout rate is a hyperparameter that specifies the fraction of neurons to be dropped, usually ranging from 0.2 to 0.5.

The primary function of a dense layer in a 1D CNN is to carry out sophisticated reasoning and decision-making processes using the characteristics that the convolutional and pooling layers have recovered. The dense layer establishes connections between each neuron and every neuron in the preceding layer, allowing the integration of acquired data to form more intricate representations. The activation function used in dense layers includes ReLU and softmax. The softmax function is particularly beneficial for classification applications as it outputs probabilities. The dense layer combines and modifies the characteristics to get the ultimate result, such as class scores in classification problems or predictions in regression assignments.

4. RESULTS

Multiple deep-learning methods are evaluated using breath analysis data from individuals with liver conditions. The confused matrix for LSTM, Bi-LSTM, GRU, and 1D-CNN is depicted in Figure 6(a)-6(d).

The loss function steadily reduces as the number of training epochs increases. The efficiency of the training and testing algorithms gradually improves with an increase in the number of epochs. Figure 7(a) - 7(d) displays the deep learning methods' outcomes in model accuracy and loss. It denotes the mean precision of various trained datasets. Multiple data sets were trained to predict liver disease with an increase. The models are trained using 80% and 20% of the data sets, and the overall accuracy is calculated using the trained data sets. The BiLSTM model demonstrates superior performance, with an impressive average accuracy rate of 99%.

Table 2 shows how accurate, specific, sensitive, precise, recall, and F1-score those suggested algorithms are. The accuracy, specificity, sensitivity, precision, recall, and F1-score were all 0.98, 0.99, and 1; for BiLSTM, precision was 1 when no disease was found and 0.92 when a disease was found.

Recall was 0.99 and 1, and the F1 score was 1 and 0.96. Based on the LSTM, the accuracy is 0.98, the precision is 0.98, and the sensitivity is 0.94. The F1-score is between 0.99 and 0.89, the precision is between 1 and 0.84, and the recall is between 0.99 and 0.95. The GRU has a 0.97 accuracy rate, a 0.99 precision rate, and a 0.96 sensitivity rate. There is a difference between 1 and 0.90 in precision, 0.99 and 0.97 in memory, and 0.99 and 0.93 in F1. The 1D-CNN has an F1-Score of 0.98, an accuracy of 0.98, a specificity of 0.98, a sensitivity of 0.97, a precision of 0.97 and 0.98, a recall of 0.98 for both disease prediction and disease not prediction, and an accuracy of 0.98. The algorithms' results showed that the BiLSTM and 1D-CNN algorithms worked well enough based on different statistical performance measures.

Table 2 Statistical performance measurements

Model	Accuracy	Specificity	Sensitivity	Precision		Recall		F1-score	
				0	1	0	1	0	1
BiLSTM	0.99	0.99	1	1	0.92	0.99	1	1	0.96
LSTM	0.98	0.98	0.94	1	0.84	0.99	0.95	0.99	0.89
GRU	0.97	0.99	0.96	1	0.90	0.99	0.97	0.99	0.93
1D CNN	0.98	0.98	0.97	0.97	0.98	0.98	0.98	0.98	0.98

5. DISCUSSION

Multiple algorithms are evaluated for comparative comparison, including LSTM, GRU, BiLSTM, and 1D-CNN. The results are derived from six primary characteristics: accuracy, specificity, and sensitivity, as depicted in Figure 8. Additionally, precision, recall, and F1-score are illustrated in Figure 9. A graph is created to compare the specified methods. The Y-axis represents the parametric values, while the X-axis represents the corresponding comparison factors. All models have a uniform number of epochs, precisely 20. The batch size for the BiLSTM model is 64, whereas for all other algorithms, it is 32. Among the algorithms mentioned above, BiLSTM achieves the best level of accuracy. This approach achieves exceptional accuracy, with a precision of 0.99 and an F1-Score of 1 for non-infected individuals. It also demonstrates a high F1-Score of 0.96 for infected individuals while maintaining a low testing loss of 0.012. This strategy is recommended for situations that require a satisfactory level of precision while minimizing any potential loss.

Regarding the second algorithm, 1D-CNN, the entire set of test samples was executed and yielded an accuracy of approximately 0.98. However, the testing loss function is 0.028. The subsequent experimental study was conducted on the LSTM model, yielding accuracy levels of 0.98 and an F1-Score of approximately 0.99 for the absence of disease and 0.89 for the presence of disease. The testing loss was measured at 0.066. The subsequent model utilized for the study was the GRU, which achieved an accuracy of 0.97 and an F1-Score of 0.99 for individuals without any disease and 0.93 for individuals with a disease.

