Natural Resources for Human Health
eISSN: 2583-1194

Original Research

Received 15 March 2026
Revised 20 April 2026
Accepted 25 May 2026

Natural Resources for Human Health 2026, Vol. 6 (1s): 196–212

KEYWORDS:
Carpal Tunnel Syndrome (CTS), Deep Learning, RNN-LSTM, Ergonomics, Repetitive Strain Injury, Wearable Sensors

Intelligent Ergonomic Risk Assessment through Hybrid RNN–LSTM Modeling for Carpal Tunnel Syndrome Prediction

Nikita Gautam[1]*, Dr. Sunil Kumar Gupta[2], Dr. Amit Shrivastava[3]

[1]Research scholar, [2,3] Professor
[1,2] Department of Electrical & Electronics Engineering, Poornima University, Jaipur (India)
[3]Professor, Department of Electrical Engineering, Poornima Institute of Engineering and Technology, Jaipur (India)
E-mail: nikitagtm@gmail.com[1], sunilsunel1980@gmail.com[2], amit.shrivastava@poornima.org[3]
*Corresponding Author: nikitagtm@gmail.com

ABSTRACT: Human activity recognition (HAR) has emerged as a critical research topic in intelligent healthcare systems and ergonomic risk analysis mainly due to the development of wearable sensor-based technologies and computing intelligence. This paper proposes a deep learning architecture that incorporates Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) models for a dynamic, time-based analysis of ergonomic motion data in order to diagnose Carpal Tunnel Syndrome (CTS) early. Within the suggested framework, the RNN module is suitable for capturing short-run temporal dependencies, and the LSTM module is appropriate for capturing long-run contextual correlation, thus improving the process of learning sequential patterns. The RNN-LSTM architecture is found to be better in accuracy, robustness, and computational efficiency in identifying abnormal wrist and hand postures associated with the development of CTS. The research design involves data creation, multimodal signal acquisition, preprocessing, feature extraction, and end-to-end supervised model training. The experimental results have shown that the hybrid model minimizes data noise and false-positive alerts and enhances the prediction of CTS risk at an early stage. Additionally, the architecture facilitates real-time data processing and integration of adaptive feedback by enabling wearables with the Internet of Things (IoT). The suggested system creates a scalable base to identify repetitive strain injuries and leads to the development of smart and preventive healthcare and ergonomic monitoring solutions.

INTRODUCTION

Carpal tunnel syndrome affects approximately 3.8% of the population each year. It is one of the most common entrapment neuropathies [1, 2]. Fibrotic changes in sub-synovial connective tissues within the carpal tunnel and limitation on median nerve (MN) movement are the major signs and symptoms indicative of carpal tunnel syndrome [3]. Currently, few prognostic indicators accurately predict the outcomes of common interventions like surgical carpal tunnel release and corticosteroid injections. The primary research problem is how to promptly identify CTS specifically in work environments where repetitive hand motions could present a threat. A novel hybrid deep learning model was introduced to tackle this problem. RNN and LSTM networks are combined to create a hybrid deep learning model that effectively analyzes dynamic ergonomic data using the best form of both architectures. The process starts with creating a dataset that is gathered by using a detection device that is attached to an Arduino UNO [4]. It captures motions and flexions in real-time, hence ensuring reliable data collection that covers all possible ergonomic positions. Processing follows data gathering and involves lead normalization as well as noise reduction to make such data open for analysis and optimization purposes. Two techniques for feature extraction and selection are used here, namely i) Principal Component Analysis (PCA) [5] and ii) Recursive Feature Elimination (RFE) [6]. The optimization of PCA and RFE is to maintain the high-potential attributes of CTS and improve the analysis results. The hybrid model, which is an important component of the chosen research methodology, involves RNN and LSTM networks. Predicted accuracy and reliability of the model are improved by taking advantage of RNN's capability to handle sequential information as well as LSTM's ability to retain long-term dependencies. The model is trained, validated, and tested with the hyperparameters being adjusted continuously to enhance performance and generalization. Evaluation measures such as recall, precision, and displacement of the median nerve in dynamic scenarios clearly indicate that the model can accurately and quickly identify suspected cases of carp CTS. These metrics, in other words, show how well the model can tell the difference between good and poor ergonomic stances. A novel approach aimed at continuous ergonomic assessment seeks not only to improve diagnostic procedures but also to facilitate the early identification of problems for better preventive strategies.

Research Literature

In CTS, the cause of migration of the median nerve (MN) is highlighted by the development of sub-synovial connective tissue fibrosis. Conventional assessment still relies on static images for calculating the MN cross-sectional area, despite the increasing popularity of dynamic images. Recent years have seen significant advancements in medical image processing that employ deep learning (DL). In this research, researchers intended to use a DL object identification model, YOLOv5, to assess MN dynamics under CTS conditions. Each group, such as a control group and a CTS group, had 20 hands. The authors took ultrasonographic short-axis images of the carpal tunnel and the MN, which are used to evaluate its displacement, velocity, and motion during finger flexion and extension. According to YOLOv5 model results presented, the recall score was 0.956, while the precision score was 0.953. The control group had radial-ulnar displacement of 3.56 mm for MN, while CTS only measured 2.04 mm. The control group's velocity was 4.22 mm/s, while the CTS group's velocity was 3.14 mm/s. Nevertheless, there were marginal decreases in scores between the two groups [7]. The paper proposes a novel classification algorithm using LSTM for detecting CTS through biomechanical data. It achieves 93% accuracy, addressing the need for an accessible, non-invasive assessment method for early-stage CTS detection [8]. The paper does not discuss a hybrid RNN-LSTM model for dynamic analysis or early detection of CTS risk factors. It focuses on deep learning models for diagnosis, prediction, and monitoring of CTS progression and treatment effectiveness. In this paper, the elevated pressure in the carpal tunnel causes compression of the median nerve; however, there are also complications after surgical treatment [9].

Research Gaps and Objectives

CTS is a major health risk, especially for those whose professions require them to frequently help others. It is imperative that CTS be detected promptly and at an early stage to effectively manage its symptoms. The challenge lies in dynamically assessing ergonomic data to differentiate between correct and incorrect hand postures in real-time — a capability that can lead to the early detection of CTS. Thus, in the present study, a hybrid deep learning model combining RNNs and LSTM networks is developed. The focus for this hybrid model is on using the RNN for contextual immediacy and the LSTM for long-term memory quality. Therefore, this method attempts to enhance the velocity and accuracy in identifying CTS using those models collectively. This technique requires the continuous input of ergonomic data at a constant pace, followed by several procedures, including feature extraction, data normalization, and dynamic analyses. A good model provides highly accurate measurements. The effectiveness of this approach is reflected in a reduction in false positives and an increase in the rate of early-stage CTS signs.

- Design an ergonomic model using RNN and the proposed LSTM to facilitate effective changes and adaptations based on ergonomic data.
- Assess the overall performance of the hybrid model in distinguishing between correct and incorrect ergonomic postures.
- Ensure improved power quality to guarantee uninterrupted power supply.
- Evaluate the model's efficiency on unseen data to ensure its general ability across diverse user interactions.
- Determine an optimal distance threshold and analyze how different steps in the hybrid deep learning model, along with associated hyperparameters, affect the model's performance and efficiency.

