https://doi.org/10.48047/AFJBS.6.Si4.2024.5719-5732

African Journal of Biological Sciences

Journal homepage: http://www.afjbs.com

Research Paper Open Access

RISK FACTORS IDENTIFICATION OF THYROID DISEASE USING DEEP LEARNING WITH FEATURE SELECTION APPROACH

[1]Selva Banu Priya T, [2]Rajabhushanam, [3]Lakshmi Krishnasamy, [4]T. Puhazhendhi[*]

[1]Research Scholar, Bharath Institute of Higher Education & Research, Chennai-600073, India. priya8517@gmail.com.

[2]Professor, Department of Computer Science & Engineering, Bharath Institute of Higher Education & Research, Chennai-600073, India rajabhushanamc@bharathuniv.ac.in.

[3]Professor, Department of Microbiology Sree Balaji Medical College and Hospital Bharath Institute of Higher Education and Research laksh45@gmail.com

[4]Department of Public Health Dentistry, Sree Balaji Dental College and Hospital, BIHER University, Pallikaranai, Chennai - 100. drpugalbds@gmail.com

Volume 6, Issue Si4, Aug 2024

Received: 15 June 2024

Accepted: 25 July 2024

Published: 15 Aug 2024

doi: 10.48047/AFJBS.6.Si4.2024.5719-5732

Abstract

The purpose of this research is to use deep learning and feature selection methods to discover potential causes of thyroid illness. We use the UCI DL Repository dataset, which has 2,800 occurrences and 28 characteristics, and we use Boruta and Recursive Feature Elimination (RFE) techniques to carefully preprocess and refine it in order to extract relevant features. Results show that RNN achieved an exceptional recall rate of 96.01% and accuracy scores above 98% after extensive examination across six DL algorithms, including Autoencoder and Long Short Term Memory Networks. Although there have been some accomplishments, there are still obstacles. One of these is that Multilayer perceptron continually has lower accuracy levels. In healthcare analytics, where even small gains in recall and accuracy can have a huge effect on diagnostic performance and patient outcomes, our results highlight the vital importance of strong preprocessing and feature selection methods. To further improve the accuracy of classification and the refinement of thyroid disease risk factor identification, future studies may investigate hybrid model architectures and innovative feature engineering techniques.

Keywords: Thyroid disease, risk factors identification, feature selection, deep learning

1. Introduction

Thyroid disease represents a prevalent endocrine disorder characterized by dysregulation in the production or function of thyroid hormones. The thyroid gland, a vital component of the endocrine system, plays a crucial role in regulating metabolism, growth, and energy expenditure. Dysfunction in thyroid hormone synthesis or secretion can lead to a spectrum of clinical manifestations, ranging from subclinical abnormalities to overt thyroid disorders such as hypothyroidism, hyperthyroidism, thyroid nodules, and thyroid cancer [1].

Identifying individuals at risk of developing thyroid disease is paramount for early intervention, effective management, and prevention of associated complications. Traditionally, risk assessment for thyroid disorders relies on clinical evaluation, biochemical testing, and imaging modalities. However, the complex interplay of genetic predisposition, environmental factors, and lifestyle variables necessitates a comprehensive and data-driven approach for accurate risk prediction [2].

In recent years, the advent of deep learning (DL) techniques has revolutionized healthcare analytics by enabling the extraction of meaningful insights from large and heterogeneous datasets. DL algorithms, particularly those employing feature selection strategies, offer a promising avenue for uncovering latent patterns and risk factors associated with thyroid disease. Feature selection techniques aim to identify the most relevant subset of predictors that contribute significantly to disease prediction while mitigating the curse of dimensionality and enhancing model interpretability [3].

This introductory discourse endeavors to elucidate the pivotal role of deep learning with feature selection in the identification of risk factors for thyroid disease. By synthesizing existing literature and elucidating key concepts in thyroid pathophysiology and predictive modeling, this narrative seeks to underscore the significance of data-driven approaches in precision medicine and clinical decision-making.

The multifactorial etiology of thyroid disease underscores the intricate interplay between genetic susceptibility and environmental influences [4]. Genetic polymorphisms within genes encoding thyroid hormone receptors, iodine transporters, and thyroid peroxidase have been implicated in the pathogenesis of autoimmune thyroid disorders such as Hashimoto's thyroiditis and Graves' disease [5]. Furthermore, epidemiological studies have elucidated the impact of environmental factors, including iodine deficiency, smoking, radiation exposure, and dietary habits, on thyroid function and disease susceptibility [6].

Despite advancements in genomic profiling and molecular epidemiology, the elucidation of causative genetic variants and environmental triggers remains a daunting challenge in thyroid research. Deep learning approaches offer a complementary framework for integrating diverse data modalities and deciphering the intricate relationships between genetic, clinical, and environmental factors underlying thyroid pathophysiology [7].

Central to the application of deep learning in thyroid disease risk assessment is the process of feature selection, which entails the identification of informative predictors from a pool of candidate variables [8]. Feature selection algorithms encompass a spectrum of techniques, ranging from filter methods

based on statistical significance to wrapper methods employing model-based evaluations [9].

Univariate feature selection methods, such as chi-squared test and analysis of variance (ANOVA), evaluate the association between individual features and the target variable, thereby facilitating the identification of relevant biomarkers and risk factors [10]. However, univariate approaches may overlook interactions and synergistic effects among variables, thereby limiting their efficacy in capturing complex relationships inherent in thyroid disease pathogenesis.

