Jurnal Informatika Ekonomi Bisnis

http://www.infeb.org

2025 Vol. 7 Iss. 1 Hal: 12-20 e-ISSN: 2714-8491

A Comparative Analysis of Deep Learning Architectures for Obesity Classification Using Structured Data

Gregorius Airalngga[1]✉

[1]Atma Jaya Catholic University of Indonesia

gregorius.airlangga@atmajaya.ac.id

Abstract

Obesity is a significant global health concern, necessitating accurate and efficient diagnostic tools to classify individuals based on obesity levels. This study investigates the performance of five deep learning architectures: Multilayer Perceptron (MLP), Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Bidirectional LSTM (BiLSTM) in classifying obesity levels using structured data. The dataset comprises clinical, demographic, and lifestyle features, and is preprocessed through normalization, label encoding, and Synthetic Minority Oversampling Technique (SMOTE) to address class imbalance. Each model was evaluated using accuracy, precision, recall, and F1-score metrics under stratified 10-fold cross-validation. The results indicate that MLP achieved the highest performance across all metrics, with an accuracy of 99.05%, followed closely by CNN at 98.77%. Sequential models, including LSTM, GRU, and BiLSTM, exhibited comparatively lower performance, achieving accuracies of 83.80%, 86.59%, and 86.78%, respectively. The superior performance of MLP and CNN underscores their suitability for structured datasets with static features, while the sequential models struggled due to the lack of temporal dependencies in the data. This study highlights the importance of aligning model architecture with dataset characteristics for optimal performance. The findings suggest that MLP and CNN are effective choices for obesity classification tasks, providing robust and computationally efficient solutions. Future work could explore hybrid models and incorporate temporal features to enhance the performance of sequential architecture.

Keywords: Obesity Classification, Deep Learning Models, Multilayer Perceptron, Convolutional Neural Networks, Sequential Models in Healthcare.

1. Introduction

Obesity, characterized by excessive accumulation of body fat, has become a leading public health crisis worldwide [1], [2], [3]. Its prevalence has doubled over the past four decades, making it a critical focus of global health policies [4], [5], [6]. The World Health Organization (WHO) estimates that over 650 million adults and 340 million children and adolescents worldwide are classified as obese [7]. Obesity contributes significantly to non-communicable diseases such as cardiovascular disorders, diabetes, and certain cancers, placing immense strain on healthcare systems [8]. Effective prediction and diagnosis are vital to mitigating the adverse health and economic impacts of this condition [9]. However, traditional diagnostic methods, which rely heavily on body mass index (BMI) and subjective clinical assessments, often fail to capture the multifactorial nature of obesity, highlighting the need for data-driven, predictive approaches [10], [11], [12].

Recent advancements in artificial intelligence (AI) have enabled significant progress in healthcare prediction models, including obesity diagnosis [13]. Traditional machine learning (ML) methods, such as decision trees, support vector machines (SVM), and logistic regression, have been widely employed for predictive tasks in healthcare due to their simplicity and interpretability [14]. For instance, a study utilized decision tree models to predict childhood obesity using dietary patterns and physical activity data, achieving moderate accuracy [15]. Similarly, another study implemented SVM to classify obesity in adults based on socio-demographic and clinical features, emphasizing the importance of feature engineering [16]. However, these models often struggle to manage high-dimensional data and fail to capture complex, non-linear interactions between features, limiting their performance in real-world scenarios.

To address these limitations, researchers have increasingly turned to deep learning (DL) architectures, which offer superior performance in modeling complex, high-dimensional data. Convolutional neural networks (CNNs), initially developed for image recognition tasks, have demonstrated their utility in healthcare domains by extracting meaningful feature representations from structured data [17]. For example, a study applied CNNs to analyze obesity-related datasets, achieving substantial performance improvements compared to traditional ML models [18]. Similarly, recurrent neural networks (RNNs) and their variants, such as long short-term memory (LSTM) networks and gated recurrent units (GRUs), have been employed to capture temporal and sequential patterns in health records. Studies by certain researchers highlight the potential of RNN-based models in predicting obesity progression based on time-

Accepted: 27-01-2025 | Revision: 09-02-2025 | Publication: 31-03-2025 | doi: 10.37034/infeb.v7i1.1090

series data, emphasizing the role of recurrent architectures in handling temporal dependencies [19], [20].

While CNNs and RNNs have demonstrated promising results individually, hybrid approaches combining multiple architectures have emerged as state-of-the-art solutions in predictive healthcare. Bidirectional LSTMs (BiLSTMs), for instance, leverage both forward and backward temporal dependencies to enhance predictive accuracy. In obesity prediction, BiLSTMs have been used to analyze sequential dietary and activity data, outperforming unidirectional models [21]. Despite these advancements, limited research exists on systematically comparing the performance of different DL architectures, particularly on datasets involving demographic, behavioral, and physiological attributes. This gap hinders the identification of the most effective DL models for obesity prediction and underscores the need for a comprehensive comparative study.