Problem Development

Carpal tunnel syndrome (CTS) is one of the most prevalent conditions affecting both younger and older populations, which can result in disability. The disability associated with CTS is a global concern. This paper presents a novel deep learning architecture that incorporates both RNN and LSTM networks to identify ergonomic patterns for the early diagnosis of CTS. RNN contributes contextual information to improve response speed, while LSTM, which enables the model to retain information over long periods, helps achieve more accurate identification of incorrect hand postures in the shortest possible time. The approach aims to minimize false positive rates and increase early-stage detection of CTS using continuous data without the need for feature extraction. The evaluation of this combined approach is based on accuracy, precision, and recall rates, together with the capability to detect CTS through dynamic ergonomic interventions, to enhance diagnostic and prevention methods.

METHODOLOGY

The methodology employs several algorithms, such as principal component analysis (PCA), Recursive feature extraction (RFE), algorithms based on recurrent RNN networks, LSTM networks, and a method known as the hybrid model. PCA is a multivariate technique for analyzing a data table with a large number of quantitative interrelated dependent variables that characterize observations. Principal components are a linear transformation of orthogonal variables derived from the extraction. These components are then used to align the correspondence between variables and observations. The result is presented in points on maps [10]. Higher dimensions can limit the value of the system state representation. More importantly, PCA is a logically sensible method for boosting feature sets because it can compress data while retaining features. In the present research, the technique of PCA is used for feature extraction. PCA is a way of reducing the dimension of the data in which a large dataset of mouse and keyboard movements is decomposed into principal components, a set of linearly uncorrelated variables. Here, simplification of RFE makes it easier to study and plot while still capturing all the necessary components that distinguish postural ergonomics.

Some subproblems can be traced to an inherent property of large datasets: the presence of irrelevant features [11]. It is evident that repeated patterns affect the performance of the classifier function and may, therefore, result in less accurate predictions. RFE helps establish which variables are essential for generating valid forecasts. This method retains the important characteristics obtained from finer-grained feature sets while simultaneously avoiding the high dimensionality of the dataset. The process of repeatedly searching to identify the optimal set of attributes is called a recursive approach. After determining the importance of attributes in a classification context, RF classifiers rank them in decreasing order of preference. The model is then retrained with the updated feature set to improve classification accuracy and remove less important features. The loop continues as long as there are additional features to be added. Figure 1 shows the workflow diagram for RFE [12].

RFE helps in identifying the most relevant features that predict CTS. The optimal subset for the predictive model is determined by methodically eliminating the least significant elements. Using the most pertinent data improves the model's precision and effectiveness. RNN is a subset of neural networks commonly used to detect patterns in sequential data [13]. These models have a generative architecture as their underlying structure. An important neural network model employed with speech or information processing is one in which the degree of the neural network fluctuates based on the duration of the data input. The difficulty with recurrent neural networks is the primary issue that arises with this kind of network. Because of this, training in RNNs is quite difficult, and because of this, it is only applied in a limited number of different areas of study. Recently, the advancement of optimization has led to a lessening of the severity of these gradient difficulties. RNNs have served as the basis for several research endeavors, many of which have become influential models, such as those involving language modeling and text recognition [14]. In this research, RNNs are used as part of a hybrid network of neural networks.

Natural Resources for Human Health
eISSN: 2583-1194

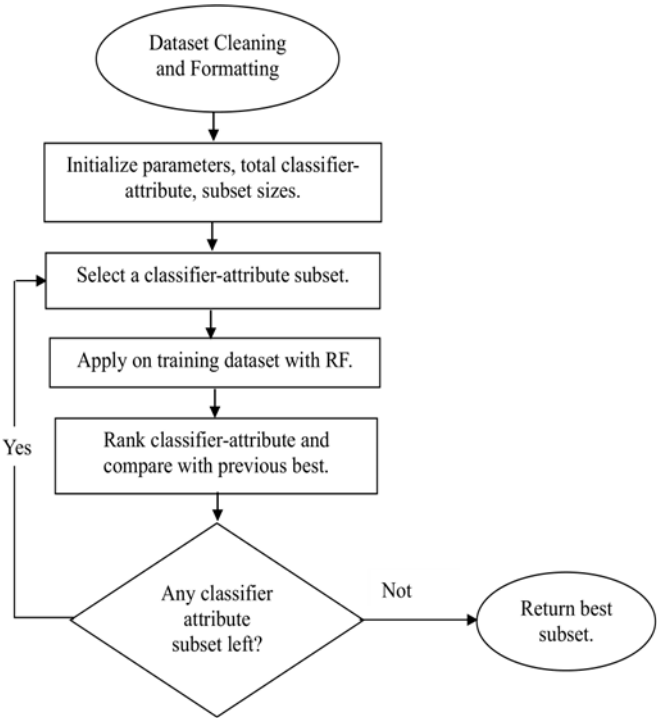

Figure 1: Workflow Diagram of Recursive Feature Elimination [12]

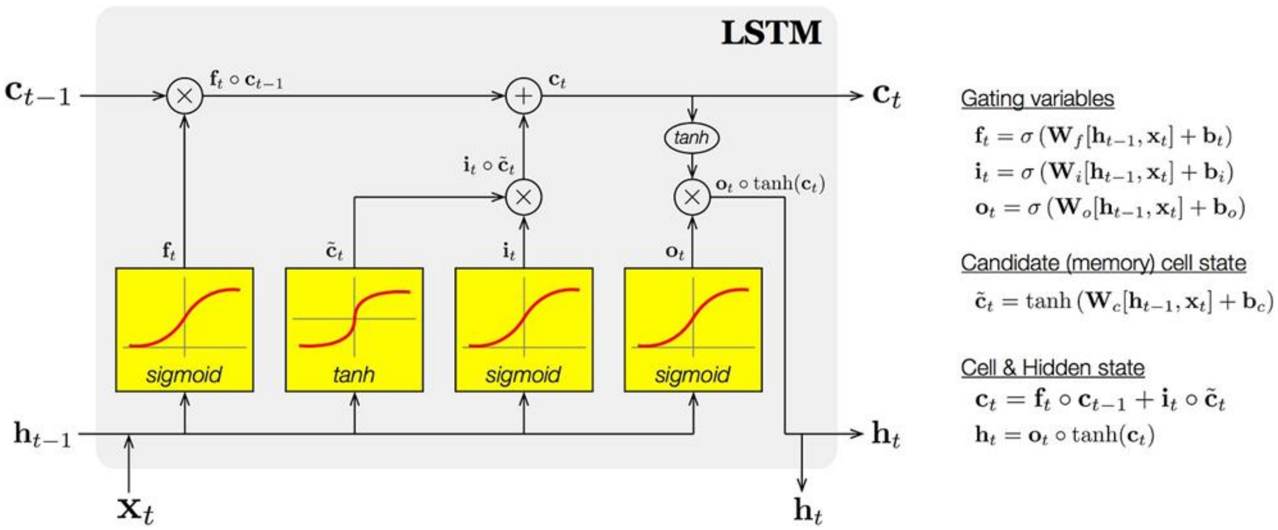

Figure 2: LSTM internal structure [15]

RNNs excel at analyzing time-series data and detecting patterns in ergonomic behaviors because they are designed to model sequential data effectively. They help recognize correlations in input feature sequences, which are crucial for understanding patterns in sequential ergonomic operations. Now, even though RNNs are quite powerful, they suffer from the vanishing gradient problem, which hinders their ability to use long-term information. While they are effective at storing memory for 3-4 instances of past iterations, a larger number of instances does not yield good results. Therefore, we don't rely solely on regular RNNs. Instead, we use an improved variation of RNNs: Long Short-Term Memory networks (LSTMs).