In contrast, wrapper methods iteratively assess subsets of features using predictive models and optimize performance metrics such as accuracy or area under the receiver operating characteristic curve (AUC-ROC) [11]. Recursive feature elimination (RFE) algorithms, a subclass of wrapper methods, sequentially eliminate less informative features based on their impact on model performance, thereby prioritizing discriminative variables and enhancing predictive accuracy [12].

Tree-based ensemble methods, including Long Short Term Memory Networks and Generative Adversarial Network machines, offer inherent feature selection capabilities by evaluating the importance of variables in predictive modeling [13]. These algorithms leverage the collective wisdom of Autoencoders to quantify the relative contribution of each feature to model performance, thereby facilitating the identification of salient risk factors and biological correlates of thyroid disease [14].

The integration of feature selection techniques with deep learning algorithms holds immense potential for elucidating novel biomarkers and etiological pathways in thyroid disease. By harnessing the power of computational analytics and data-driven inference, researchers can unravel the intricate interplay of genetic, clinical, and environmental determinants underlying thyroid pathophysiology [15].

Moreover, deep learning models offer unparalleled flexibility in accommodating heterogeneous data sources and diverse data modalities, ranging from structured electronic health records to unstructured clinical narratives and imaging data [16]. The integration of multimodal data streams enables comprehensive phenotyping and risk stratification, thereby empowering clinicians with actionable insights for personalized patient care and disease management.

In summary, the convergence of deep learning with feature selection heralds a new era in thyroid disease research and clinical practice. By harnessing the synergistic potential of computational analytics, bioinformatics, and translational research, we can unravel the complex etiology of thyroid disorders and advance precision medicine paradigms for improved patient outcomes and population health. Through collaborative interdisciplinary endeavors and rigorous validation frameworks, we can harness the transformative power of data-driven insights to mitigate the burden of thyroid disease and foster a future of proactive and personalized healthcare interventions.

2. Methodology

2.1 Dataset description

Dataset obtained from the UCI DL Repository. With a total of 28, it has 2,800 occurrences. Twenty of the thirty-eight characteristics are of a more general type; these include: questions about thyroxine, about antithyroid medication, about being sick, about being

pregnant, about having thyroid surgery, about receiving I131 treatment, about having hypothyroid or hyperthyroid symptoms, about lithium, about having a goitre or tumour, about having hypopituitary problems, about mental health, about having TSH, T3, TT4, T4U, FTI, or TBG measurements taken, and finally, about seeking a referral. For the most part, they can be categorised as either true or untrue. A total of six characteristics are continuous, as shown in Table 1. One patient in the overall population is classified as having hyperthyroidism, while the other is classified as normal, according to the results of the diagnostic tests. Although 23% percent of the samples tested negative for thyroid disease, 77% tested positive. There are certain features that are missing values, denoted by a question mark ("?").

Table 1 Values that are continuous in the dataset

Parameters	**Parameter type**	**Value in Range**
Age	Patient Age	10-45
TSH	Thyroid-stimulating hormone	0.0 – 40
T3	Tri-iodo-thyronine	0.01 – 15
TT4	Blood throxine level	1 – 450
T4U	Rate of thyroxine used	0.20 – 2.20
FTI	Detecting TD using thyroxine	1 – 350

2.2. Model diagram

In addition to identifying the essential features of TD, this study also developed a highly predictive model. A 7th-generation Intel Core i5 processor with 8 Gradient Boost of RAM was used for all the trials on a laptop. Python v3.9.10 was used to write and execute all the required code in Jupiter Notebook. The Numpy, Pandas, Matplotlib, and Sklearn libraries were utilised for the dataset analysis. The data gathering, cleaning, preprocessing, feature engineering, splitting, model construction, and outcome prediction are the seven processes that make up this system's workflow. The framework's primary process started with gathering data. During the data cleaning step, any rows with full null or zero values, as well as duplicate rows and features with more than 70% missing values, were removed. Following the removal of outliers and the imputed median values for missing values, the dataset was scaled; the last step of preprocessing was to balance the dataset. In the fifth stage of our approach, known as feature engineering, we employ three different methods to identify the most suitable features. After that, an 80:20 partition was used to divide the dataset. In the sixth stage of our system design, we used many DL techniques to train and evaluate the model. At last, the classifier decides if a person has TD. The model diagram of the suggested system is shown in Figure 1.

2.3. Preprocessing

The data preparation process always has a major influence on the generalizability performance of DL algorithms. Research datasets are often imperfect because of things like noise, distortions, and missing values. The DL algorithms struggle to produce reliable predictions since the dataset is biased because of the inconsistent data. Some missing values and unusual values are also present in the dataset that was utilised to do this investigation. Many different methods were used to get the dataset ready. Due to its inaccurate information and more than 70% missing values, the "TBG" field was removed from the dataset. In this study, the two categories are represented by 0 and 1, and the median values are utilised to fill in the missing data. You can see that this dataset has outliers in Table 1. Discovering and eliminating outliers is, thus, the preferred course of action. To filter out extreme values, we employ the IQR, or interquartile range. Scaling features based on their distance from the origin is an important part of preprocessing when DL algorithms are used. In order

to scale characteristics following the removal of outliers and missing data, the robust scaler approach is employed. Due to the dataset's extreme imbalance, SMOTE used to restore its equilibrium.