To begin with, the combination of BiLSTM and 1D-CNN is highly effective for detecting liver diseases in patients. BiLSTM is the recommended choice in terms of accuracy. A comparative analysis is conducted on the algorithms, and the findings are shown in a chart that provides a comprehensive understanding of model accuracy, loss function, and F1-Score. The researcher selects an algorithm that best suits the user's requirements based on these metrics. Figure 8-9 illustrates the performance comparison of deep learning models for detecting liver illness.

This comparative model utilizes six parameters to assess the efficacy of each deep learning model based on the recorded values for each parameter. BiLSTM and 1D-CNN algorithms significantly advance metric accuracy, supporting their potential for scientific and research applications. Hence, the BiLSTM and 1D-CNN algorithms are regarded as the most effective methods for early-stage liver disease prediction.

6. CONCLUSION

This study report presented many prediction algorithms for early-stage detection and diagnosis of liver disease. The dataset displayed the input parameters followed by the corresponding training models. The accuracy of predicting liver illness was enhanced by assessing the algorithms using a comprehensive collection of attributes and a well-trained dataset. These findings reveal new characteristics that classifiers can use, especially in the early stages of diagnosing liver disease. LSTM, GRU, BiLSTM, and 1D-CNN models are implemented to forecast liver illness. The results indicated that the BiLSTM and 1D-CNN models made precise predictions for patients with liver illness. While other algorithms showed satisfactory results under specific parameters, BiLSTM and 1D-CNN consistently outperformed them in every phase. Therefore, these algorithms are the optimal and most encouraging methods for predicting liver disease. This study aims to offer the medical field, data scientists, and research community a straightforward deep-learning model that can be utilized for the early detection of liver illness. The current dataset is satisfactory, but future efforts should focus on collecting more breath samples.

Additionally, further work is required to develop more precise models that can yield improved outcomes. In the future, it is possible to create a system for implementing and executing the models suggested by the current remarkable progress. Subsequently, the model will be implemented on mobile applications, specifically Android and IOS.

Ethical Statement

This study contains no studies with human or animal subjects performed by any authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest in this work.

Data Availability Statement

Data is available upon reasonable request from the corresponding author.

7 REFERENCES

[1] Sumeet K Asrani, Harshad Devarbhavi, John Eaton, Patrick S Kamath, Burden of liver diseases in the world, *Journal of Hepatol,* volume 70, issue 1, p151-171, January 2019.

[2] The Polaris, Global prevalence, and genotype distribution of hepatitis C virus infection in 2015: a modeling study 2015, *Lancet Gastroenterol Hepatol* 2017; 2: 161–76.

[3] Christina Fitzmaurice, The Burden of Primary LiverCancer and underlying etiologies from 1990 to 2015 at the global, regional and national level: Results from the Global Burden of Disease Study 2015, *JAMA Oncology,* 2017; 3(12):1683-1691.

[4] Robert Scott, Indra Neil Guha, Non-invasive monitoring of liver fibrosis, *Br Med Bull.* 2014 Dec;112(1):97-106.

[5] Antonio De Vincentis, Umberto Vespasiani-Gentilucci,

Anna Sabatini, Raffaele Antonelli-Incalzi, Antonio Picardi, Exhaled breath analysis in hepatology: state of the art and perspectives," *World J Gastroenterol* 2019 August 14; 25(30): 4043-4050.

[6] Marc Philippe van der Schee, Tamara Paff, Paul Brinkman, Willem Marinus Christiaan van Aalderen, Eric Gerardus Haarman, Peter Jan Sterk,Breathomics in lung disease, *Chest. 2015* Jan;147(1):224-231.

[7] Ronny Schnabel, Rianne Fijten, Agnieszka Smolinska, Jan Dallinga, Marie-Louise Boumans, Ellen Stobberingh, Agnes Boots, Paul Roekaerts, Dennis Bergmans, Frederik Jan van Schooten, analysis of volatile organic compounds in exhaled breath to diagnose ventilator-associated pneumonia, *Scientific Reports* volume 5, Article number: 17179, 2015

[8] Sagnik Das, Mrinal Pal, Non-invasive monitoring of human health by exhaled breath analysis: A comprehensive review, *Journal of The Electrochemical Society*, Volume 167, Number 3,2020.

[9] Pedro Catalao Moura, Maria Raposo, Valentina Vassilenko, Breath VOCs as biomarkers for the diagnosis of pathological conditions: A review, *Biomedical Journal*, Volume 46, Issue 4, August 2023, 100623.

