engineering proceedings

MDPI

*Proceeding Paper*

# Classification of Stress Levels Using Convolutional Neural Networks and a Wrist-Worn Wearable Device †

**Mauro Rosas-Lara, Julio C. Mendoza-Tello * and Fausto A. Torres-Montalván**

Facultad de Ingeniería y Ciencias Aplicadas, Universidad Central del Ecuador, Quito 170129, Ecuador; mrosas@uce.edu.ec (M.R.-L.); faustotm711@outlook.com (F.A.T.-M.)

* Correspondence: jcmendoza@uce.edu.ec

† Presented at the Sixth International Conference on Information Systems and Software Technologies (ICI2ST 2026), Quito, Ecuador, 23–25 September 2026.

**Abstract**

Stress is a common factor in people's daily lives. This condition affects health and, consequently, personal, emotional, and work environment. Although stress affects individuals in different ways and at varying levels of intensity, it can lead to a wide range of diseases involving different parts of the human body. Stress is caused by pressures related to past events and concerns about the future, producing symptoms such as anxiety and emotional instability, panic attacks, headaches, insomnia, cardiovascular problems, among others. Monitoring physiological parameters makes it possible to identify risk factors early and thus prevent diseases. In this context, a wrist-worn wearable device captures physiological information through electronic sensors that measure body temperature, heart rate and electrodermal activity. These parameters are correlated to classify stress levels. In recent years, convolutional neural networks have demonstrated the ability to predict and classify several medical conditions using trained models and datasets. Therefore, the aim of this paper is to develop a model based on convolutional neural networks to support the detection of stress levels using wrist-worn wearable device. For this, the research was conducted in two phases: data mining and web development. Regarding data mining, the CRISP-DM methodology was used to define a model through five activities: business understanding, data understanding, data preparation, modeling, and evaluation. The findings indicate that performance metrics derived from the confusion matrix exceeded 90%, including an overall accuracy of 95.47%. Then, the model is implemented within an app using the waterfall model. Conclusions and future work are detailed at the end of the paper.

**Keywords:** stress; wearable; convolutional neural network; confusion matrix; health

Check for updates

Academic Editor: Leonardo Valdivieso

Published: 18 September 2026

**Copyright:** © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

## 1. Introduction

Stress is a state of worry or emotional tension caused by a difficult or demanding situation. Nowadays, stress has become a common condition in people's daily lives. It is a growing and significant problem that causes physiological and emotional disorders, negatively impacting human health [1]. Depression and anxiety represent the most common health problems. It has been proven that uncontrolled stress increases the risk of cardiovascular diseases, with significant consequences for the nervous system. Therefore, it is necessary to monitor stress levels to prevent health problems [2].

Currently, wearable devices (such as fitness watches and activity bracelets) facilitate the real-time collection of physiological data through heart rate, electrodermal activity

and skin temperature [3]. In this context, one of the most promising methods in the data analytics field is deep learning.

The integration of deep learning algorithms (as data processing techniques) and wearable devices (as data collection instruments) has the potential to contribute to the identification of stress-related patterns [4]. From this perspective, the aim of this research is to develop a model based on convolutional neural networks to support the detection of stress levels using wrist-worn wearable device.

The sections of the paper are as follows. Section 2 describes the methodology and materials used to conduct the experiment. Section 3 details the tasks performed using CRISP-DM. Then, it is explained how the model was implemented within an app. Finally, Section 4 outlines the conclusions of our research.

## 2. Methods and Materials

This section covers the methodology, concepts and materials used in the research.

### *2.1. Methodology*

In this research, four phases were carried out, namely theoretical background, data mining, web development, and conclusions. Figure 1 shows a summary of the activities conducted during these phases.

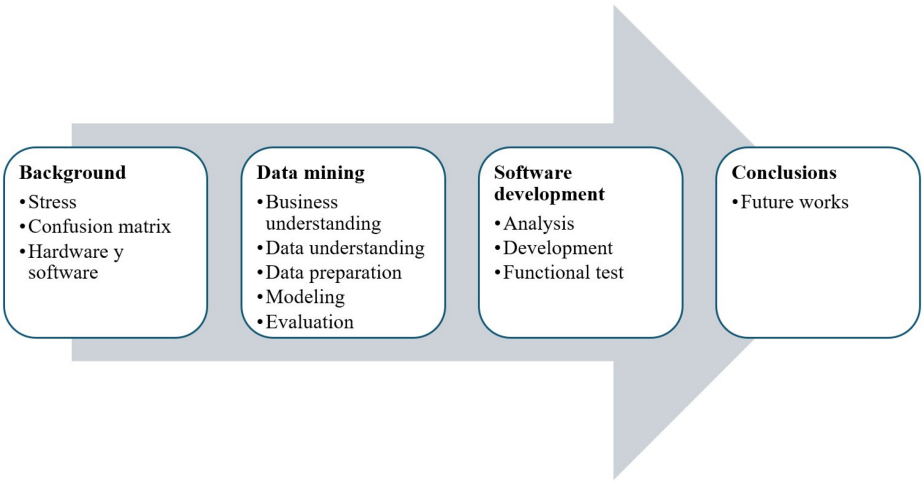



**Figure 1.** Research methodology.

First, theoretical background. A brief description of stress and its effects on human health was provided to address our case study. Along these lines, the physiological parameters used in this research to determine stress levels using wearable devices are explained. A definition of the convolutional neural network and its architecture are also provided to develop a prediction model. The metrics (based on the confusion matrix) used to evaluate the model and interpret the results are described. Additionally, the software and hardware used for these phases are described.