Long-Short Term Memory Figure 2: The novel intermediate network architecture LSTM leverages gradient-based learning to fit the data. LSTM can capture and learn long-term relationships in sequential data, making it suitable for time series forecasting. Components of LSTMs

- Forget Gate “f” (A neural network with sigmoid)
- Candidate layer “C” (A NN with Tanh)
- Input Gate “I” (A NN with sigmoid)
- Output Gate “O”(A NN with sigmoid)
- Hidden state “H” (A vector)
- Memory state “C” (A vector)
- Inputs to the LSTM cell at any step are X_t (current input), H_{t-1} (previous hidden state), and C_{t-1} (previous memory state).

Outputs from the LSTM cell are H_t (current hidden state) and C_t (current memory state). Time-series data with complex patterns can be captured by the LSTM model, which includes temporal correlations. LSTMs encompass long-term data dependencies within a single hybrid model; they are a type of RNN. In contrast to RNNs, LSTMs can acquire knowledge from distant data points because of their ability to circumvent the issue of vanishing gradients. This helps identify enduring ergonomic habits that can contribute to the development of CTS.

Hybrid Model: In machine learning, the hybrid model is an improvement by using the features of various architectures and techniques of several types of models. As a result of this research, a new model is proposed that is a fusion of both RNNs and LSTM networks. In the case of data that involves sequences with dependencies, a byproduct of both RNN and LSTM is created to solve the problem. By integrating both RNNs and LSTMs, this enables the model to effectively break down and analyze sequences of ergonomic behaviors, such as movements of the mouse and keyboard input, by fully exploiting RNNs, which easily handle sequential data for short-term patterns, and LSTMs, which retain long-term dependencies and are not affected by limitations such as the vanishing gradient problem. Evaluating this model based on its ability to analyze behavioral data and make a clear differentiation between appropriate and inappropriate ergonomic postures has been demonstrated to be effective in predicting and explaining risk factors that predispose patients to CTS. This integrated method guarantees a complete understanding of the short- and long-term ergonomic behavior patterns, which, however, enhances the efficacy of the calculated accuracy and reliability of the model in real-world circumstances.

The proposed methodology for the dynamic analysis of CTS using a hybrid deep RNN and LSTM for improved accuracy involves several key steps. Firstly, data on mouse and keyboard usage is collected and reprocessed to reduce noise and inconsistencies. Feature extraction and selection are then performed using PCA and RFE techniques, respectively. To assess the performance of the models, the dataset is divided into three parts: training, validation, and testing. A hybrid deep RNN and LSTM model is constructed to capture both short- and long-term dependencies in the data. The model is trained and fine-tuned with the help of their respective sets to avoid overfitting: the training set and the validation set. Finally, patterns related to carpal tunnel syndrome are identified by evaluating the performance of the model in the test set.

$$y = mx + c \qquad (1)$$

$$y = mx^2 + c_y \qquad (2)$$

$$y = \lambda x^2 + c_y \qquad (3)$$

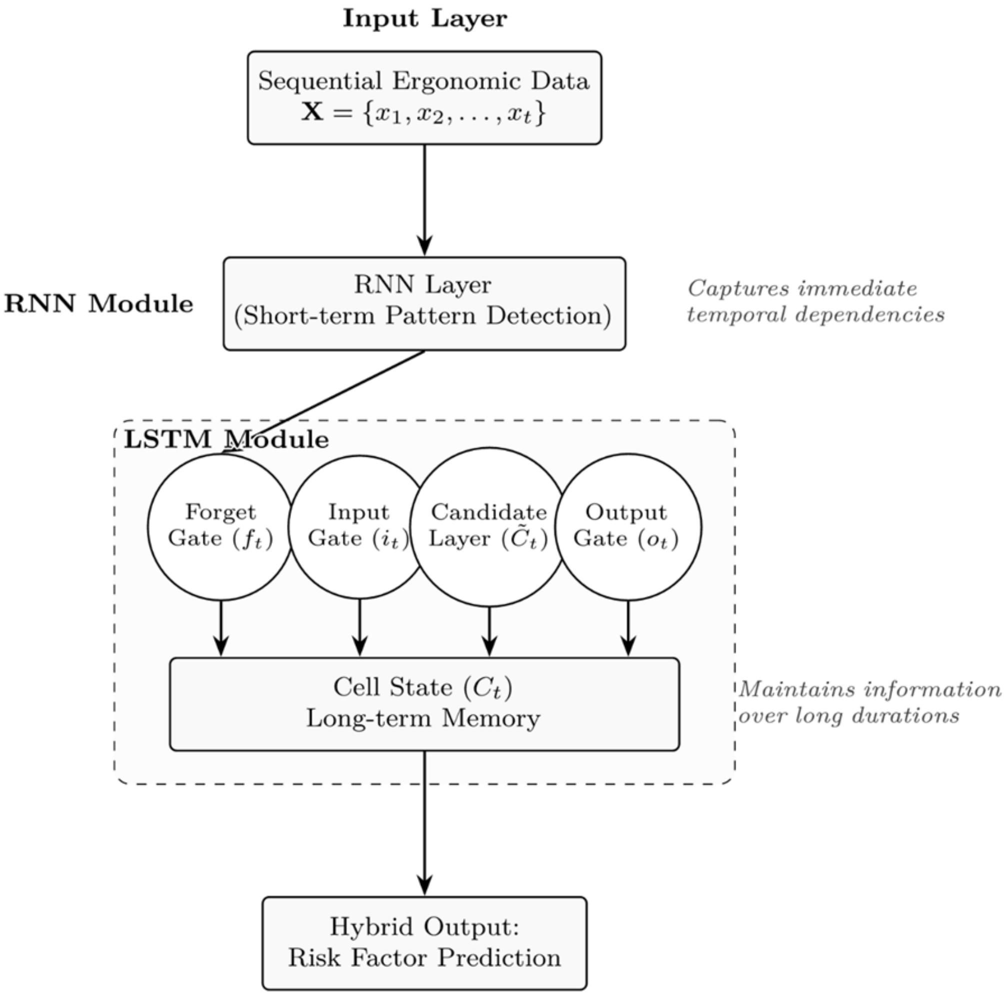

Figure 3: Architecture of the Hybrid RNN–LSTM Model for Sequential Ergonomic Risk Factor Prediction

Figure 3 shows the architecture of the proposed Hybrid RNN & LSTM model for sequential sensor data-based ergonomic risk factor prediction. Hybrid models leverage the strengths of both conventional RNNs, which excel at modelling short-term temporal relationships, and LSTMs, which can effectively capture long-term temporal patterns in time-series data. The process starts from the input layer, where the sensor data from the sequential ergonomics is converted to a time-series sequence,

$X=\{x_1,x_2,\ldots,x_t\}$,

X (x_t denotes) represents the observation of the sensor at step t. The observations here could come from the MPU6050 and flex sensors, and include acceleration, angular velocity, and bending angle readings. The input sequence is first passed through the RNN layer, where immediate temporal relationships between successive observations are captured. This layer is able to make efficient use of the local dependency and short-term motion dynamics of the sequential ergonomic data, while providing meaningful temporal feature representations.