This research aims to address this gap by evaluating five state-of-the-art DL models: multilayer perceptron (MLP), CNN, LSTM, GRU, and BiLSTM on a dataset comprising clinical and lifestyle features associated with obesity. Unlike previous studies that focus on individual models or lack robust evaluation frameworks, this study employs a rigorous methodology incorporating ten-fold stratified cross-validation. This approach ensures unbiased performance assessment while providing insights into the generalizability of each model. Additionally, the preprocessing pipeline involves standardized techniques such as label encoding and feature scaling to ensure consistency and reproducibility.

The significance of this study lies in its comprehensive evaluation framework and practical implications. The MLP, with its dense, fully connected architecture, serves as a baseline for comparison, while the CNN and RNN-based models explore advanced feature extraction and sequence modeling capabilities. By systematically comparing these architectures, this research provides critical insights into their suitability for obesity prediction tasks. Furthermore, the findings have practical relevance for developing automated diagnostic tools, enabling early interventions and personalized healthcare strategies. The remainder of this article is organized as follows. Section 2 describes the research method including dataset, preprocessing techniques, model architectures, and evaluation metrics. Section 3 provides an in-depth analysis of experimental results and discusses the implications and limitations of the findings. Finally, Section 4 concludes the study by summarizing key contributions and outlining future research directions.

2. Research Method

2.1. Dataset

The dataset used in this study is a rich and diverse collection of clinical, demographic, behavioral, and physiological attributes relevant to obesity prediction [22]. It consists of (N) samples, each described by (d)-dimensional feature vectors ($x_i \in R^d$) and a corresponding categorical target label (y_i), representing the obesity class. The dataset offers a comprehensive view of factors influencing obesity, making it well-suited for evaluating the effectiveness of deep learning models. This section describes the dataset's structure, characteristics, and the specific challenges encountered in its preparation.

2.2. Dataset Overview

The dataset is formally represented as $\mathcal{D} = \{(x_i, y_i) \mid i = 1,2, \dots, N\}$, where each ($x_i$) contains a combination of numerical and categorical features capturing the clinical, demographic, and behavioral aspects of an individual. The target variable (y_i) is a categorical value indicating the individual's obesity classification, which is divided into (K) classes. The features in (x_i) include numerical measurements such as age, BMI, daily caloric intake, and exercise duration, as well as categorical variables such as gender, physical activity level, and dietary habits. Numerical features like age and BMI are critical for capturing physiological trends related to obesity. For example, BMI is a continuous variable defined as the ratio of weight (in kilograms) to the square of height (in meters). This feature plays a pivotal role in obesity classification, as it directly measures body fatness. Other numerical features, such as caloric intake and exercise duration, provide insights into an individual's lifestyle and activity levels, which are strongly correlated with obesity. Categorical features, such as gender and smoking status, further enrich the dataset by adding behavioral and demographic dimensions. The target variable (y_i) is multi-class, categorizing individuals into one of several predefined obesity levels, including underweight, normal weight, overweight, and three classes of obesity (Class I, Class II, and Class III). This classification reflects the severity of obesity and provides an opportunity to investigate the performance of deep learning models in handling multi-class prediction tasks.

2.3. Statistical Characteristics of the Dataset

The dataset exhibits a variety of statistical properties that influence the model design and evaluation. One of the critical aspects is the class distribution, which is imbalanced. Certain obesity categories, such as underweight and Obesity Class III, are underrepresented compared to others. This imbalance in class proportions can lead to biased models that favor majority classes, a challenge that requires mitigation during preprocessing. In terms of feature distribution, numerical variables such

as BMI and caloric intake are often skewed, with extreme values representing outliers in the data. For instance, individuals with extremely high BMIs or caloric intakes are likely to fall into the obesity classes, while those with unusually low BMIs are categorized as underweight. These outliers, though rare, can have a disproportionate impact on the training process by influencing model parameters. Correlations between features also play a crucial role in understanding the dataset's structure. For example, BMI is positively correlated with caloric intake but negatively correlated with exercise duration. These correlations highlight the interdependence among features and underscore the need for models capable of capturing such complex relationships. The Pearson correlation coefficient is used to quantify these relationships can be seen on Equation 1.

$$r_{xy} = \frac{\sum_{i=1}^{N}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{N}(y_i-\bar{y})^2}} \quad (1)$$

Where $(\bar{x})$ and $(\bar{y})$ represent the means of (x) and (y), respectively. Another notable characteristic is the variability in feature scales. Numerical features like BMI and caloric intake have different ranges and units, which can introduce challenges during training, particularly when using gradient-based optimization methods. Models are sensitive to feature scales, and discrepancies can result in slower convergence or suboptimal solutions.