Second, data mining. In this phase, CRISP-DM (Cross Industry Standard Process for Data Mining) was used as the methodology. To achieve this, five phases are addressed: business understanding, data compression, data understanding, modeling, and model evaluation.

Third, web development. A mobile app based on the convolutional model was designed using the waterfall model. To achieve this, three activities were conducted: requirements analysis, development, and functional testing.

Fourth, conclusions. conclusions and future work are presented at the end of the paper.

### *2.2. Theoretical Background*

In this section, the theoretical concepts that support the research are explained.

#### 2.2.1. Stress

Stress is an emotional, physical, or chemical factor that causes physical and emotional tension, which can lead to illness. In turn, stress is considered a biological response that alters the behavior of the human body in the face of adversity. Although stress affects people in different ways, its effects can be summarized into three categories.

- Physiological effects, such as cardiovascular, immunological, digestive, hormonal, musculoskeletal, metabolic, muscular and respiratory problems.
- Mental effects, such as anxiety, depression, inability to make decisions, guilt, lack of motivation, irritability, mental fatigue, insecurity, sleep disorders, catastrophic thinking, low self-esteem, and difficulty concentrating.
- Social effects, such as social isolation, communication problems, impact on interpersonal relationships, difficulties in the workplace, and social inequality [2].

This research uses three physiological parameters to evaluate stress, namely (a) temperature, (b) heart rate (HR), and (c) electrodermal activity. A normal body temperature can range between 36 and 37 degrees Celsius.

Regarding heart rate (HR), it is a noninvasive parameter used to analyze and assess cardiac activity. This parameter can be obtained from an electrocardiogram. Wearable devices (such as the Apple Watch) are capable of recording an electrocardiogram in 30 s. Heart rate is measured through the number of heart beats per minute (BPM) and varies according to age [5].

Regarding electrodermal activity (EDA), it refers to changes in the skin's electrical properties during periods of sweating, emotional excitement, and stress. This is because the nervous system activates sweat glands, which influence the regulation of body temperature. Thus, electrodermal activity is a physiological indicator of the emotional state of a person [6].

#### 2.2.2. Convolutional Neural Networks

Neurons are connected together for data processing and detection of specific patterns. From this perspective, a convolutional neural network (CNN) is composed of three layers: convolution, pooling, and fully connected [7].

First, the convolution layer. A convolution is a mathematical operator that transforms two functions—f(x) and g(x)—into a third function, which indicates to what extent f(x) and $g^{-1}$ (x) overlap to capture the relationships between neighboring data values. In this layer, parameters such as filters, weights, and activation functions are defined. The filters are used to explore the characteristics of the input data.

Second, the pooling layer. Pooling is known as subsampling because it reduces dimensions by decreasing the number of input parameters. The clustering operation sweeps the entire input with a filter, but the difference is that the filter has no weight. Instead, the kernel applies an aggregation function to the values within the receptive field and fills the output matrix. Average pooling performs downsampling, dividing the input into rectangular regions and obtaining the average of each region. Max pooling applies a square matrix to the convolution results and preserves the maximum value of each region. Third, the fully connected layer. It is the final layer of the network. It combines and processes features extracted from the previous layers before generating the final output [8].

https://doi.org/10.3390/engproc2026153042

2.2.3. Confusion Matrix

In machine learning, the confusion matrix is a tool to evaluate the performance of a classification model. It is a square matrix that compares actual instances with predicted instances of a class [9]. That is, the matrix compares predicted values against actual values of a data set. In other words, the matrix supports the identification of correct and incorrect predictions in classification task [10].

The matrix includes four types of values, namely: true positive (TP), true negative (TN), false positive (FP), and false negative (FN). In this context, TP is the number of positive predictions that are actually positive, TN is the number of negative predictions that are actually negative, FP is the number of positive predictions that are actually negative, and FN is the number of negative predictions that are actually positive. The matrix is useful to calculate other metrics, such as accuracy, precision, recall, specificity, and F1-score [11].

- Accuracy. It is the ratio between correct predictions (TP + TN) and the total predictions. It is calculated using Equation (1).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

- Precision. It is the ratio of the correct positive predictions and the total positive predictions (TP + FP). It is calculated using Equation (2).

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

- Sensitivity (or Recall). It is the rate of true positives that were correctly detected by the model; that is, the relationship between TP and the total of real cases of that class. It is calculated using Equation (3).

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

- F1-Score. It is the predictive ability of the model. It is the harmonic mean between precision and recall. It is calculated using Equation (4).

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \quad (4)$$

- Specificity. This metric represents the ratio between correctly predicted negative instances (TN) and the total number of actual negative cases (TN + FP). It is calculated using Equation (5).

$$Specificity = \frac{TN}{TN + FP} \quad (5)$$

2.2.4. Hardware and Software

Hardware and software were used to build the model. Regarding hardware, a PC with the following features was employed: 16 GB graphics card, 100 GB HDD, 16 GB RAM, 8th generation Intel Core i7, 16 GB NVIDIA GTX 1050, and Windows 10 OS.