The extracted features are then passed to the LSTM module, which is used to address the vanishing gradient issue that often faces traditional RNNs. The LSTM network has four functional components:

- The Forget Gate (f_t) decides which information from the previous state of the cell is to be thrown away.
- Input Gate (i_t) determines how much new information is to be stored in the memory.
- Candidate Layer ($\tilde{C}_t$) creates candidate memory values for the current input, and the previous hidden state.
- Output Gate (o_t) controls the information flowing from the memory cell towards the hidden output.

Gates collectively change the state of the cell (C_t), representing the long-term memory of the network. The LSTM can selectively retain the most relevant historical information, while discarding the redundant data, which would enable the model to capture long-range temporal dependency, critical to continuous human movements and posture variations in both an ergonomic and an ergonomic uncomfortable manner.

Finally, the learned temporal representations are passed to the hybrid output layer, in which the extracted features are used to make predictions of ergonomic risk factors. Combined with the motion characteristics of the short term and the sequential dependence of the long term, the hybrid RNN–LSTM network is an appropriate model for human posture assessment, occupational risk analysis, rehabilitation monitoring, and human activity recognition. In general, the proposed hybrid architecture increases the robustness and reliability of ergonomic risk prediction by combining the advantages of RNN and LSTM networks, while boosting the performance of the hybrid network on sequential sensor-based datasets.

HARDWARE DEVELOPMENT

The work done includes several significant steps towards developing a composite deep RNN and LSTM model for dynamic CTS analysis. This starts with the generation of a dataset, where one builds a particular detection device and defines pin settings. Primary and secondary units of the data acquisition and analysis system that has been proposed include a data acquisition unit and a data analysis unit. These components are meant for ergonomic data collection and processing. The proposed hybrid model comprises RNNs and LSTM networks to enable the model to capture proportional relationships in the data and the long-term dependencies that are embraced by LSTMs. There are several steps in the methodology: formation of data sets, preparation of the data sets, extraction of features, and training of models. These lead to the evaluation and analysis of the model in terms of the relevance of patterns discovered for CTS. This comprehensive strategy allows the development of a solid predictive model as a basis for knowledge of and protection against risk factors associated with CTS. The creation of datasets is a vital step in the process of employing an innovative deep RNN and LSTM model for dynamic investigation of carpal tunnel syndrome. This involves the building of a special detecting device and aligning it to get the right data acquisition. The subsequent techniques are vital in the creation of datasets.

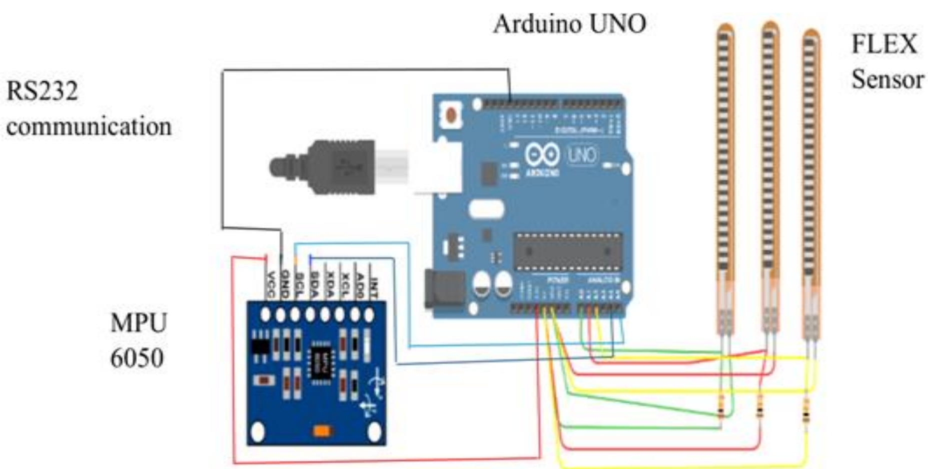

Figure 4: Pin Configuration of Hardware

Pin Configuration

The detection system used in carpal tunnel syndrome consists of an Arduino UNO as the main board together with an MPU6050 sensor and flex sensors to detect motion and bends. The bigger side of the flex sensor is joined with one end of the 10K resistor, and the VCC (+5V) of the Arduino UNO is connected to the V_{CC} of the MPU6050. The parallel port connections of Arduino, MPU6050, and the narrow side of the flex sensor are set up in the same way. Here the SCL pin and SDA of the MPU6050 are connected to A5 and A4 of the Arduino, having a module to transfer data in the I2C protocol. The flex sensors are linked with Arduino's A0, A1, and A2 pins as separate analog pins. This setup is described in Figure 4.

Proposed System

The system's architecture consists of two key components: the data analysis unit and the data collection unit. These components collaborate to identify and examine ergonomic tendencies that are indicative of carpal tunnel syndrome. The proposed system and its hardware configuration are depicted in the block diagram shown in Figure 5. The following studies are accessible for users to explore and delve into further details.

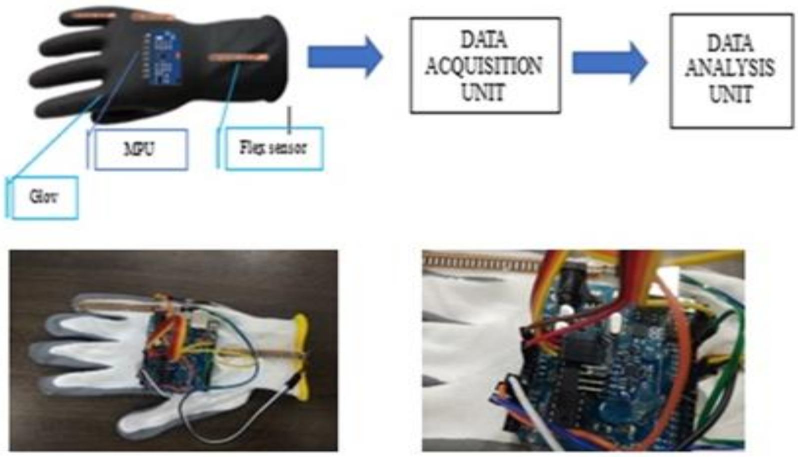

Figure 5: Practical Design of the Hardware.