2.4. Challenges in the Dataset

The dataset presents several challenges that necessitate careful preprocessing. One of the primary issues is the presence of missing data, where certain records have incomplete values for numerical or categorical features. Missing data, represented as $(x_{ij} = \text{NaN})$, reduces the effective size of the dataset and can lead to biased models if not handled properly. This study opts for a complete case analysis by removing all records with missing values, ensuring that the cleaned dataset is defined as $\mathcal{D}cen = \{xi, yi \mid xi \text{ is complete}\}$. Class imbalance is another significant challenge, as the dataset contains disproportionately fewer samples for certain obesity categories. This imbalance skews model training, leading to predictions biased toward majority classes. To address this issue, oversampling techniques such as Synthetic Minority Oversampling Technique (SMOTE) are applied. SMOTE generates synthetic examples for minority classes by interpolating between existing samples and their nearest neighbors in feature space which can be seen on Equation 2.

$$x_{synthetic} = x_i + \lambda(x_j - x_i) \quad (2)$$

Where $(\lambda \sim \text{Uniform}(0,1))$, and (x_j) is a nearest neighbor of (x_i). Outliers in numerical features pose additional challenges. Extreme values in BMI or caloric intake, often arising from measurement errors or rare cases, can disproportionately influence model training. Outlier detection and mitigation techniques are considered during preprocessing to ensure that these extreme values do not dominate the learning process. The diversity in feature types: numerical and categorical further complicates preprocessing. Numerical features require normalization to address differences in scale, while categorical features require encoding to convert them into numerical representations suitable for machine learning models. For categorical features, label encoding is applied, transforming each category into a unique integer. For example, a feature (x_j) with categories $(\{c_1, c_2, \ldots, c_m\})$ is mapped to integers $(\{0, 1, \ldots, m-1\})$ using the function $(\phi_j: C_j \rightarrow \{0,1, \ldots, m-1\})$, where $(x_{ij} = \phi_j(x_{ij}))$.

2.5. Dataset Preparation

The dataset preparation process involves multiple steps to ensure that it is ready for model training and evaluation. First, missing values are addressed by removing incomplete records, resulting in a clean dataset. Next, categorical features are encoded using label encoding, while the target variable is transformed into a one-hot encoded format to facilitate multi-class classification. Numerical features are normalized using z-score normalization, ensuring that all features have zero mean and unit variance. The normalized value of a numerical feature (x_j) is given by Equation 3.

$$x_j^{\text{norm}} = \frac{x_j - \mu_j}{\sigma_j} \quad (3)$$

Where (μ_j) and (σ_j) are the mean and standard deviation of (x_j), respectively. Finally, the dataset is split into training and testing subsets using stratified sampling to preserve the class distribution. For models requiring sequential input, such as LSTM or CNN, the feature matrix (X) is reshaped to include a temporal dimension, resulting in $(X_{seq} \in R^{N \times d \times 1})$. This ensures compatibility with sequence-based architecture.

2.6. Preprocessing Techniques

Preprocessing is a crucial step in machine learning workflows, especially when working with datasets that include a mix of numerical and categorical features, missing values, and imbalanced class distributions. Proper preprocessing ensures that the data is in a suitable format for model training and evaluation while addressing challenges such as inconsistent scales, missing data, and outliers. In this study, a systematic and rigorous preprocessing pipeline was applied to prepare the dataset for deep learning models. This subsection describes each preprocessing stage in detail, emphasizing the mathematical foundations and rationale behind each technique.

2.6.1. Handling Missing Values

Missing data is a common issue in real-world datasets and can occur for various reasons, such as data entry

errors, sensor malfunctions, or non-responses in surveys. In this dataset, missing values were observed in both numerical and categorical features. To address this, the complete-case analysis approach was employed, where records with missing values were removed entirely. Let (x_{ij}) denote the value of the (j)-th feature for the (i)-th sample. If $(x_{ij} = \text{NaN})$ for any (j), the entire sample $((x_i, y_i))$ was excluded from the dataset. The cleaned dataset is mathematically expressed on Equation 4.

$$\mathcal{D}_{clean} = \{(x_i, y_i) \in \mathcal{D} \mid \forall j, x_{ij} \neq \text{NaN}\} \quad (4)$$

This approach ensures that no incomplete records are present during model training, thereby avoiding potential biases caused by missing data. Although this method reduces the overall size of the dataset, it guarantees that all samples used in the analysis are complete and consistent.

2.6.2. Encoding Categorical Variables

Categorical features are inherently non-numerical, and their direct use in machine learning models is not feasible. To make these features suitable for training, label encoding was applied to convert categorical values into numerical representations. Let (x_j) represent a categorical feature with (m) unique categories, denoted as $(C_j = \{c_1, c_2, \dots, c_m\})$. A mapping function $(\phi_j: C_j \rightarrow \{0,1, \dots, m-1\})$ was defined, where each category (c_k) is assigned to a unique integer. The transformation for each sample (x_{ij}) is given by $x_{ij} = \phi_j(x_{ij})$, where $(x_{ij} \in C_j)$ is the original categorical value, and $(\phi_j(x_{ij}))$ is the corresponding integer-encoded value. This method preserves the ordinal relationships in categorical features where applicable.

For the target variable (y_i), a more advanced transformation, one-hot encoding, was used to represent the multi-class labels as binary vectors. For a dataset with (K) classes, the one-hot encoded representation of (y_i) is defined as $y_i = [y_{i1}, y_{i2}, \dots, y_{iK}]^T$, where $(y_{ik} = 1 \text{ if } y_i = k;\ y_{ik} = 0 \text{ otherwise})$. This representation ensures that the target variable is compatible with categorical cross-entropy loss functions used in multi-class classification tasks.