Wrist-worn wearable devices are used to collect physiological data. A wearable is an electronic device (equipped with sensors and processors) to monitor in real time the location and health of the human body that carries it. Due to the miniaturization of electronic components, these devices are very comfortable and non-intrusive for the user. During the last decade, these devices have expanded their usefulness, especially in the field of occupational medicine, telemedicine and clinical studies, both ambulatory and preventive. Wrist-worn wearable devices provide physiological data such as electrodermal activity (EDA), heart rate (HR), and temperature. These physiological data are essential to monitor stress and implement suggestions to combat it.

Regarding the software, Python-based libraries were used, such as OS (to interact with the operating system), Keras 3.7.0 and TensorFlow 2.18.0 (to build and train machine learning models), Scikit-learn 1.6.0 (to classify, cluster, and reduce dimensionality), Pandas 2.2.3 (for data manipulation and analysis), NumPy 2.0.2 (to perform numerical calculations), Imbalanced-Learn 0.14.2 (to classify unbalanced classes). The Adam algorithm was employed to optimize and tune the neural network weights during training. In this context, Google Colab (with Python 3.9) was used as a training environment using GPU. Finally, Jupyter Notebook 6.5.7 was used as an integrated development environment.

## 3. Results and Discussion

In this section, each of the phases of CRISP-DM are addressed, namely business understanding, data understanding, data preparation, modeling, and evaluation [12].

### *3.1. Business Understanding Phase*

This phase is important because it aligns the business approach with the data mining approach. The business objective is to support the detection of work stress levels through three physiological parameters: heart rate, body temperature, and electrodermal activity. For this, a model based on convolutional neural networks is developed and evaluated to obtain an accuracy value greater than 95%, with respect to the identification of three classes: non-stress, medium stress, and high stress. Then, the model is implemented within a mobile application that provides a user-friendly and easy-to-use interface.

### *3.2. Data Understanding Phase*

A dataset was collected from the Dryad open data platform [13]. This dataset was obtained through an Empatica E4 wristband [14], which collected physiological information from 15 hospital nurses during the COVID pandemic. Each stress case is a record containing 3 variables and registers one of the following labels: 0 (No stress), 1 (medium stress), and 2 (high stress), as described in Table 1. These labels were obtained through automatic detection using physiological characteristics such as electrodermal activity (EDA), heart rate (HR), and skin temperature.

**Table 1.** Data description.

| Variable | Description |
|---|---|
| timestamp | Timestamp |
| w_eda | Electrodermal activity |
| w_hr | Heart rate |
| w_temp | Body temperature |
| w_label | Stress level per 40 timestamps: non-stress (0), medium stress (1), high stress (2). |

In total, a sample of 7,420,866 stress cases was collected and divided into three classes: non-stress (1,974,543), medium stress (709,011), and high stress (4,737,312).

### *3.3. Data Preparation Phase*

The physiological signals were segmented using a 10 s sliding window sampled at 4 Hz, resulting in windows composed of 40 consecutive time steps and three physiological variables: w_eda, w_hr, and w_temp. Each window represents a training instance for the CNN. After segmentation, the complete dataset contained 7,420,866 windows, which were subsequently divided into training and test subsets.

First, data normalization. This activity is necessary because the data types are heterogeneous. Furthermore, it allows features with small dimensions to be considered during the

model training phase. For this, MinMaxScaler function was used to scale each feature in a given range (0 to 1), except for w_label. Table 2 shows a brief description of the normalized data. Each record is a sample corresponding to the physiological values of an individual.

**Table 2.** Brief normalized columns for each input sample or stress cases.

| timestamp | w_eda | w_hr | w_temp | w_label |
|---|---|---|---|---|
| $t_1$ | 0.11 | 0.37 | 0.57 | |
| $t_2$ | 0.11 | 0.37 | 0.57 | |
| $t_3$ | 0.11 | 0.37 | 0.57 | |
| $t_4$ | 0.11 | 0.37 | 0.57 | |
| $t_5$ | 0.11 | 0.37 | 0.57 | 2 |
| . | . | . | . | |
| . | . | . | . | |
| $t_{40}$ | 0.11 | 0.37 | 0.57 | |

Second, data construction. In the sample, w_label stores the stress categories, which correspond to three possible values: non-stress, medium stress, and high stress. In this variable, there is an imbalance related to the values for each category, which can cause a bias toward the majority class or category. To mitigate this issue, it is necessary to apply the Synthetic Minority Over-sampling Technique (SMOTE) to the dataset.

SMOTE is a technique that generates new instances for the minority class; that is, it increases the number of minority class samples instead of simply duplicating existing ones [15]. To do this, new samples are created by interpolating between nearby instances within the minority class. In this context, SMOTE focuses more on the feature space of the minority class rather than on the sample space.

The dataset was divided into two subsets: approximately 80% was allocated to the training set, while the remaining 20% was assigned to the test set. Table 3 presents the distribution of the training and test sets. It is important to emphasize that the SMOTE technique was applied exclusively to the training set after the dataset was split. Consequently, the test set remained unchanged, ensuring an unbiased evaluation of the model and preventing data leakage.