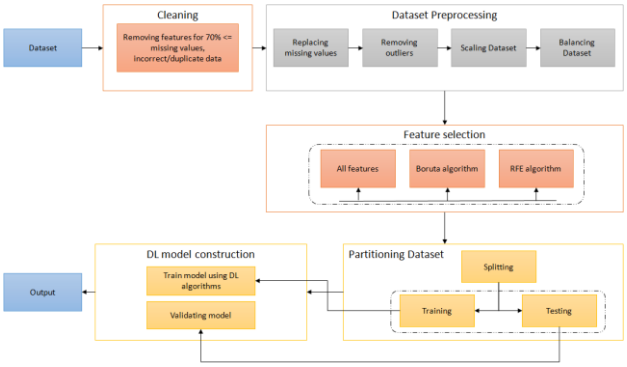

Fig. 1 The proposed system's model diagram

2.4. Feature selection

Feature selection has the potential to improve model learning, reduce calculation time, and overfitting. In order to increase classification accuracy, reduce processing cost, and extract the best features for classification, feature selection often involves picking the most significant characteristics from a dataset while eliminating redundant or unneeded ones. Our suggested system uses three feature selection approaches—Boruta and RFE—to find the best subset of features for performance optimisation.

2.5. Boruta algorithm

The Boruta algorithm is an encapsulation of the Long Short Term Memory Networks classification method. To determine significance, it uses the Z score. The Z score needs an outside reference, so it can't be used to find the significance of any one character. The information system has to have some random attributes added to it for this to work. Each randomly selected attribute should have a corresponding "shadow" property whose value is determined by dispersing the initial attribute's relevance over all instances. After incorporating these shadow features into the initial dataset, train the system using an Long Short Term Memory Networks classifier. The model's most important unique features are those that outweigh the most striking shadow feature.

2.6. Recursive feature elimination (RFE)

In order to choose relevant attributes, the RFE methodology is often used. This method streamlines a model by removing superfluous details and focusing on the most important ones. In order to achieve peak performance, the selection procedure iteratively eliminates characteristics that aren't crucial. By applying an arbitrary attribute or callable, the significance of each outcome is established after training the estimating model with the original set of features. The next step is to remove the characteristics that aren't crucial from the existing set of traits. The process then iteratively continues on the compressed set until the target feature selection is reached. By integrating CV and RFE, we can score many feature subsets and identify the optimal features, allowing us to choose the best-scoring set.

The cost function looks like this:

$$b(\theta)=\frac{1}{S}\sum_{j=1}^{n}(z_i - z_k)^2+\alpha\sum_{j=1}^{S} \left| b_k \right| \quad (1)$$

This is where S represents the number of rows, n stands for the number of columns, z_i i signifies the training value, $z_k$signifies the predicted value, $\propto$ denotes the hyperparameter, and b_k is the coefficient of the k-th feature.

3. Balancing dataset

The unbalanced dataset must be balanced in order to improve DL accuracy. When one of the target class labels has a significantly lower number of

observations compared to the other class labels, the dataset is considered imbalanced. When a dataset is imbalanced or an uncommon event occurs, it will be difficult to create a meaningful and effective prediction model because of the lack of data..

3.1. Synthetic minority oversampling technique

The healthcare business frequently uses SMOTE to overcome class-imbalance issues. Instead of duplicating instances, synthetic ones were created to make sure the data was distributed evenly. One of the most important steps in making synthetic samples using these methods—the RNN algorithm—is computing the distance between instances. The SMOTE sampling process begins with selecting a cluster of nearby instances in the feature space, continues with drawing a line between them, and then selects a point on that line to create a new sample.

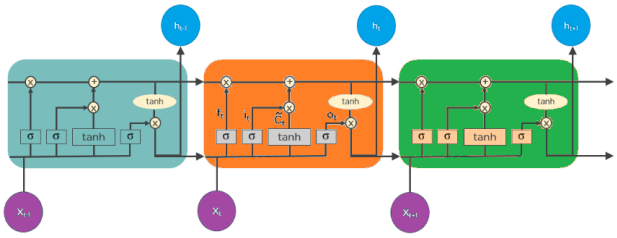

Figure 2: Algorithms used Long Short Term Memory Network (LSTM)

Since Long Short-Term Memory (LSTM) is a specific sort of recurrent neural network (RNN) architecture that aims to solve the issue of the vanishing gradient problem by including a more intricate memory cell structure. LSTMs differ from standard RNNs by incorporating a gating mechanism that controls the transmission of information. This enables them to effectively capture relationships that span over extended sequences of data. An LSTM cell consists of a cell state that acts as a memory unit, capable of retaining information across lengthy sequences. It also includes a forget gate, responsible for determining which information to discard from the cell state, an input gate that decides what new information to incorporate into the cell state, and an output gate that regulates the information to be produced by the cell. LSTMs are highly successful in tasks that involve processing sequential data, such as natural language processing, time series prediction, and speech recognition. This is because LSTMs have a complex architecture that allows them to accurately represent contextual dependencies, which is critical for achieving correct results in these tasks. Figure 2 is a schematic depicting the Long Short Term Memory Networks method.