2.6.3. Normalization of Numerical Features

Numerical features in the dataset, such as age, BMI, and caloric intake, exhibit varying scales and ranges. For example, BMI values typically range from 15 to 40, while caloric intake can range from hundreds to thousands of kilocalories. Such discrepancies in scale can adversely affect gradient-based optimization during model training. To address this issue, z-score normalization was applied to standardize all numerical features. For a given numerical feature (x_j), its normalized value (x_j^{norm}) is computed as $x_j^{\text{norm}} = \frac{x_j - \mu_j}{\sigma_j}$, where (μ_j) and (σ_j) are the mean and standard deviation of (x_j) across the dataset, respectively on Equation 5 and Equation 6.

$$\mu_j = \frac{1}{N}\sum_{i=1}^{N} x_{ij} \quad (5)$$

$$\sigma_j = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_{ij} - \mu_j)^2} \quad (6)$$

This transformation ensures that all numerical features have zero mean and unit variance, effectively mitigating the impact of feature scale discrepancies. Moreover, normalization helps prevent features with large ranges from dominating the learning process, leading to more stable and efficient optimization.

2.6.4. Balancing Imbalanced Classes

Class imbalance is a prevalent issue in many real-world datasets, including this one, where certain obesity categories, such as underweight and Obesity Class III, are significantly underrepresented. Without addressing this imbalance, models tend to favor majority classes, resulting in poor performance on minority classes. To mitigate this issue, the Synthetic Minority Oversampling Technique (SMOTE) was applied to generate synthetic examples for minority classes. SMOTE works by interpolating existing samples and their nearest neighbors in the feature space. For a minority class sample (x_i), a synthetic sample $(\text{x}_{\text{synthetic}})$ is generated as Equation 7.

$$\text{x}_{\text{synthetic}} = \text{x}_\text{i} + \lambda(\text{x}_\text{j} - \text{x}_\text{i}) \quad (7)$$

Where $(\lambda \sim \text{Uniform}(0,1))$ is a random scalar, and (x_j) is a randomly chosen nearest neighbor of (x_i). By introducing synthetic samples, SMOTE effectively balances the class distribution and improves the model's ability to generalize to minority classes.

2.6.5. Train-Test Splitting and Reshaping

To evaluate the performance of the deep learning models, the dataset was divided into training and testing subsets using stratified sampling. Stratified sampling ensures that the class distribution in both subsets mirrors the overall class distribution in the dataset. Let $(\mathcal{D}_{train})$ and $(\mathcal{D}_{test})$ represent the training and testing subsets, respectively, such that $\mathcal{D}_{train} \cup \mathcal{D}_{test} = \mathcal{D}$, $\mathcal{D}_{train} \cap \mathcal{D}_{test} = \{\}$. For sequence-based models, such as LSTM and CNN, the feature matrix (X) was reshaped to include a temporal dimension. For a dataset with (N) samples and (d) features, the reshaped input is represented as $X_{seq} \in R^{N \times d \times \mathbb{1}}$. This reshaping ensures compatibility with convolutional and recurrent layers, which process data with temporal or spatial structures. By following this preprocessing pipeline, the dataset was transformed into a format that is robust, consistent, and well-suited for training deep learning models. Each step addressed specific challenges in the data, ensuring that the resulting models are not only accurate but also generally unseen data.

2.7. Model Architectures

The models used in this study are designed to capture the complex relationships between the features in the dataset and the target obesity classification labels. A variety of deep learning architectures were explored, each tailored to exploit specific patterns in the data, such as non-linear interactions, temporal dependencies, and hierarchical feature representations. This subsection provides an in-depth explanation of the architecture, including their mathematical formulations and the rationale behind their design.

The MLP is a fully connected feedforward neural network that serves as a baseline model in this study. It is designed to capture non-linear relationships between input features and the target labels. An MLP consists of an input layer, one or more hidden layers, and an output layer. Each layer applies a weighted transformation to the input, followed by a non-linear activation function. Let $(x \in R^d)$ represent the input feature vector with (d) features. The forward propagation in the MLP is defined as Equation 8.

$$(h^{(l)} = f^{(l)}\big(W^{(l)}h^{(l-1)} + b^{(l)}\big),) \quad (8)$$

Where ($l = 1, 2, \dots, L$). In this equation, $(W^{(l)})$ is the weight matrix for layer (l), $(b^{(l)})$ is the bias vector, $(h^{(l-1)})$ is the output from the previous layer (with $(h^{(0)} = x)$), and $(f^{(l)})$ is the activation function. The ReLU activation function is used in the hidden layers, defined as $(f^{(l)}(z) = \max(0, z))$. The output layer uses the softmax activation function, which converts the logits into probabilities, defined as Equation 9.

$$\widehat{y_k} = \frac{\exp(z_k)}{\sum_{j=1}^{K} \exp(z_j)} \quad (9)$$

Where (z_k) is the logit for class (k) in the output layer, and (K) is the total number of classes. Dropout regularization is applied after each hidden layer to reduce overfitting, where dropout randomly sets a fraction (p) of the neurons in a layer to zero during training, expressed as $\mathrm{h}_{(\ )} = \mathrm{m} \odot \mathrm{h}^{(l)}, \mathrm{with}(\mathrm{m} \sim \mathrm{Bernoulli}(1 - \mathrm{p}))$.