**Table 3.** Dataset for training and testing.

| Class | Original Dataset | 80% for Training (Approx.) | 20% for Testing (Approx.) |
|---|---|---|---|
| 0 | 1,974,543 | 1,579,634 | 394,909 |
| 1 | 709,011 | 567,209 | 141,802 |
| 2 | 4,737,312 | 3,789,850 | 947,462 |
| Total | 7,420,866 | 5,936,693 | 1,484,173 |

To improve the reproducibility of the proposed methodology, the dimensions of the dataset are reported at each stage of the preprocessing pipeline. Specifically, the outputs obtained after temporal segmentation, dataset partitioning, Min-Max normalization, SMOTE oversampling, and the final tensor preparation for the 1D CNN are presented below. These dimensions demonstrate the consistency between the reported dataset size and the input shape of the proposed CNN architecture.

- Dataset after segmentation:

$$\text{x.shape} = (7{,}420{,}866, 40, 3)$$

$$\text{y.shape} = (7{,}420{,}866,)$$

$$\text{Total number of windows} = 7{,}420{,}866$$

- After the train_test_split:

x_train.shape = (5,936,693, 40, 3)

x_test.shape = (1,484,173, 40, 3)

y_train.shape = (5,936,693,)

y_test.shape = (1,484,173,)

Total = 7,420,866

- After MinMaxScaler:

x_train_flat.shape = (5,936,693, 120)

x_test_flat.shape = (1,484,173, 120)

- After SMOTE:

x_train_smote.shape = (11,369,550, 120)

y_train_smote.shape = (11,369,550,)

- Final inputs for CNN

x_train.shape = (11,369,550, 40, 3)

x_test.shape = (1,484,173, 40, 3)

where

- x: Input dataset after temporal segmentation, where each sample corresponds to a window of 40-time steps and 3 physiological features (w_eda, w_hr, and w_temp). Its shape is (number_of_samples, 40, 3).
- y: Target labels associated with each segmented window. Its shape is (number of samples).
- x_train: Training subset obtained after the 80/20 train-test split. Its shape is (number of training samples, 40, 3).
- x_test: Testing subset obtained after the train-test split. Its shape is (number of testing samples, 40, 3).
- y_train: Class labels corresponding to x_train.
- y_test: Class labels corresponding to x_test.
- x_train_flat: Reshaped version of x_train from (N, 40, 3) to (N, 120) to enable feature scaling and SMOTE processing.
- x_test_flat: Reshaped version of x_test with dimensions (N, 120).
- x_train_smote: Balanced training dataset obtained after applying SMOTE to x_train_flat. After oversampling, it is reshaped back to (N, 40, 3) before being used as input to the CNN.
- y_train_smote: Class labels associated with the balanced training dataset generated by SMOTE.

### *3.4. Modeling Phase*

Two activities were carried out, namely hyperparameter tuning and model development. The configured hyperparameters were as follows:

- Batch size. It is the number of samples processed simultaneously during model training. In this case, a value of 500 was used.
- Epochs. It refers to the number of complete passes through the training dataset. In this case, the model was trained for 35 epochs.

https://doi.org/10.3390/engproc2026153042

- EarlyStopping. It is the technique that stops training when the model performance stop improving. This prevents overfitting and reduces training time. For this, 3 parameters were used, namely: (i) monitor = 'val loss', to monitor the validation loss. (ii) patience = 3, which stops training if no improvement is observed for 3 consecutive times. (iii) restore_best_weights = 'True', which restores the model weights from the epoch with the best monitored performance".

Regarding model development. At the beginning of the network, a one-dimensional convolutional layer (Conv1D) followed by a one-dimensional max-pooling layer (MaxPooling1D) was incorporated to automatically extract local feature patterns and reduce the dimensionality of the learned representations. Although the dataset is tabular, the input variables were arranged as a one-dimensional sequence of features, enabling the convolutional operation to capture local interactions among neighboring features before they are processed by the fully connected classification layers. The algorithm describing the neural network architecture is presented below.

1: model.add(Conv1D(filters = 32, kernel_size = 3, activation = 'relu', input_shape = (40,3)))

2: model.add(MaxPooling1D(pool_size = 2))

3: model.add(Flatten())

4: model.add(Dense(16, activation = 'relu'))

5: model.add(Dense(8, activation = 'relu'))

6: model.add(Dense(6, activation = 'relu'))

7: model.add(Dense(3, activation = 'softmax'))

8: early_stop = EarlyStopping(monitor = 'val_loss', patience = 3, restore_best_weights = True)

9: model.compile(optimizer = 'adam', loss = 'sparse_categorical_crossentropy', metrics = ['accuracy'])

The architecture is described as follows.

1: Conv1D to apply 32 filters of size 3 to extract local patterns between consecutive variables. Each sample contains a time window of 40 instants and 3 physiological variables.

2: MaxPooling1D to reduce the dimensionality of feature maps, preserving the most relevant information and decreasing computational cost.

3: Flatten to transform feature maps into a one-dimensional vector.

4: Dense(16) is the first fully connected layer for high-level learning.

5: Dense(8) is the second hidden layer.

6: Dense(6) is the third hidden layer.

7: Dense(3, softmax) is the output layer for three-class classification.

8: EarlyStopping to stop training when validation loss does not improve, restoring the best weights.