Data Acquisition Unit

This is the entire flow of an Arduino UNO-based data acquisition and pre-processing system, which can collect motion information and bending information from multiple sensors, as shown in Figure 6. This system comprises an MPU6050 inertial measurement unit (IMU) and three flex sensors that will capture, process, and store data from the flex sensors to be analysed later.

Data acquisition starts with two kinds of sensors: MPU6050 (Accelerometer and Gyroscope): MPU6050 is used to measure three-axis acceleration and angular velocity. It is connected to the Arduino UNO using the I²C interface on pins A4 (SDA) and A5 (SCL). Three Units Flex Sensors: Three flex sensors are attached to the analog input pins on the Arduino (A0, A1, and A2). These sensors generate analog voltages that vary with the amount of bend or flex and can be used to measure the bending angle.

Once the raw sensor data is received, the Arduino will process it in the following manner: Real-time Data Reception to Arduino: The Arduino receives the real-time data from the analog input (flex sensor) and also from the accelerometer and gyroscope (MPU6050) through the I²C connection. Sensor Initialization and Configuration: The MPU6050 sensor is initialized, and the required Degrees Per Second (DPS) range is set to get accurate measurements from the gyroscope sensor, depending on the application.

Arduino Data Acquisition: The Arduino program repeatedly reads the acceleration, angular velocity, and flex sensor readings from the interfaces.

Data Separation and Standardization: The values gathered from the sensors are split into accelerometer, gyroscope, and flex sensor values. Standardization allows the sensor output to have a consistent format so that it can be processed the same way.

Data Normalization: The values collected from the sensors are normalized to a common scale of numbers, which minimizes the impact of variations in measurement units and values. This is a pre-processing step that enhances the quality and consistency of the data.

Data Storage and Transmission: The normalized data is stored, viewed or sent to the Data Storage/Analysis Unit for further processing.

Data Analysis and Applications: The post-processing of the acquired sensor data can be used for plot graphing, machine learning, pattern recognition, gesture recognition, rehabilitation monitoring, motion tracking, and human activity analysis.

Note: DPS (Degrees Per Second) is the measurement range of the angular velocity gyroscope that is set for the MPU6050 sensor.

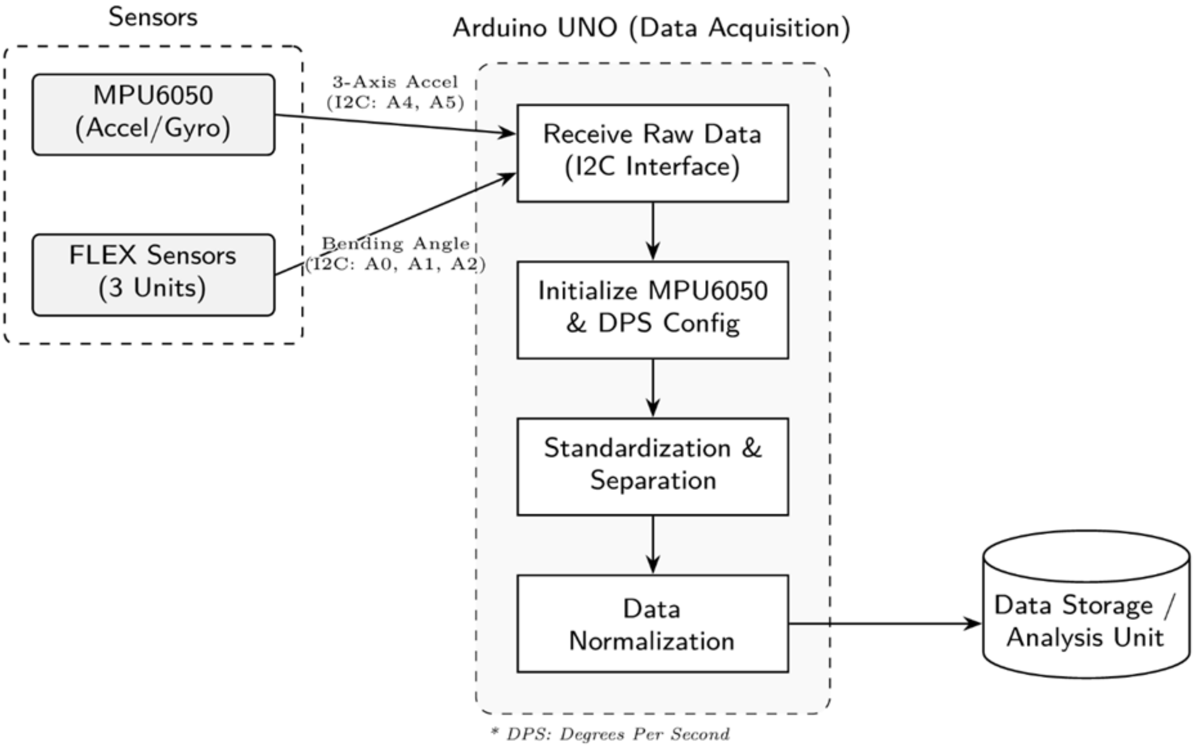

Figure 6. Block Diagram of the Arduino-Based Sensor Data Acquisition and Preprocessing

Data Analysis Unit

The Data Analysis Unit continuously receives information from the DAQ system to determine the presence of movement or bending of the hand in the palm. The two separate components that make up this entire system are the training unit and the testing unit. Datasets, including vibration signatures acquired during common tasks, such as using a mouse, make up the training unit. The training unit also contains vibration signature samples recorded during a variety of hand actions. Send the information to the central processing unit. The goal of data normalization is to make the data more efficient by organizing and transforming it to eliminate duplication and maximize organization. The three FLEX sensors and the MPU 6050 communicate with the Arduino Uno over the I2C protocol. Data in raw form has been received via the Arduino pins A_0, A_1, A_2, A_4, etc. Now, the process flow is running to acquire data in real time. The main purpose of the training unit is to collect data through data gathering and to analyze palm motion and flexion by comparing the collected data with the existing dataset. Figure 7 shows the process flow diagram of the data acquisition unit.