[10] Or Herman-Saffar, Zvi Boger, Shai Libson, David Lieberman, Raphael Gonen, Yehuda Zeiri, Early non-invasive detection of breast cancer using exhaled breath and urine analysis, *Computers in Biology, and Medicine*, Volume 96, 1 May 2018, Pages 227-232

[11] Alexander Aliper, Sergey Plis, Artem Artemov, Alvaro Ulloa, Polina Mamoshina and Alex Zhavoronkov, Deep learning applications for predicting pharmacological properties of drugs and drug repurposing using transcriptomic data, *Molecular Pharmaceutics* Volume 13, Issue 7 July 5, 2016, Pages2141-2602.

[12] Suchi Saria, Anna Goldenberg, What it is and its role in precision medicine, *IEEE Intelligent Systems*, 2015.

[13] Naim Alkhouri, Frank Cikach, Katharine Eng, Jonathan Moses, Nishaben Patel, Chen Yan, Ibrahim Hanouneh, David Grove, Rocio Lopez, Raed Dweik, Analysis of breath VOC as a non-invasive tool to diagnose non-alcoholic fatty liver disease in children, *Eur J Gastroenterol Hepatol*. 2014 Jan; 26(1):82-7.

[14] B de Lacy Costello, A Amann, H Al-Kateb, C Flynn, W Filipiak, T Khalid, D Osborne, N M Ratcliffe, A review of the volatiles from the healthy human body, *J Breath Res*. 2014 Mar;8(1):014001.

[15] Naim Alkhouri, Tavankit Singh, Eyad Alsabbagh, John Guirguis, Tarek Chami, Ibrahim Hanouneh, David Grove, Rocio Lopez, Raed Dweik, Isoprene in the exhaled breath is a novel biomarker for advanced fibrosis in patients with chronic liver disease: A pilot study, *Clin Transl Gastroenterol* 2015 Sep 17; 6(9):e112.

[16] Rohit Sinha, Khalida A. Lockman, Natalie Z.M, Homer, Edward Bower, Paul Brinkman, Hugo H. Knobel, Jonathan A. Fallowfield, Alan J. Jaap, Peter C. Hayes, John N. Plevris, Volatomic analysis identifies compounds that can stratify non-alcoholic fatty liver disease, *JHEP Reports* 2020 vol. 2 j 100137.

[17] Giuseppe Ferrandino et al. Breath biopsy assessment of liver disease using an exogenous volatile organic compound toward improved detection of liver impairment, *Clin Transl Gastroenterol*. 2020 Sept.11(9):e00239.

[18] Owlstone Medical, Breathprinting of liver diseases, 22 Aug 2019, Under *Cardiovascular & metabolic disease*.

[19] Mikolaj Wieczorek, Alexander Weston, Matthew Ledenko, Jonathan Nelson Thomas, Rickey Carter, Tushar Patel, A deep learning approach for detecting liver cirrhosis from volatolomic analysis of exhaled breath, *Front Med (Lausanne*), 2022 Sep 29 :9:992703.

[20] Kirsten E. Pijls, Agnieszka Smolinska, Daisy M. A. E. Jonkers,

Jan W. Dallinga, Ad A. M. Masclee, Ger H Koek, Frederik-Jan van Schooten, A profile of volatile organic compounds in exhaled air as a potential non-invasive biomarker for liver cirrhosis, *Scientific Reports* volume 6, Article number: 19903(2016).

[21] Rakesh Kumar Patnaik, Yu-Chen Lin, Ming Chih Ho, J. Andrew Yeh, Selection of consistent breath biomarkers of abnormal liver function using feature selection: a pilot study, *Health and Technology*, Volume 13, pages 957–969, (2023).

[22] Giuseppe Ferrandinoet.al. Breath biopsy To identify exhaled volatile organic compounds biomarkers for liver cirrhosis detection, *J Clin Transl Hepatol*. 2023 Jun 28; 11(3):638-648.

[23] Saeed Mohsen, Recognition of human activity using GRU deep learning algorithm, *Multimedia Tools, And Applications*, 82:47733–47749, 2023.

[24] Dechun Zhao, Renpin Jiang, Mingyang Feng, Jiaxin Yang, Yi Wang, Xiaorong Hou, Xing Wang, “A deep learning algorithm based on 1D CNN-LSTM for automatic sleep staging”, *Technology and Health Care* 30 (2022) 323–336.

[25] Oona Rainio, JarmoTeuho, Riku Klén, Evaluation metrics and statistical tests for machine learning, *Scientific Reports*, (2024) 14:6086.

[26] Dewanand A Meshram, Dipti D Patil, Predicting Sour or Sweet: Exploring Advance DL Methods for Odor Perception Based on Molecular Properties, *International Journal of Intelligent Systems and Applications in Engineering*, 2023/9/6

[27] Gulbakshee Dharmale, Dipti Patil, Swati Shekapure, Aditi Chougule, AI-Based Medicine Intake Tracker, *Nature-Inspired Methods for Smart Healthcare Systems and Medical Data*, 2023/12/2