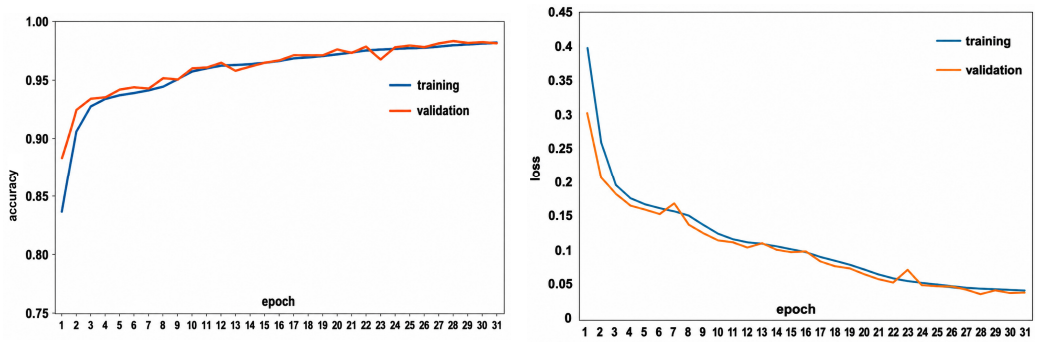
9: Adam + sparse_categorical_crossentropy as an optimizer and loss function, both of which are appropriate for multiclass classification. In this way, the temporal information of the physiological signals is preserved, enabling the CNN to automatically learn the patterns associated with stress.

### *3.5. Model Evaluation Phase*

The model was evaluated according to two considerations: (i) accuracy and loss for both training and validation, and (ii) metrics based on the confusion matrix (accuracy, precision, recall, and F1-score).

The accuracy gradually increases and stabilizes at 0.98 by epoch 30 for both the training and validation phases. In contrast, the loss gradually decreases and stabilizes

at 0.02 by epoch 30 for both phases. Figure 2a,b illustrate the behavior of the accuracy and loss, respectively.




(**a**) (**b**)

**Figure 2.** (**a**) Accuracy and (**b**) loss during training and validation.

Figure 3 shows the confusion matrix. Four values (TP, FN, FP, and TN) were obtained for each class: non-stress (0), medium stress (1), and high stress (2). Five metrics were calculated: precision, accuracy, recall, specificity, and F1-score. For class 0, the model achieved 95.83% accuracy and 94% precision. For class 1, the values obtained were 98.64% accuracy and 93.05% precision. For class 2, the model achieved 96.49% accuracy and 96.41% precision. In this context, the overall accuracy of the model reached 95.47%. Table 4 summarizes these results, including other metrics such as recall, specificity, and F1-score.

| | | | |
|---|---|---|---|
| Clase 0 | 355,695 | 5000 | 34,214 |
| Clase 1 | 10,000 | 131,365 | 437 |
| Clase 2 | 12,704 | 4807 | 929,951 |
| | Clase 0 | Clase 1 | Clase 2 |

**Figure 3.** Confusion matrix.

The results obtained by our model were compared with standard machine learning baselines such as Logistic Regression, Random Forest, Gradient Boosting, and Linear SVM. Due to the extremely large number of training instances, Linear SVM was used instead of kernel SVM, as the latter has quadratic computational complexity and is not feasible for millions of samples. All models were trained using the same training/testing partition. Classical machine learning models received flattened feature vectors ($40 \times 3 = 120$ features), whereas the proposed CNN directly processed the temporal representation of each window $(40, 3)$. SMOTE was applied exclusively to the training set to mitigate class imbalance, while the test set remained unchanged for unbiased evaluation. Table 5 shows this comparison.

**Table 4.** Confusion matrix and metrics.

| Class | Confusion Matrix | Metrics | Overall Accuracy |
|---|---|---|---|
| Class 0 | TP: 355,695<br>FN: 39,214<br>FP: 22,704<br>TN: 1,066,560 | Precision: 94.00%<br>Recall: 90.07%<br>Specificity: 97.92%<br>F1-score:91.99%<br>Accuracy: 95.83% | 95.47% |
| Class 1 | TP: 131,365<br>FN: 10,437<br>FP: 9807<br>TN: 1,332,564 | Precision: 93.05%<br>Recall: 92.64%<br>Specificity: 99.27%<br>F1-score: 92.85%<br>Accuracy: 98.64% | |
| Class 2 | TP: 929,951<br>FN: 17,511<br>FP: 34,651<br>TN: 502,060 | Precision: 96.41%<br>Recall: 98.15%<br>Specificity: 93.54%<br>F1-score: 97.27%<br>Accuracy: 96.49% | |

**Table 5.** Performance comparison of the proposed CNN model with standard machine learning baselines.

| Model | Input Representation | Overall Accuracy (%) | Macro F1 (%) |
|---|---|---|---|
| Gradient Boosting | Flattened windows (120 features) | 94.21 | 92.87 |
| Linear SVM | Flattened windows (120 features) | 90.35 | 88.64 |
| Logistic Regression | Flattened windows (120 features) | 88.72 | 86.91 |
| Random Forest | Flattened windows (120 features) | 93.82 | 92.15 |
| Proposed 1D CNN | Temporal windows (40 × 3) | 95.47 | 94.20 |

### *3.6. App Development*

In the development phase, the Waterfall methodology was used to conduct three activities: functional requirements analysis, development, and functional testing.