3.2. Autoencoder

An autoencoder is an artificial neural network that is specifically designed for unsupervised learning tasks, with a special focus on data reduction and feature extraction. The system is comprised of two primary components: an encoder and a decoder. The encoder condenses the input data into a latent representation, usually with a smaller dimensionality than the input, capturing its fundamental characteristics. The decoder subsequently reconstructs the initial input data from this compressed form. Autoencoders are taught to minimise the discrepancy between the input and output data, effectively acquiring a condensed and significant representation of the input. These algorithms are used in a range of domains including image denoising, anomaly detection, and dimensionality reduction. As seen in Figure 3, the Autoencoder algorithm is depicted schematically.

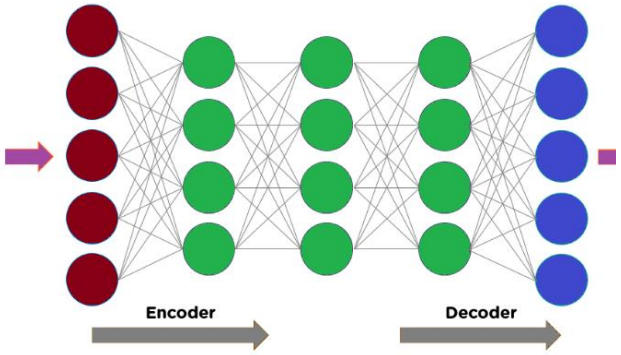

Fig. 3 Autoencoder algorithm.

3.3. Multilayer perceptron

A multilayer perceptron (MLP) is an artificial neural network that consists of many layers of linked nodes, sometimes referred to as neurons. The neural network is composed of an input layer, one or more hidden layers, and an output layer. Every neuron in the MLP applies an activation function to its input, usually a non-linear function such as the rectified linear unit (ReLU) or sigmoid function. This incorporates non-linearity into the model, allowing it to learn intricate correlations in the data. During the training process, the Multilayer Perceptron (MLP) modifies the weights of the connections between its neurons using the backpropagation and gradient descent algorithms. The goal is to minimise a predetermined loss function, such as mean squared error or cross-entropy. MLPs are extensively employed in diverse machine learning applications, such as classification, regression, and pattern recognition, because of their capacity to acquire complex patterns and representations from data.

3.4. RNNs

Recurrent Neural Networks (RNNs) are a type of artificial neural networks specifically created to handle sequential data processing. This makes them very efficient for applications like time series prediction, natural language processing, and speech recognition. RNNs, in contrast to feedforward neural networks, possess recurrent connections that enable them to retain a recollection of past inputs. RNNs has the ability to capture temporal relationships in data, allowing them to excel in tasks that require consideration of context and order. Traditional recurrent neural networks (RNNs) encounter the vanishing gradient problem, which poses a difficulty in learning long-term relationships. In order to tackle this problem, researchers have created advanced models like as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), which provide enhanced memory retention and learning capacities. Recurrent Neural Networks (RNNs) have been extensively used in several domains including text production, machine translation, sentiment analysis, and stock market prediction. Figure 4 represents schema of RNN algorithm

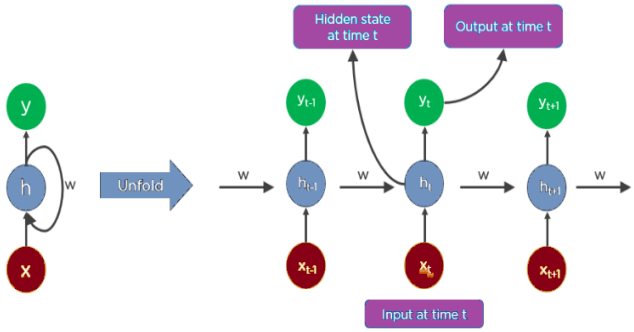

Fig. 4 RNN algorithm

3.5. Deep belief networks

A Deep Belief Network (DBN) is a type of artificial neural network that consists of multiple layers of hidden units, with connections between layers but not within layers. It is structured in a deep, hierarchical manner, with each layer learning increasingly abstract

and complex features from the input data. DBNs are typically composed of a stack of Restricted Boltzmann Machines (RBMs), where the bottom layer is the input layer and the subsequent layers are hidden layers. Training a DBN involves a two-phase process: unsupervised pre-training, where each RBM is trained layer by layer using contrastive divergence or other learning algorithms, followed by supervised fine-tuning using backpropagation to adjust the network's parameters for specific tasks such as classification or regression. DBNs have been successfully applied in various domains, including image recognition, natural language processing, and speech recognition, due to their ability to capture intricate patterns and hierarchical representations in data.

3.6. Generative Adversarial Network

A General Adversarial Network (GAN) is a type of deep learning architecture that comprises two neural networks, the generator, and the discriminator, engaged in a competitive learning process. The generator creates synthetic data samples, such as images or text, aiming to generate outputs that are indistinguishable from real data, while the discriminator's role is to differentiate between real and fake data. The networks are trained simultaneously in an adversarial manner, with the generator striving to fool the discriminator, and the discriminator becoming more adept at distinguishing real from fake data over time. This iterative process leads to the generator producing increasingly realistic outputs as training progresses. GANs have applications in various domains, including image generation, style transfer, data augmentation, and anomaly detection. In Fig. 5 we can see the GAN's schematic.