CNNs are designed to extract hierarchical features from structured data. While CNNs are commonly used in image processing, they are also effective for structured datasets when reshaped into a format suitable for convolution operations. The input (x) is reshaped into $(X \in R^{d \times 1})$, where (d) is the number of features. The key operation in CNNs is the convolution, which applies a set of learnable filters (W) to the input. For a one-dimensional input, the convolution operation at position (t) is defined as Equation 10.

$$h_t = f\big(\sum_{i=1}^{k} W_i x_{t+i-1} + b\big) \quad (10)$$

Where (k) is the filter size, (W_i) are the filter weights, (b) is the bias term, and (f) is the activation function (ReLU in this study). Pooling layers are applied after convolution to reduce the dimensionality and retain the most significant features, where max pooling is used, defined as Equation 11.

$$h_{pooled} = \max_{i=1,\dots,k} h_i \quad (11)$$

The flattened output from the final pooling layer is passed through fully connected layers and a softmax output layer, like the MLP.

LSTM networks are a type of recurrent neural network (RNN) designed to model sequential dependencies in data. Unlike standard RNNs, LSTMs can effectively learn long-term dependencies by using memory cells and gates. For a sequence of inputs $(\{x_t\}_{t=1}^{T})$, the LSTM computes the hidden state (h_t) and cell state (c_t) at each time step (t). The updates are defined as Equation 12 to Equation 16.

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (12)$$

$$f_t = \sigma\big(W_f x_t + U_f h_{t-1} + b_f\big) \quad (13)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (14)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (15)$$

$$h_t = o_t \odot \tanh(c_t) \quad (16)$$

Where (i_t), (f_t), and (o_t) are the input, forget, and output gates, respectively, (c_t) is the cell state, (σ) is the sigmoid activation function, and ($\odot$) represents element-wise multiplication. The final hidden state (h_T) is passed through fully connected layers and a softmax output layer.

The GRU is a simplified variant of the LSTM that combines the input and forget gates into an update gate. The updates for GRUs are defined as Equation 17, 18 and 19.

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \quad (17)$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \quad (18)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tanh(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h) \quad (19)$$

BiLSTMs extend LSTMs by processing the input sequence in both forward and backward directions. The final hidden state is obtained by concatenating the hidden states from both directions, represented as Equation 20.

$$h_t^{BiLSTM} = h_t^{forward}, h_t^{backward} \quad (20)$$

By employing these architectures, this study explores their ability to capture diverse patterns in the data, ranging from simple non-linear relationships to complex temporal dependencies.

2.7. Evaluation Metrics

The evaluation of machine learning models is a critical aspect of this study, as it provides quantitative insights into their performance in classifying obesity levels based on the dataset. To ensure a comprehensive assessment, this study utilizes several widely accepted metrics, including accuracy, precision, recall, and F1-score. These metrics collectively evaluate the models' ability to correctly classify samples, balance between positive and negative predictions, and handle imbalanced datasets effectively. This subsection provides a detailed explanation of each metric, including its mathematical formulation, interpretation, and relevance to the problem at hand.

The accuracy metric is one of the most intuitive measures of model performance. It represents the proportion of correctly classified samples out of the total samples. Mathematically, accuracy is defined as Equation 21.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{21}$$

where TP, TN, FP, and FN represent true positives, true negatives, false positives, and false negatives, respectively. True positives are samples that belong to a particular class and are correctly predicted as such, while true negatives are samples that do not belong to the class and are correctly predicted as such. False positives and false negatives, on the other hand, represent misclassified samples. Although accuracy is straightforward and useful, it can be misleading when the dataset is imbalanced. In such cases, the model may achieve high accuracy by simply predicting the majority class, while ignoring the minority classes. To address the limitations of accuracy in imbalanced datasets, precision is used to measure the proportion of true positive predictions out of all samples predicted as positive. Precision is particularly important when the cost of false positives is high. For a specific class \(k \), precision is defined as Equation 22.

$$\text{Precision}_k = \frac{\text{TP}_k}{\text{TP}_k + \text{FP}_k} \tag{22}$$

where (TP_k) and (FP_k) are the true positives and false positives for class (k), respectively. High precision indicates that the model has a low false positive rate, making it reliable in identifying the positive class without including incorrect predictions. The recall, also known as sensitivity or true positive rate, measures the proportion of actual positive samples that are correctly identified by the model. Recall is particularly crucial when the cost of false negatives is high, as in medical diagnoses where missing a positive case can have severe consequences. For a specific class (k), recall is defined as Equation 23.