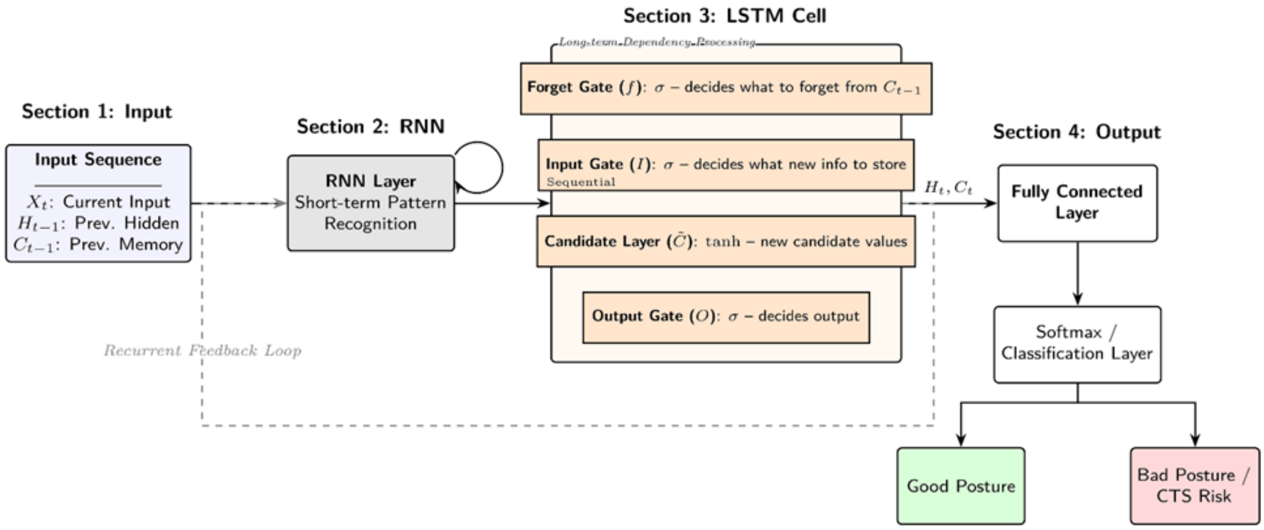

Figure 7: Data Analysis (AI/ML) Unit Process Flow.

SIMULATION

Sensors specifically designed to record posture-related data are used to create the dataset for this study, as shown in Table 1. To efficiently monitor posture, these sensors are positioned at key locations on the hands, such as the wrists, fingers (to measure bending), and areas for detecting ulnar deviation. Real-time data are collected during various activities, such as typing and using a mouse.

Postures are labeled as 'good' or 'bad' based on predetermined ergonomic standards. Data preprocessing involves removing incomplete records, addressing missing values through interpolation, standardizing sensor readings for consistency, and extracting features that directly influence posture analysis to enhance model performance.

To ensure accurate assessment and generalizability of the model, the processed CSV dataset is divided into training, validation, and testing.

Table 1: Sample data received from sensor. [16]

Time	CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8	CH9
51:41.8	-1604	-712	17020	-38	612	-189	1023	594	607

Simulation parameters are the framework upon which the performance and behavior of the posture detection model are assessed. The sensors that provide the dataset are posture detection sensors and in CSV format: Product Key, Date, Time, Posture Client-Side Accelerometer X, Posture Client-Side Accelerometer Y, Posture Client-Side accelerometer, Posture Client-Side gyroscope, Posture Client-Side median frequency, Posture Client-Side position angle, Posture Server-Side Accelerometer X, Posture Server accelerometer Y, Posture Server accelerometer Z, Posture Server gyroscope

The acquisition rate refers to the number of samples of the sensor data that are taken in order to permit a depiction of the posture dynamics. In feature selection, relevant traits from sensor data are obtained, which include angles and motion patterns, to enhance the precision of the model. Also, model hyperparameters such as the learning rate, batch size, number of epochs, and the optimizer, which is allowed to be either Adam or SGD, are well selected and fine-tuned depending on the model in a bid to ensure the best number is obtained during the training process.

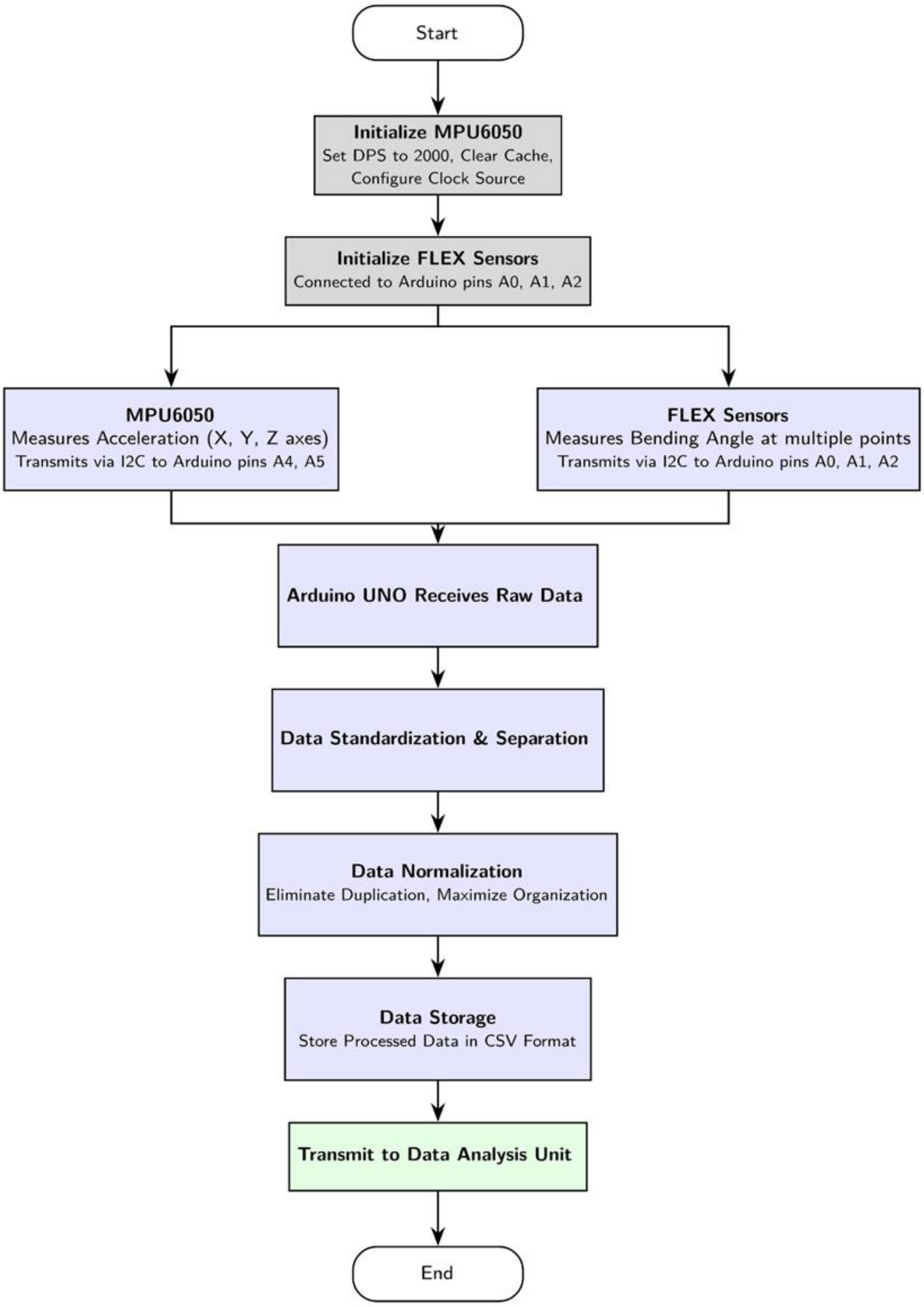

Figure 8: Flowchart of the Sensor Data Acquisition, Preprocessing, and Storage Process Using Arduino UNO

The proposed Arduino UNO-based sensor data acquisition and pre-processing system is shown in Figure 8. The process starts with the initialization of the sensing devices, data acquisition, preprocessing, storing, and transmitting of data to the data analysis unit.