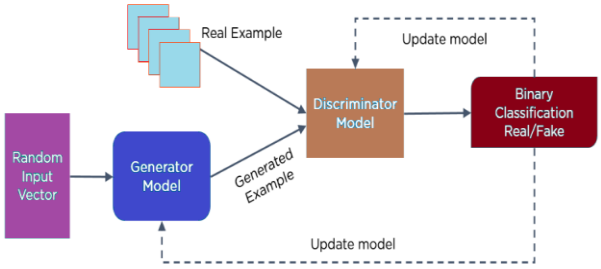

Fig. 5 General Adversarial network algorithm

Table 2 Complexity of training and prediction methods for DL

Models	**Train**	**Prediction**
Long Short Term Memory Networks	P(o^2qo_t)	P(qo_t)
Multilayer perceptron	P(o^2q+o^3)	P(qo_{sv})
Deep belief network	P(oqo_t)	P(qo_t)
Autoencoder	O(n2 p) P(o^2q)	P(q)
Generative Adversarial Network	P(oqo_t)	P(qo_t)
RNNs	N/A	P(oq)

3.7. Computational complexity

A subfield of computer science known as "computational complexity" studies algorithms with an eye towards how much processing power is required to operate or execute them. For algorithms' time complexity, Big P notation is the de facto norm. The training complexity of RNN algorithms is zero. An expression of o, where o is the input size and q is the number of attributes, is a common way to indicate complexity. The number of trees is represented by o_t, while the support vectors are denoted by nsv. The algorithms utilised in this investigation are listed in Table 2 along with their training and prediction complexity.

4. Evaluation metrics

Each study is assessed using a variety of metrics, each with its own distinct importance. A true positive (TP) is the outcome that is produced by a prediction model when that model's predictions end up being correct. A prediction system produces a True Negative (TN) if its predictions are, in fact, incorrect. A predictive model produces what seems like real results but is actually incorrect information; this is called a false positive (FP). A false negative (FN) is the outcome of a prediction model that is misleading but actually accurate..

Accuracy: This metric shows what percentage of the input samples were correct for the predictions.

$$\text{accuracy} = \frac{\text{Number of correct prediction}}{\text{Total number of predictions}}$$

(2)

Precision: The fraction of correct positive predictions is defined by the accuracy. It is determined by comparing the proportion of true positive results to the number of positive results that the classifier had predicted.

$$\text{precision} = \frac{TP}{TP + FP}$$

(3)

Recall: The percentage of TP that was misclassified is typically calculated. The proportion of TP to the sum of TP and FN.

$$\text{recall} = \frac{TP}{TP + FN}$$

(4)

F1-score: It entails testing a binary classification model using positive class predictions. It is computed using Precision and Recall. F1-scores range from zero to one. It shows the classifier's resilience and accuracy, or the number of examples it correctly labels.

$$F1 - \text{score} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

(5)

5. Result and discussion

Classification models and their results were analysed from several angles in this part of the research. We used six DL algorithms—RNN, Long Short Term Memory Networks, Autoencoder, Multilayer perceptron, Deep belief network, and Generative Adversarial Network—in this work. Table 3 displays the results of applying these classification algorithms to a 5-fold CV of the parameters. Our results using all features were first shown, and then we narrowed it down to the most important ones.

Table 3 Variations in the study's algorithms' parameters

Models	**Attributes**
RNN	k=5
Long Short Term Memory Networks	Default
Autoencoder	Depth=1, Criteria = “gini”, state = 1
Multilayer perceptron	Default
Deep belief network	random_state=1 Total estimate = 50, Rate of learning = 0.5, Estimator=1, state = 1
Generative Adversarial Network	Rate of learning = 0.1

5.1. Experimental results with all features

We run DL classifier comparisons on our dataset's full set of features. Some classifiers did well on evaluation metrics, whereas others did not. Ensemble models that are based on trees, such Autoencoder and Long Short Term Memory Networks, have incorporated this data.

We also make use of the Generative Adversarial Network and Deep belief network tree-based boosting models. Table 4 presents the final verdict for feature-based classification across various deep learning (DL) models. The table outlines the performance metrics including Accuracy (Accu %), Precision (Prec %), Recall (Recall %), and F1 score (F1score %) for each model. Among the models, RNN achieved an accuracy of 95.35%, with precision at 62.79%, recall at 79.99%, and an F1 score of 65.07%. Long Short Term Memory Networks exhibited an accuracy of 97.64%, with precision at 81.02%, recall at 76.03%, and an F1 score of 76.92%. Autoencoder model achieved an accuracy of 96.34%, precision at 76.10%, recall at 72.58%, and an F1 score of 73.47%. Multilayer perceptron (MLP) showed an accuracy of 75.78%, precision at 52.60%, recall at 72.69%, and an F1 score of 51.82%. Deep belief network attained an accuracy of 97.28%, precision at 84.67%, recall at 75.29%, and an F1 score of 80. Generative Adversarial Network emerged with the highest accuracy at 97.69%, precision at 81.33%, recall at 81.05%, and an F1 score of 81.1. These metrics collectively depict the comparative performance of each model in the feature-based classification task, highlighting Generative Adversarial Network as the most effective model in terms of accuracy and F1 score, while RNN showed comparatively lower precision and F1 score despite a good recall rate. Multilayer perceptron achieves the worst performance (Fig. 6).