Regarding the requirements analysis, two main functional requirements were defined: (i) reading physiological data extracted from a wrist-worn wearable device, and (ii) predicting the user's stress level using the algorithm. Similarly, two non-functional requirements were identified: (i) compatibility with mobile operating systems (Android and iOS), and (ii) an intuitive and user-friendly interface.

Regarding development, the MVC (Model-View-Controller) pattern was used to design the software architecture. In this context, the user sends a request to a controller function. The controller, in turn, manages the workflow and decides which model function will serve the user's request. Furthermore, the controller determines which view file will be used as the user interface. The model contains the business logic, including processing functions and a TensorFlow-based module for predicting stress levels. Figure 4 shows the proposed architecture.

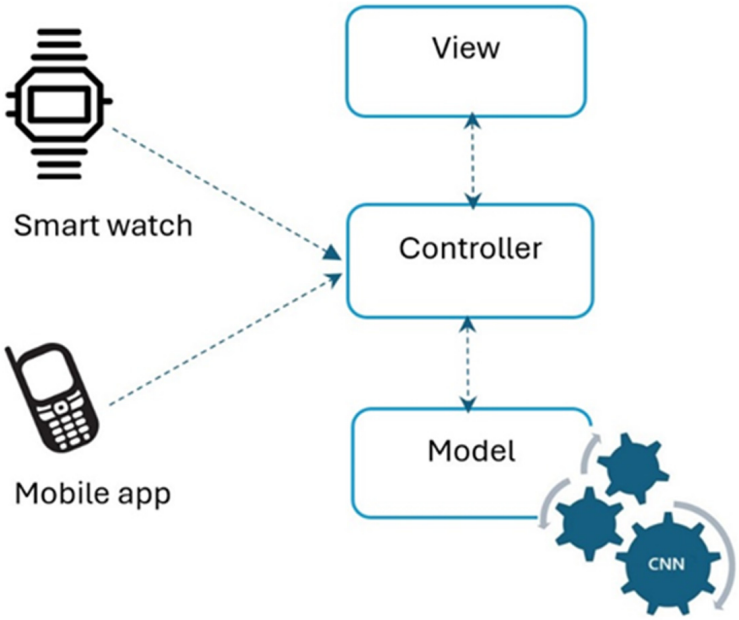



**Figure 4.** MVC architecture for the proposed solution.

Regarding the development of the app, it was approached through six steps. (i) Creation of the project using React Native 0.76.0 (ii) Installation of dependencies. (iii) Creation of a Python-based API, which serves as an interface between the mobile application and the stress prediction model. (iv) Development of functionality so that the app can interact with the prediction model through the API. (v) Definition of a method to extract the classification result. (vi) Creation of graphical user interface components.

Regarding functional testing, Figure 5 shows the app's functionality. Using the button (a triangle inside a circle), the app captures data from the wearable device. The app then sends the data to the controller URL and retrieves the stress level.

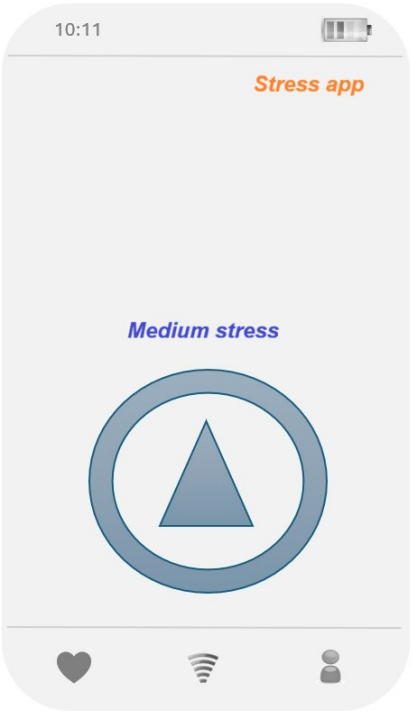



**Figure 5.** App interface.

### *3.7. Comparison of Reported Performance Across Stress Detection Modalities*

Previous studies have developed stress detection models using a variety of sensing modalities. It should be noted that this comparison is not intended to demonstrate the superiority of the proposed deep neural network over other classification approaches. Such a comparison would not be methodologically appropriate because previous studies were conducted using different datasets, feature extraction procedures, sensing modalities, and experimental protocols. Instead, the purpose of this comparison is to contextualize the performance of the proposed approach within the broader field of stress detection research by illustrating the range of predictive performance achieved through different data acquisition techniques. Therefore, the reported results should be interpreted as a qualitative reference to the current state of the art rather than as a direct comparison between classification algorithms.

Based on the above context, a convolutional neural network along with optimization algorithms (African Buffalo and Artificial Bee Colony) was designed to predict stress through sentiment analysis [16]. For detection, physiological data such as heart rate, BMI, blood pressure, glucose level, and skin response were used. Using this information, the study achieved an accuracy of 96%. In addition, a model was developed with a one-dimensional CNN combined with a multilayer perceptron neural network to classify three emotional states, namely baseline, stress, and amusement. The model achieved an accuracy of 99.55% in classifying these emotions [17].