The first step of the workflow is to initialize the MPU6050 sensor, which includes setting the gyroscope to a range of 2000 Degrees Per Second (DPS). The sensor cache is flushed during initialization, and the clock source is set up to guarantee stable and accurate operation. Then the three flex sensors attached to analog pins (A0, A1, and A2) of the Arduino UNO are set up for angle bending detection.

After initialization, both the sensing modules run in parallel. The MPU6050 has three-axis acceleration sensors which are used to measure the acceleration on the X, Y, and Z axes and send this data to the Arduino UNO via the I2C communication protocol via the I2C SDA (A4) and I2C SCL (A5) pins. Meanwhile, the bending angles at several points are measured by the flex sensors, and the analog signals are fed to the Arduino on pins A0, A1, and A2.

The raw data from both sensing modules is then fed into the Arduino UNO, which processes it and separates it into different data streams in a uniform format: the acceleration and bending data are stored in different data streams in the same format. The data are then processed and undergo data normalization to eliminate duplicate or redundant data, ensure data consistency, and normalize the data measurements into a range appropriate for further analysis.

The sensor data is then normalized and saved in a CSV (Comma-Separated Values) file, which is a convenient and easily transportable format for later analysis and machine learning. Finally, the processed data is sent to the data analysis unit, which can be used for motion analysis, gesture recognition, rehabilitation monitoring, or other intelligent decision-making applications. Once data has been successfully transferred, the workflow is finished, and one acquisition cycle has occurred.

Note: DPS (Degrees Per Second) represents the measurement range of the angular velocity of the MPU6050 sensor configured in the gyroscope. The entire workflow guarantees synchronous acquisition, preprocessing, storage, and transmission of multimodal sensor data, enhancing reliability and usability of the gathered data set.

Simulation Flow

Figure 9 below shows the flow chart, which shows a way of performing a hybrid/mixed deep learning model using both RNN and LSTM networks, especially to recognize patterns related to CTS.

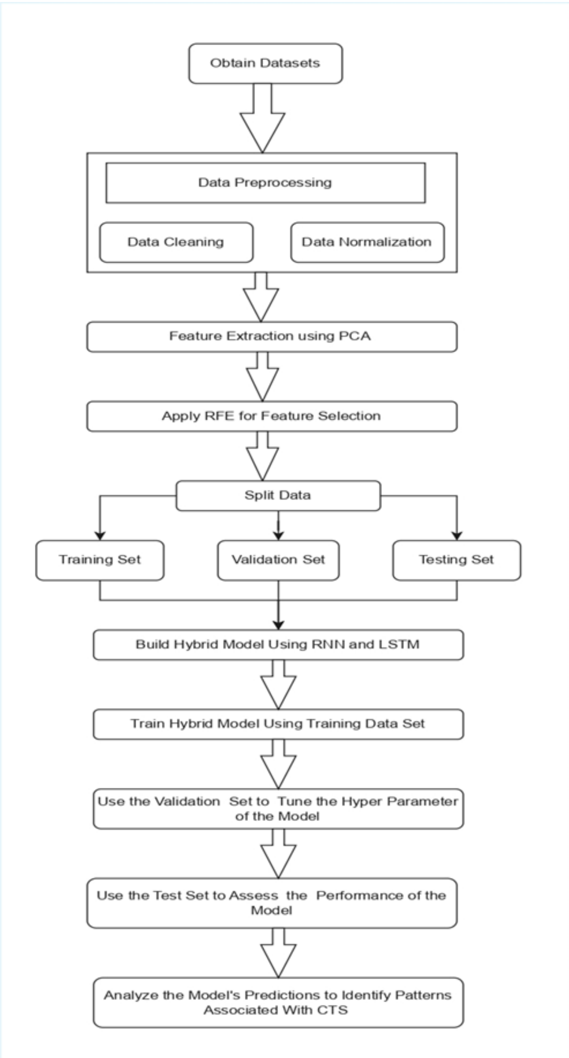

Figure 9: Proposed Methodology

RESULT AND DISCUSSION

The confusion matrix indicates perfect classification without misclassification among the four classes. However, 100% accuracy on training and validation suggests potential overfitting and requires further testing on unseen data. The training and validation accuracy graphs indicate that the model achieved high accuracy, near 100%. Overall, the results demonstrate strong model performance and warrant further evaluation on unseen data.

Model Performance Analysis

Table 2: The model handles sequential data, and due to its temporal feature learning capacity, it is able to identify posture patterns. All the input sequences obtained by reading sensors from CSV datasets represent the sequences of time steps of accelerometer, gyroscope, or other posture information.

Table 2: Comparison of RNN and LSTM for posture detection [17]

Parameter	RNN	LSTM
Input Data Format	Sequential data from CSV (time-series)	Sequential data from CSV (time-series)
Architecture	Simple recurrent layers with feedback	Long Short-Term Memory units with gating mechanisms to handle long-term dependencies
Handling Temporal Data	Struggles with long-term dependencies due to vanishing gradients	Excels at learning long-term dependencies using memory cells and gates
Training Time	Faster due to simpler architecture	Slower due to complex computations in LSTM cells
Memory Usage	Lower	Higher due to additional parameters (forget, input, output gates)
Feature Extraction	Limited capability for complex patterns	Better for extracting temporal patterns from sensor data
Accuracy	Moderate (depends on sequence length)	Higher for long sequences due to better context retention
Loss Convergence	Slower and prone to oscillations	Faster and more stable due to efficient gradient flow
Suitability for Posture Detection	Effective for short-term patterns	Superior for capturing complex and long-term sensor patterns
Evaluation Metrics	Lower precision and recall	Higher precision, recall, and F1-score
Best Use Cases	Simple posture sequences	Complex and multi-activity posture detection

Computational Analysis

The computational analysis entails training an LSTM or RNN on data that is preprocessed and then hyperparameter optimization, such as the number of LSTM units, learning rate, batch size, and epochs. To determine the efficiency of a model in recognizing bad or good posture, accuracy, precision, recall, and F1 score are used to determine model performance. Also, the proposed models are evaluated with the time of training and memory used to process large sequential datasets [17]. Validation is performed on the test data, and performance visualizations are used, which include confusion matrices and loss/accuracy plots, to obtain performance trends and gain information about the model. Table 3 shows the specifics of the hybrid model layers, including their outputs and specifications of the parameters.

The model has a total of 63,566 parameters (248.31 KB) (21,188 trainable and 0 non-trainable). The number of parameters included in the optimizer parameters is 42,378 (165.54 KB). First, the model was 100 percent accurate without dropout layers, and this brought about overfitting concerns. To solve this problem and improve generalization, dropout layers were implemented, which served to solve overfitting to a large extent. The last model structure is presented in Table 3.