$$\text{Recall}_k = \frac{\text{TP}_k}{\text{TP}_k + \text{FN}_k} \tag{23}$$

where (TP_k) and (FN_k) are the true positives and false negatives for class (k), respectively. High recall indicates that the model is effective in capturing most of the actual positive samples, even if it includes some false positives. To achieve a balance between precision and recall, the F1-score is used as a harmonic mean of the two metrics. The F1-score for a specific class (k) is defined as Equation 24.

$$\text{F1-Score}_k = 2 \cdot \frac{\text{Precision}_k \cdot \text{Recall}_k}{\text{Precision}_k + \text{Recall}_k} \tag{24}$$

The F1-score ranges between 0 and 1, with higher values indicating better performance. It is particularly useful in imbalanced datasets, as it considers both false positives and false negatives in a single metric. For multi-class classification problems, as in this study, these metrics are extended by averaging across all classes. Two common averaging strategies are macro-averaging and weighted-averaging. Macro-averaging calculates the metric for each class independently and then takes the unweighted mean as defined in Equation 25.

$$\text{Macro Average} = \frac{1}{K}\sum_{k=1}^{K} \text{Metric}_k \tag{25}$$

where (K) is the total number of classes. This approach treats all classes equally, regardless of their size. Weighted-averaging, on the other hand, accounts for class imbalance by weighting each class's metric by its proportion of samples in the dataset as defined in Equation 26.

$$\text{Weighted Average} = \frac{\sum_{k=1}^{K} N_k \cdot \text{Metric}_k}{\sum_{k=1}^{K} N_k} \tag{26}$$

Where (N_k) is the number of samples in class (k). The weighted average provides a more realistic evaluation for imbalanced datasets, as it reflects the performance across all classes while considering their distribution. To ensure robust evaluation, the study employs stratified (k)-fold cross-validation with $(k = 10)$. This technique divides the dataset into (k) equally sized folds while preserving the class distribution in each fold. During cross-validation, the model is trained on $(k - 1)$ folds and tested on the remaining fold, rotating the test fold in each iteration. Let $(\mathcal{D})$ represent the dataset and $(\mathcal{D}_i)$ denote the test fold in the (i)-th iteration. The

average metric across all folds is calculated as Equation 27.

$$\text{Metric}_{\text{CV}} = \frac{1}{k}\sum_{i=1}^{k} \text{Metric}(\mathcal{D}_i) \quad (27)$$

This process reduces the risk of overfitting and ensures that the evaluation metrics generalize well to unseen data.

3. Results and Discussion

The results of the experiments conducted on the dataset using five deep learning models: MLP, CNN, LSTM, GRU, and BiLSTM are presented in terms of accuracy, precision, recall, and F1-score as presented in the Table 1. These metrics provide a holistic evaluation of each model's performance under a stratified 10-fold cross-validation setting. This section delves into the quantitative outcomes, followed by a detailed discussion of the comparative performance of the models.

Table 1. Deep Learning Performance

Model	Accuracy	Precision	Recall	F1-Score
MLP	0.9905	0.9907	0.9905	0.9905
CNN	0.9877	0.9879	0.9877	0.9877
LSTM	0.8380	0.8407	0.8380	0.8374
GRU	0.8659	0.8686	0.8659	0.8653
BiLSTM	0.8678	0.8705	0.8678	0.8673

3.1. Quantitative Results

The MLP model achieved the highest performance across all metrics, with an accuracy of (0.9905), precision of (0.9907), recall of (0.9905), and F1-score of (0.9905). These results indicate that the MLP model is capable of learning non-linear relationships in the dataset effectively and generalizes well across all obesity classes. The CNN model, while slightly less accurate than the MLP, also performed exceptionally well with an accuracy of (0.9877), precision of (0.9879), recall of (0.9877), and F1-score of (0.9877). The hierarchical feature extraction capability of CNNs likely contributed to its high performance, although its results were marginally inferior to those of the MLP.

On the other hand, the sequential models LSTM, GRU, and BiLSTM demonstrated comparatively lower performance. The LSTM model achieved an accuracy of (0.8380), precision of (0.8407), recall of (0.8380), and F1-score of (0.8374). GRU performed slightly better than LSTM, with an accuracy of (0.8659), precision of (0.8686), recall of (0.8659), and F1-score of (0.8653). The BiLSTM model achieved the highest performance among the sequential models, with an accuracy of (0.8678), precision of (0.8705), recall of (0.8678), and F1-score of (0.8673).

3.2. Discussion

The results highlight a clear distinction in performance between the fully connected and convolutional models on the one hand, and the sequential models on the other. The superior performance of MLP and CNN can be attributed to the inherent structure of the dataset, which likely contains more static and non-temporal features. This characteristic makes it less suitable for models designed to capture temporal dependencies, such as LSTM, GRU, and BiLSTM. The MLP model's performance demonstrates its effectiveness in handling structured datasets where features are independent and non-sequential. The fully connected architecture of the MLP allows it to learn complex non-linear mappings between the input features and the target labels. The addition of dropout regularization further enhances its ability to generalize by reducing overfitting. This explains its slight edge over CNN, which relies on convolutional filters to capture local patterns. While CNNs excel in processing spatial or sequential data, their performance in this study indicates that the dataset does not benefit significantly from such hierarchical feature extraction.