A stress classification algorithm based on Doppler spectral features for a CNN was developed by [18]. The approach uses electroencephalogram (EEG) signals and trains a one-dimensional CNN, which consists of two stages. The first stage processes EEG data from Stroop tests, arithmetic tasks, and mirror tasks, while the second stage classifies features according to stress intensity (high, moderate, and low). The classification scheme achieved 95% accuracy and precision. In addition, Ramteke et al. developed a CNN model using EEG images and obtained 95.71% accuracy [19]. Mane et. al also used an EEG dataset [20]. For this purpose, a hybrid Long Short-Term Memory (LSTM) and CNN model was designed for stress detection from EEG signals. The EEG signals were decomposed into alpha, beta, and theta bands to generate images, which were then processed by the LSTM network. The model achieved 97.8% precision.

A machine learning model was developed for stress detection from photoplethysmography (PPG) signals. The model processes PPG signals and converts them into spectrograms, which capture both time and frequency characteristics of the dataset. The PPG dataset is based on pulse signal variations. The results showed training and validation accuracies of 98.87% and 93.28%, respectively [21]. A CNN was trained using hemodynamic response function (HRF) in functional near-infrared spectroscopy (fNIRS). The model uses a one-dimensional CNN with fully connected layers and pooling operations. Stress detection using the model achieved 98% accuracy [22]. In addition, a CNN-LSTM model was designed to detect driver stress. A non-invasive dataset including ocular, vehicle, and environmental data was used, achieving an accuracy of 95.5% [23].

## 4. Conclusions

This research developed a convolutional neural network-based model to detect stress levels using a wrist-worn wearable device. This research used three physiological parameters to estimate stress levels: heart rate, body temperature, and electrodermal activity. To this end, the study was approached from two perspectives: data mining and web development. Regarding the data mining approach, CRISP-DM was used as the methodology, which includes five main tasks: business understanding, data understanding, data preparation, modeling, and evaluation. As a result, a model was trained and validated to classify three

levels: non-stress, medium stress, and high stress. Metrics based on the confusion matrix were obtained: accuracy, precision, recall, specificity, and F1-score. For all of these metrics, the model exceeded 90%; it is worth noting that an overall accuracy of 95.47% was obtained.

Next, the model was implemented within an app using MVC (as a software architecture) and waterfall model (as a software development methodology). This work can be expanded in several ways. The development of a model using other deep learning algorithms and generative artificial intelligence is key. In this context, new physiological parameters can be considered, such as blood volume pulse (BVP) and respiratory rate (RR). Thus, it will be possible to evaluate the robustness of the results and their impact on stress level prediction. In this way, the model could serve as a useful tool for occupational and preventive medicine.

**Author Contributions:** Conceptualization, M.R.-L.; Formal analysis, M.R.-L. and J.C.M.-T.; Investigation, J.C.M.-T., M.R.-L. and F.A.T.-M.; Methodology, J.C.M.-T. and M.R.-L.; Software, F.A.T.-M.; Validation, J.C.M.-T. and M.R.-L.; Writing—original draft Preparation, J.C.M.-T.; Writing—review and editing Preparation, J.C.M.-T. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** The analyses were conducted using a publicly available anonymized dataset, and no new data were collected from human participants. Ethical review and approval were waived for this study because the research involved non-invasive physiological measurements acquired through a wrist-worn wearable device and posed minimal risk to participants. All data were anonymized prior to analysis, and no personal identifiable information was collected, accessed, or processed at any stage of the study.

**Informed Consent Statement:** Informed consent was obtained from all subjects involved in the study.

**Data Availability Statement:** Data are available in this manuscript.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Baran, K. Smartphone Thermal Imaging for Stressed People Classification Using CNN+MobileNetV2. *Procedia Comput. Sci.* **2023**, *225*, 2507–2515. [CrossRef]
2. Alskaf, E.; Scannell, C.M.; Crawley, R.; Suinesiaputra, A.; Masci, P.G.; Young, A.; Perera, D.; Chiribiri, A. Qualitative Stress Perfusion American Heart Association Plot and Outcome Prediction Using Artificial Intelligence. *Inform. Med. Unlocked* **2024**, *49*, 101537. [CrossRef] [PubMed]
3. Ciccarelli, M. Wearable Devices for Stress Detection: Insights on Exploring. *Procedia Comput. Sci.* **2025**, *253*, 2014–2023. [CrossRef]
4. Cannady, R.; Warner, C.; Yoder, A.; Miller, J.; Crosby, K.; Elswick, D.; Kintziger, K.W. The Implications of Real-Time and Wearable Technology Use for Occupational Heat Stress: A Scoping Review. *Saf. Sci.* **2024**, *177*, 106600. [CrossRef]
5. Lee, Y.; Chung, H.; Hsia, S.; Chan, O.; Lee, E.; Lin, K.; Lin, J.; Wu, H. Pediatrics & Neonatology Heart Rate Variability as an Early Predictor for Children with SARS-CoV-2-Associated Encephalopathy. *Pediatr. Neonatol.* **2025**, *66*, 527–533. [CrossRef] [PubMed]
6. Ishikawa, Y.; Sugio, T.; Shiga, K.; Izumi, K.; Minato, K.; Kitazawa, M.; Hanashiro, S.; Takemura, R.; Uchida, H.; Kishimoto, T. Electrodermal Activity and Skin Temperature Characteristics Related to Stress and Depression: A 4-Week Observational Study of Office Workers. *J. Affect. Disord. Rep.* **2025**, *20*, 100877. [CrossRef]
7. Larsen, C.F.; Pedersen, M. Comparison of Convolutional Neural Networks for Classification of Vocal Fold Nodules from High-Speed Video Images. *Eur. Arch. Oto-Rhino-Laryngol.* **2023**, *280*, 2365–2371. [CrossRef] [PubMed]
8. Jamsrandorj, A.; Jung, D.; Kumar, K.S.; Arshad, M.Z.; Lim, H.; Kim, J.; Mun, K.R. View-Independent Gait Events Detection Using CNN-Transformer Hybrid Network. *J. Biomed. Inform.* **2023**, *147*, 104524. [CrossRef] [PubMed]
9. Theissler, A.; Thomas, M.; Burch, M.; Gerschner, F. ConfusionVis: Comparative Evaluation and Selection of Multi-Class Classifiers Based on Confusion Matrices. *Knowl.-Based Syst.* **2022**, *247*, 108651. [CrossRef]
10. Ahmad, S.; Ansari, S.U.; Haider, U.; Javed, K.; Rahman, J.U.; Anwar, S. Confusion Matrix-Based Modularity Induction into Pretrained CNN. *Multimed. Tools Appl.* **2022**, *81*, 23311–23337. [CrossRef]