Table 4 Final verdict for feature-based classification across all DL models

Models	**Accu (%)**	**Prec (%)**	**Re$_{call}$ (%)**	**F1$_{score}$ (%)**
RNN	95.35	62.79	79.99	65.07
Long Short Term Memory Networks	97.64	81.02	76.03	76.92
Autoencoder	96.34	76.10	72.58	73.47
Multilayer perceptron	75.78	52.60	72.69	51.82
Deep belief network	97.28	84.67	75.29	80
Generative Adversarial Network	97.69	81.33	81.05	81.1

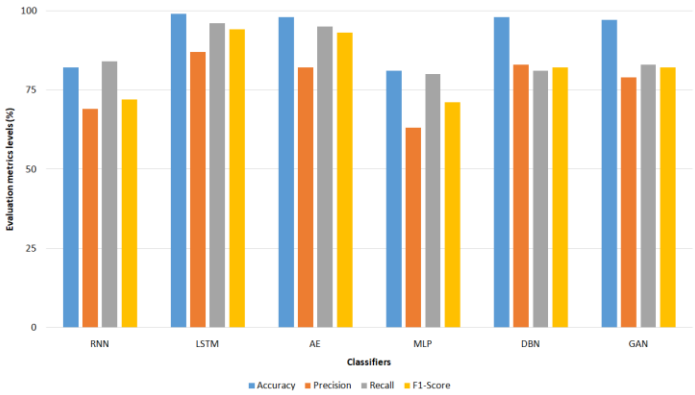

Fig. 6 All features assessment metrics compared in a graph

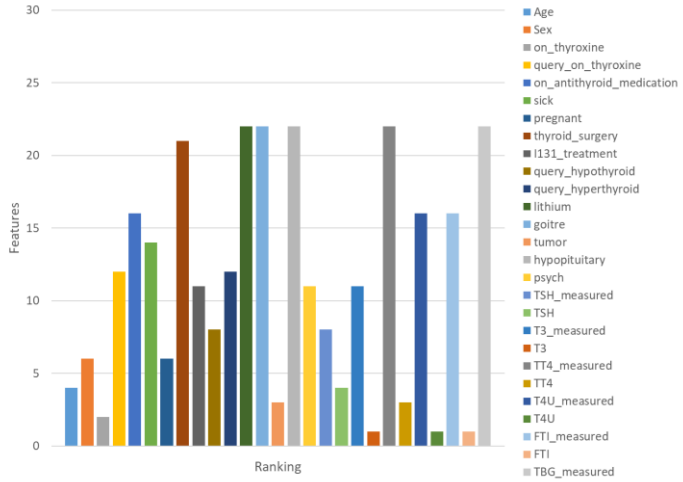

Fig. 7 Rankings for Boruta

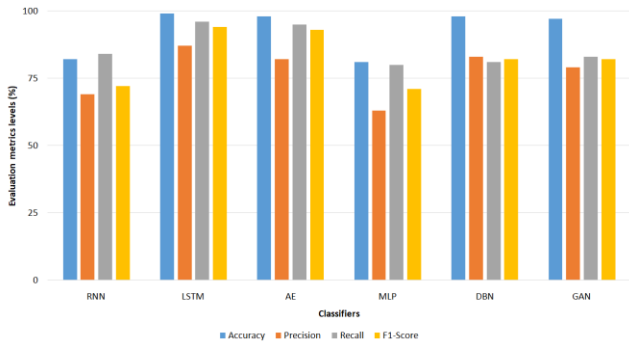

Fig. 8 Boruta feature selection for comparing several evaluation metrics

Table 5 The algorithms' performance metrics following Boruta

Models	**Accu (%)**	**Prec (%)**	**Re$_{call}$ (%)**	**F1$_{score}$ (%)**
RNN	94.39	67.1	90.83	73.26
Long Short Term Memory Networks	98.64	95.97	89.71	94.09
Autoencoder	99.45	90.43	82.64	86.82
Multilayer perceptron	93.2	64.02	83.65	67.21
Deep belief network	97.53	83.68	77.73	79.26
Generative Adversarial Network	97.80	82.86	75.28	78.45