The CNN model's strong performance can be attributed to its ability to process the dataset as a sequence of features after reshaping. By applying convolutional filters, the CNN model captures local feature patterns, which likely play a secondary role in distinguishing between obesity levels. However, the lack of a significant spatial structure in the dataset may explain why CNN underperformed slightly compared to MLP. The sequential models, including LSTM, GRU, and BiLSTM, demonstrated lower performance compared to MLP and CNN. This is likely because the dataset lacks a strong temporal or sequential structure, which these models are specifically designed to exploit. The relatively modest performance of LSTM can be attributed to its reliance on capturing long-term dependencies in sequences, which may not be relevant in this context. GRU, being a simplified variant of LSTM, performed slightly better due to its reduced computational complexity, which allows for faster convergence and fewer overfitting issues.

BiLSTM marginally outperformed both LSTM and GRU, suggesting that processing the input data in both forward and backward directions contribute to improved learning. However, the overall performance of BiLSTM still falls short compared to MLP and CNN, reinforcing the notion that the dataset's characteristics are better suited for models that do not rely on sequential dependencies. Another notable factor influencing the performance of sequential models is the inherent class imbalance in the dataset. While techniques such as SMOTE were applied to mitigate this issue, sequential models may still struggle with accurately learning minority class patterns due to their reliance on maintaining dependencies over a sequence. This

limitation becomes more apparent when evaluating metrics such as recall, where capturing minority class instances is critical.

The differences in performance metrics among the models highlight their varying capabilities in capturing distinct data patterns. For instance, the F1-score, which balances precision and recall, is consistently lower for sequential models compared to MLP and CNN. This suggests that the sequential models are less effective in simultaneously minimizing both false positives and false negatives. On the other hand, the precision and recall metrics for MLP and CNN are nearly identical, indicating their balanced capability to correctly identify positive instances without overpredicting.

3.3. Implications and Recommendations

The findings of this study have implications for the selection of machine learning models for structured datasets, particularly in the context of obesity classification. The superior performance of MLP and CNN suggests that simpler, non-sequential architectures may suffice for datasets lacking temporal dependencies. These models are computationally efficient and require less hyperparameter tuning compared to sequential models, making them ideal for practical applications.

However, sequential models should not be dismissed entirely. In scenarios where datasets exhibit temporal or sequential patterns, models such as LSTM, GRU, and BiLSTM may outperform MLP and CNN. Future research could explore incorporating temporal features or designing hybrid architecture that combines the strengths of sequential and non-sequential models to achieve improved performance. Moreover, the results underscore the importance of evaluating multiple metrics to gain a comprehensive understanding of model performance. While accuracy provides a general overview, precision, recall, and F1-score offer deeper insight into the models' ability to handle imbalanced datasets and capture patterns across all classes.

4. Conclusion

This study evaluates five deep learning architectures: MLP, CNN, LSTM, GRU, and BiLSTM for obesity classification using a structured dataset. A stratified 10-fold cross-validation framework assesses accuracy, precision, recall, and F1-score. MLP outperforms all models with the highest accuracy (0.9905), precision (0.9907), recall (0.9905), and F1-score (0.9905), excelling in capturing complex non-linear relationships in independent, non-sequential features. CNN follows closely, benefiting from convolutional filters despite the dataset's limited spatial structure. Conversely, sequential models perform worse, with LSTM (0.8380), GRU (0.8659), and BiLSTM (0.8678) showing lower accuracy due to the absence of temporal dependencies. Though BiLSTM marginally surpasses LSTM and GRU, its performance remains inferior to MLP and CNN. Class imbalance posed challenges, with SMOTE improving results, but sequential models struggled to capture minority class patterns, reflected in their lower recall and F1-scores. This study underscores that MLP and CNN are optimal for structured datasets with static features, offering superior performance with lower computational demands. While sequential models are less effective here, they may still be valuable for datasets with temporal dependencies. Future research could explore hybrid models that integrate sequential and non-sequential architectures.

References

[1] Ryan, D., Barquera, S., Barata Cavalcanti, O., & Ralston, J. (2021). The global pandemic of overweight and obesity: Addressing a twenty-first century multifactorial disease. In *Handbook of Global Health* (pp. 739–773). Springer. https://doi.org/10.1007/978-3-030-45009-0_39

[2] Ahmed, B., & Konje, J. C. (2023). The epidemiology of obesity in reproduction. *Best Practice & Research Clinical Obstetrics & Gynaecology, 89*, 102342. https://doi.org/10.1016/j.bpobgyn.2023.102342

[3] Shafiee, A., Nakhaee, Z., Bahri, R. A., Amini, M. J., Salehi, A., Jafarabady, K., Seighali, N., Rashidian, P., Fathi, H., & Esmaeilpur Abianeh, F. (2024). Global prevalence of obesity and overweight among medical students: *A systematic review and meta-analysis. BMC Public Health, 24*(1), 1673. https://doi.org/10.1186/s12889-024-19184-4