11. Luque, A.; Carrasco, A.; Martín, A.; de las Heras, A. The Impact of Class Imbalance in Classification Performance Metrics Based on the Binary Confusion Matrix. *Pattern Recognit.* **2019**, *91*, 216–231. [CrossRef]
12. Schröer, C.; Kruse, F.; Gómez, J.M. A Systematic Literature Review on Applying CRISP-DM Process Model. *Procedia Comput. Sci.* **2021**, *181*, 526–534. [CrossRef]
13. Hosseini, S.; Gottumukkala, R.; Katragadda, S.; Bhupatiraju, R.T.; Ashkar, Z.; Borst, C.W.; Cochran, K. A Multimodal Sensor Dataset for Continuous Stress Detection of Nurses in a Hospital. *Sci. Data* **2022**, *9*, 255. [CrossRef] [PubMed]
14. Stritzelberger, J.; Kirmse, M.; Borutta, M.C.; Gollwitzer, S.; Reindl, C.; Welte, T.M.; Hamer, H.M.; Koehn, J. Validity of Empatica E4 Wristband for Detection of Autonomic Dysfunction Compared to Established Laboratory Testing. *Diagnostics* **2025**, *15*, 2604. [CrossRef] [PubMed]
15. Soomro, A.A.; Mokhtar, A.A.; Muhammad, M.B.; Saad, M.H.M.; Lashari, N.; Hussain, M.; Palli, A.S. Data Augmentation Using SMOTE Technique: Application for Prediction of Burst Pressure of Hydrocarbons Pipeline Using Supervised Machine Learning Models. *Results Eng.* **2024**, *24*, 103233. [CrossRef]
16. Alruily, M. Sentiment Analysis for Predicting Stress among Workers and Classification Utilizing CNN: Unveiling the Mechanism. *Alex. Eng. J.* **2023**, *81*, 360–370. [CrossRef]
17. Li, R.; Liu, Z. Stress Detection Using Deep Neural Networks. *BMC Med. Inform. Decis. Mak.* **2020**, *20*, 285. [CrossRef] [PubMed]
18. Naren, J.; Babu, A.R. EEG Stress Classification Based on Doppler Spectral Features for Ensemble 1D-CNN with LCL Activation Function. *J. King Saud Univ.-Comput. Inf. Sci.* **2024**, *36*, 102013. [CrossRef]
19. Ramteke, R.B.; Gajbhiye, G.O.; Thool, V.R. Acute Mental Stress Level Detection: ECG-Scalogram Based Attentive Convolutional Network. *Frankl. Open* **2025**, *10*, 100233. [CrossRef]
20. Mane, S.A.M.; Shinde, A. StressNet: Hybrid Model of LSTM and CNN for Stress Detection from Electroencephalogram Signal (EEG). *Results Control Optim.* **2023**, *11*, 100231. [CrossRef]
21. Kumar, G.S.; Ankayarkanni, B. Leveraging Conv-XGBoost Algorithm for Perceived Mental Stress Detection Using Photoplethysmography. *Intell. Med.* **2025**, *11*, 100209. [CrossRef]
22. Hakimi, N.; Jodeiri, A.; Mirbagheri, M.; Setarehdan, S.K. Proposing a Convolutional Neural Network for Stress Assessment by Means of Derived Heart Rate from Functional near Infrared Spectroscopy. *Comput. Biol. Med.* **2020**, *121*, 103810. [CrossRef] [PubMed]
23. Mou, L.; Zhou, C.; Zhao, P.; Nakisa, B.; Rastgoo, M.N.; Jain, R.; Gao, W. Driver Stress Detection via Multimodal Fusion Using Attention-Based CNN-LSTM. *Expert Syst. Appl.* **2021**, *173*, 114693. [CrossRef]


**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.