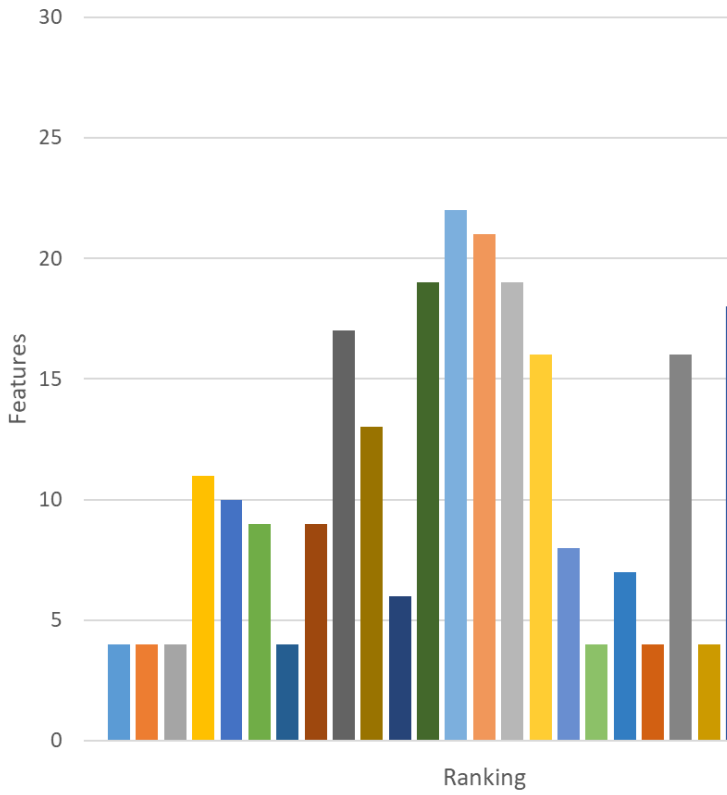

Fig. 9 The ranking of RFE features

Table 6 The algorithms' performance metrics following RFE

Models	**Accu (%)**	**Prec (%)**	**Re$_{call}$ (%)**	**F1$_{score}$ (%)**
RNN	96.93	75.93	96.01	82.10
Long Short Term Memory Networks	98.24	88.89	93.26	90.62
Autoencoder	98.53	87.60	91.68	89.36
Multilayer perceptron	93.45	61.48	91.93	67.71
Deep belief network	97.35	84.55	81.33	81.67
Generative Adversarial Network	97.98	82.45	84.83	83.66

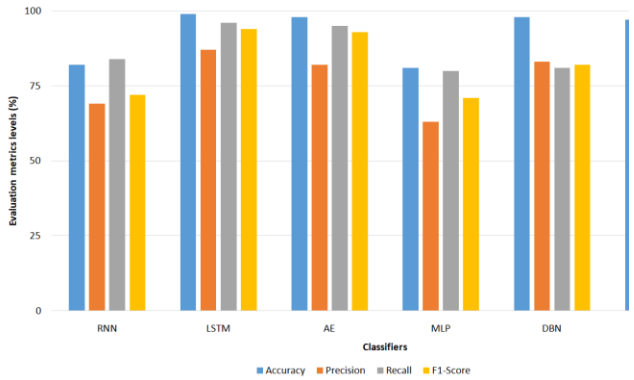

Fig. 10 Recursive Feature Elimination (RFE) Feature Selection for Evaluation Metrics Comparison

5.2. Experiment outcomes using Boruta-selected characteristics

Figure 7 shows the four relevant thyroxin characteristics: T4U, FTI, and T3. These were selected using the Boruta feature selection algorithm. When running Boruta, only features with a rank of one are taken into account; features with a rank higher than one are ignored. Then, we use DL classifiers trained with the SMOTE method to check how well these attributes work. Training and evaluating the classifiers follows the elimination of the least significant features identified by Boruta. The classifiers' performance saw a considerable improvement once the superfluous data was removed. Table 5 provides an overview of the algorithms' performance metrics subsequent to Boruta feature selection. The table showcases the Accuracy (Accu %), Precision (Prec %), Recall (Recall %), and F1 score (F1score %) for each model. RNN achieved an accuracy of 94.39%, precision at 67.1%, recall at 90.83%, and an F1 score of 73.26%. Long Short Term Memory Networks exhibited the highest accuracy among the models at 98.64%, with precision at 95.97%, recall at 89.71%, and an F1 score of 94.09%. Autoencoder model attained an accuracy of 99.45%, precision at 90.43%, recall at 82.64%, and an F1 score of 86.82%. Multilayer perceptron (MLP) demonstrated an accuracy of 93.2%, precision at 64.02%, recall at 83.65%, and an F1 score of 67.21%. Deep belief network achieved an accuracy of 97.53%, precision at 83.68%, recall at 77.73%, and an F1 score of 79.26. Generative Adversarial Network emerged with an accuracy of 97.80%, precision at 82.86%, recall at 75.28%, and an F1 score of 78.45. These metrics collectively illustrate the comparative performance of each algorithm post-Boruta feature selection, indicating Autoencoder as the top-performing model in terms of accuracy, precision, and F1 score, while Long Short Term Memory Networks exhibited the highest precision and F1 score despite slightly lower recall compared to Autoencoder. Once again, Multilayer perceptron did not do very well.

5.3. Experiment outcomes using RFE

Figure 9 demonstrates that six different types of DL classifiers are used to the variables that have been narrowed down using the RFE approach: age, sex, on thyroxine, pregnant, TSH, TT4, T4U, T3, and FTI features with rank one. Table 6 and Figure 10 represents the algorithms' performance metrics subsequent to Recursive Feature Elimination (RFE). The table displays Accuracy (Accu %), Precision (Prec %), Recall (Recall %), and F1 score (F1score %) for each model. RNN achieved an accuracy of 96.93%, precision at 75.93%, recall at 96.01%, and an F1 score of 82.10%. Long Short Term Memory Networks exhibited strong performance with an accuracy of 98.24%, precision at 88.89%, recall at 93.26%, and an F1 score of 90.62%. Autoencoder model attained an accuracy of 98.53%, precision at 87.60%, recall at 91.68%, and an F1 score of 89.36%. Multilayer perceptron (MLP) demonstrated an accuracy of 93.45%, precision at 61.48%, recall at

91.93%, and an F1 score of 67.71%. Deep belief network achieved an accuracy of 97.35%, precision at 84.55%, recall at 81.33%, and an F1 score of 81.67. Generative Adversarial Network emerged with an accuracy of 97.98%, precision at 82.45%, recall at 84.83%, and an F1 score of 83.66. These metrics collectively illustrate the comparative performance of each algorithm post-RFE, highlighting RNN's notably high recall, Long Short Term Memory Networks's strong precision and F1 score, and Autoencoder's overall balanced performance across metrics, while MLP showed lower precision but high recall.