[4] Koliaki, C., Dalamaga, M., & Liatis, S. (2023). Update on the obesity epidemic: After the sudden rise, is the upward trajectory beginning to flatten? *Current Obesity Reports, 12*(4), 514–527. https://doi.org/10.1007/s13679-023-00527-y

[5] Boutari, C., & Mantzoros, C. S. (2022). A 2022 update on the epidemiology of obesity and a call to action: As its twin COVID-19 pandemic appears to be receding, the obesity and dysmetabolism pandemic continues to rage on. *Metabolism, 133*, 155217. https://doi.org/10.1016/j.metabol.2022.155217

[6] Popkin, B. M., & Ng, S. W. (2022). The nutrition transition to a stage of high obesity and noncommunicable disease prevalence dominated by ultra-processed foods is not inevitable. *Obesity Reviews, 23*(1), e13366. https://doi.org/10.1111/obr.13366

[7] Shmyhol, I., Hrytsai, N., & Onipko, V. (2021). Problems of overweight and obesity among students at general secondary educational institutions. *Journal of Physical Education and Sport, 21*, 2901–2907. https://doi.org/10.7752/jpes.2021.s5385

[8] Goswami, N. (2024). A dual burden dilemma: Navigating the global impact of communicable and non-communicable diseases and the way forward. *International Journal of Medical Research, 12*(3), 65–77.

[9] Abdallah, S., Sharifa, M., Almadhoun, M. K. I., Khawar Sr, M. M., Shaikh, U., Balabel, K. M., Saleh, I., Manzoor, A., Mandal, A. K., & Ekomwereren, O. (2023). The impact of artificial intelligence on optimizing diagnosis and treatment plans for rare genetic disorders. *Cureus, 15*(10), e46860. https://doi.org/10.7759/cureus.46860

[10] Foppiani, A. (2021). *Machine learning applied to clinical nutrition: Clinical decision support and new patterns recognition* (Master's thesis, Università degli Studi di Milano).

[11] Rautiainen, I. (2024). *Prediction methods for assessing the development of individual health status* (JYU Dissertations. University of Jyväskylä).

[12] Coman, L.-I., Ianculescu, M., Paraschiv, E.-A., Alexandru, A., & Bădărău, I.-A. (2024). Smart solutions for diet-related disease management: Connected care, remote health monitoring

systems, and integrated insights for advanced evaluation. A*pplied Sciences*, *14*(6), 2351. https://doi.org/10.3390/app14062351

[13] An, R., Shen, J., & Xiao, Y. (2022). Applications of artificial intelligence to obesity research: Scoping review of methodologies. J*ournal of Medical Internet Research*, *24*(12), e40589. https://doi.org/10.2196/40589

[14] Alanazi, A. (2022). Using machine learning for healthcare challenges and opportunities. *Informatics in Medicine Unlocked*, *30*, 100924. https://doi.org/10.1016/j.imu.2022.100924

[15] Iparraguirre-Villanueva, O., Mirano-Portilla, L., Gamarra-Mendoza, M., & Robles-Espiritu, W. (2023). Predicting obesity in nutritional patients using decision tree modeling. *Proceedings of the International Conference on Data Science and Information Technology*, 1–6.

[16] Lee, S., & Chun, J. (2024). Identification of important features in overweight and obesity among Korean adolescents using machine learning. *Children and Youth Services Review*, *161*, 107644. https://doi.org/10.1016/j.childyouth.2024.107644

[17] Li, M., Jiang, Y., Zhang, Y., & Zhu, H. (2023). Medical image analysis using deep learning algorithms. *Frontiers in Public Health*, *11*, 1273253. https://doi.org/10.3389/fpubh.2023.1273253

[18] Safaei, M., Sundararajan, E. A., Driss, M., Boulila, W., & Shapi'i, A. (2021). A systematic literature review on obesity: Understanding the causes & consequences of obesity and reviewing various machine learning approaches used to predict obesity. *Computers in Biology and Medicine*, *136*, 104754.

[19] Moen, H., Raj, V., Vabalas, A., Perola, M., Kaski, S., Ganna, A., & Marttinen, P. (2024). Towards modeling evolving longitudinal health trajectories with a transformer-based deep learning model. *arXiv preprint* arXiv:2412.08873.

[20] Shen, H. (2023). Enhancing diagnosis prediction in healthcare with knowledge-based recurrent neural networks. *IEEE Access*.

[21] Liu, J., Chen, P., Song, H., Zhang, P., Wang, M., Sun, Z., & Guan, X. (2023). Prediction of cholecystokinin-secretory peptides using bidirectional long short-term memory model based on transfer learning and hierarchical attention network mechanism. *Biomolecules*, *13*(9), 1372. https://doi.org/10.3390/biom13091372

[22] Palechor, F. M., & De la Hoz Manotas, A. (2019). Dataset for estimation of obesity levels based on eating habits and physical condition in individuals from Colombia, Peru and Mexico. *Data in Brief*, *25*, 104344. https://doi.org/10.1016/j.dib.2019.104344