6. Conclusion:

In the pursuit of identifying risk factors associated with thyroid disease using deep learning and feature selection approaches, our study delved into a dataset comprising 2,800 occurrences and 28 attributes sourced from the UCI DL Repository. Employing methodologies like Boruta and Recursive Feature Elimination (RFE), we meticulously refined the dataset to distill the most pertinent features. Through rigorous evaluation across six DL algorithms, including Autoencoder and Long Short Term Memory Networks, we achieved notable performance metrics, with accuracy scores surpassing 98% and a remarkable recall rate of 96.01% observed in RNN. Despite these successes, challenges persisted, notably with Multilayer perceptron, which consistently exhibited lower precision levels. Our findings underscore the critical importance of robust preprocessing and feature selection techniques in healthcare analytics, where even marginal improvements in accuracy and recall can significantly impact diagnostic efficacy and patient outcomes. Moving forward, further research endeavors could explore hybrid model architectures and innovative feature engineering strategies to enhance classification accuracy and bolster the identification of thyroid disease risk factors.

Reference

[1] Tabassum, S., Rumky, S. F. F., & Shahariar, M. F. (2022). *Thyroid Disease Analysis and Prediction by Using Machine Learning and Deep Learning: A comparative Approach* (Doctoral dissertation, East West University).

[2] Mohammed, S. J. (2019). *Thyroid disorders prediction using long short term memory (LSTM) technique with non dominated sorting genetic algorithm (NSGA-II) as risk factor feature determination* (Master's thesis, Fen Bilimleri Enstitüsü).

[3] Li, L. N., Ouyang, J. H., Chen, H. L., & Liu, D. Y. (2012). A computer aided diagnosis system for thyroid disease using extreme learning machine. *Journal of medical systems*, *36*, 3327-3337.

[4] Latif, M. A., Mushtaq, Z., Arif, S., Rehman, S., Qureshi, M. F., Samee, N. A., ... & Al-masni, M. A. Improving Thyroid Disorder Diagnosis via Ensemble Stacking and Bidirectional Feature Selection.

[5] Bhatt, V. K., & Pal, V. K. (2019, March). An intelligent system for diagnosing thyroid disease in pregnant ladies through artificial neural network. In *International Conference on Advances in Engineering Science Management & Technology (ICAESMT)-2019, Uttaranchal University, Dehradun, India*.

[6] Kurnaz, S., Mohammed, M. S., & Mohammed, S. J. (2020, June). A high efficiency thyroid disorders prediction system with non-dominated sorting genetic algorithm NSGA-II as a feature

selection algorithm. In *2020 International Conference for Emerging Technology (INCET)* (pp. 1-6). IEEE.

[7] Hosseinzadeh, M., Ahmed, O. H., Ghafour, M. Y., Safara, F., Hama, H. K., Ali, S., ... & Chiang, H. S. (2021). A multiple multilayer perceptron neural network with an adaptive learning algorithm for thyroid disease diagnosis in the internet of medical things. *The Journal of Supercomputing*, *77*, 3616-3637.

[8] Geetha, K., & Baboo, S. S. (2016). An empirical model for thyroid disease classification using evolutionary multivariate Bayseian prediction method. *Global journal of computer science and technology*, *16*(H1), 1-9.

[9] Rasheeduddin, S., & Rao, K. R. (2019). Extreme Learning Machine for Thyroid Nodule Classification with Graph Cluster Ant Colony Optimization Based Feature Selection. *International Journal of Recent Technology and Engineering*, *8*(2), 2277-3878.

[10] Gokilavani, M., Sriram, Vijayaragavan, S. P., & Nirmalrani, V. (2023). Chicken Swarm-Based Feature Selection with Optimal Deep Belief Network for Thyroid Cancer Detection and Classification. In *Computational Intelligence for Clinical Diagnosis* (pp. 21-35). Cham: Springer International Publishing.

[11] Sutradhar, A., Al Rafi, M., Ghosh, P., Shamrat, F. J. M., Moniruzzaman, M., Ahmed, K., ... & Moni, M. A. (2023). An Intelligent Thyroid Diagnosis System Utilising Multiple Ensemble and Explainable Algorithms with Medical Supported Attributes. *IEEE Transactions on Artificial Intelligence*.

[12] SALMAN, K. A. (2021). *The efficiency of classification techniques in predicting thyroid disease* (Doctoral dissertation).

[13] Wang, Y., Yue, W., Li, X., Liu, S., Guo, L., Xu, H., ... & Yang, G. (2020). Comparison study of radiomics and deep learning-based methods for thyroid nodules classification using ultrasound images. *Ieee Access*, *8*, 52010-52017.

[14] Shankar, K., Manickam, P., Devika, G., & Ilayaraja, M. (2018, December). Optimal feature selection for chronic kidney disease classification using deep learning classifier. In *2018 IEEE international conference on computational intelligence and computing research (ICCIC)* (pp. 1-5). IEEE.

[15] Zhang, Q., Zhang, S., Pan, Y., Sun, L., Li, J., Qiao, Y., ... & Li, X. (2022). Deep learning to diagnose Hashimoto's thyroiditis from sonographic images. *Nature Communications*, *13*(1), 3759.

[16] Ma, X., & Zhang, L. (2022). Diagnosis of thyroid nodules based on image enhancement and deep neural networks. *Computational Intelligence and Neuroscience*, *2022*.