Table 3: Model summary report

Layer (type)	Output Shape	Parameters
(LSTM)	(None, 9, 64)	16,896
(Dropout)	(None, 9, 64)	0
(SimpleRNN)	(None, 32)	3,104
(Dropout)	(None, 32)	0
(Dense)	(None, 32)	1,056
(Dropout)	(None, 32)	0
(Dense)	(None, 4)	132

Figure 10 represents the confusion matrix, which demonstrates perfect classification performance, with no misclassifications among the four classes. Figure 11 shows the training and validation accuracy graphs, indicating that the model achieved near-perfect accuracy, close to 100%. However, this may suggest potential overfitting to the training data. Overall, the results reflect strong model performance, warranting further evaluation on unseen data. Figure 12 shows the training and validation loss graphs; on the vertical axis, which represents the loss values, the number of epochs is illustrated on the horizontal axis. The blue color is used to indicate the training loss, and the red color represents the validation loss. The various loss values for this model lie between 0 and 1.5. By adding the dropout layers for regularization, as for the training loss, it remains almost equal to the value of approximately 0.2, whereas the validation loss remains nearly unchanged at a score of around 0.5.

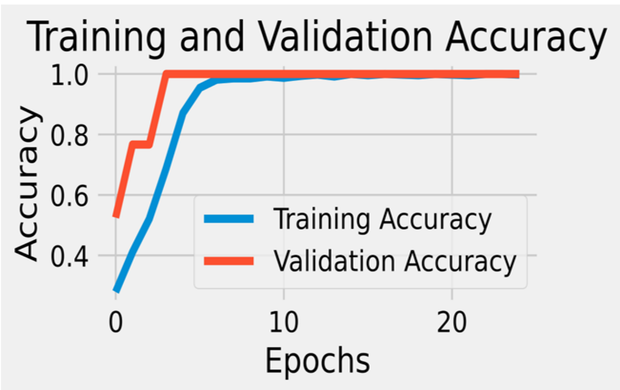

Figure 10: Training & Test Accuracy

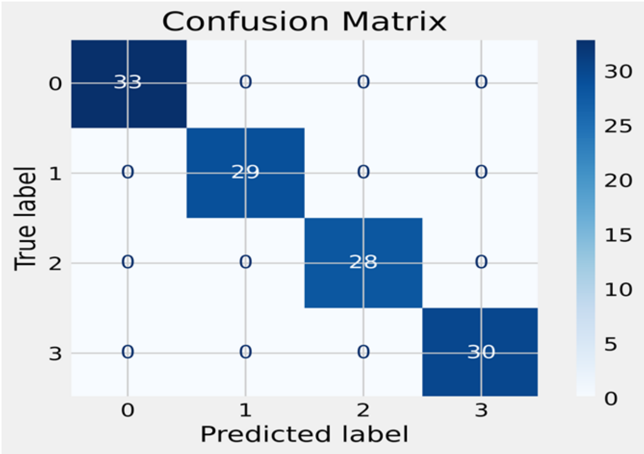

Figure 11: Confusion Metrics

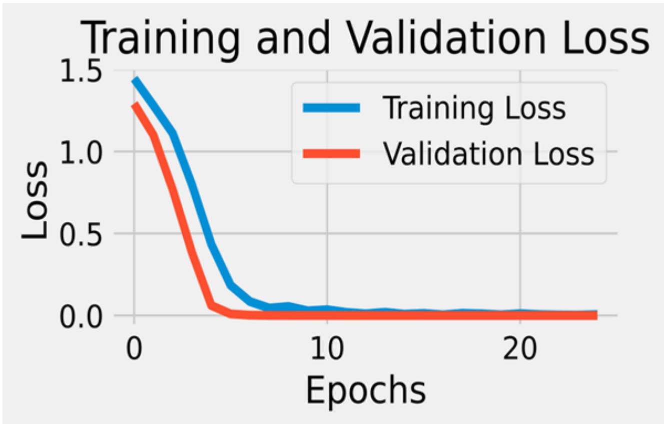

Figure 12: Testing & Validation Loss

CONCLUSION

With the fast-developing sphere of human activity recognition, the growing focus is on inquiries into the health problems that come with awkward postures and repetitive stress. This paper will present an innovative deep learning model that combines Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) to analyze the ergonomics of Carpal Tunnel Syndrome (CTS) and diagnose it at an early stage. The RNN component is effective at reflecting the contextual dependencies in the short term, whereas the LSTM component is able to retain the long-term temporal relationships and thus identify the wrong hand postures. Sensors are used to acquire sensor-based data, which is preprocessed and whose feature dimensionality is reduced with Principal Component Analysis (PCA) and Recursive Feature Elimination (RFE). The model will be trained and optimized to be as accurate and efficient as possible with minimal noise and false alarms. The combined method supports the timely diagnosis of CTS, increases the accuracy of diagnosis, and helps create a reference program to prevent such issues in the future, focusing on enhancing the general musculoskeletal condition.

ACKNOWLEDGEMENT

None.

FUNDING

The authors affirm that they did not receive any financial support, grants, or other forms of assistance during the preparation of this manuscript.

CONFLICT OF INTEREST/COMPETING INTERESTS

The authors declare that there is no conflict of interest.

AUTHOR CONTRIBUTIONS

Nikita Gautam conceptualized and authored the research paper under the supervision of Dr. Sunil Kumar Gupta and Dr. Amit Shrivastava, who provided continuous academic guidance and domain expertise throughout the study. Dr. Gupta and Dr. Shrivastava also contributed to the methodological rigor by assisting in the selection of appropriate analytical techniques and ensuring the accuracy of data processing and interpretation. All authors reviewed and approved the final manuscript.

DATA AVAILABILITY STATEMENT

Data cannot be made publicly available upon publication because they are being used in ongoing research. Data supporting the findings of this study are available upon reasonable request from the authors.

ETHICS APPROVAL

This study was conducted in accordance with ethical research standards. Informed consent was obtained from all individuals who participated in the data collection process, which involved capturing posture angles associated with both correct and incorrect hand positions. Participants were briefed on the purpose of the study and voluntarily contributed ergonomic data relevant to the analysis of Carpal Tunnel Syndrome (CTS) risk factors.

ABBREVIATION

CTS- Carpal Tunnel Syndrome

MN- Median Nerve

RNN- Recurrent Neural Network

LSTM- Long-Short Term Memory

DL- Deep Learning

PCA-Principal Component Analysis

RFE-Recursive Feature Elimination

ORCID iD

Nikita Gautam : https:/orcid.org/0000-0002-3265-0636

Sunil Kumar Gupta: https:/orcid.org/0000-0002-8510-1032

Amit Shrivastava: https:/orcid.org/0000-0002-9141-7199

REFERENCES

[1] Urits I, Gress K, Charipova K, Orhurhu V, Kaye AD, Viswanath O. Recent advances in the understanding and management of carpal tunnel syndrome: a comprehensive review. Current pain and headache reports. 2019 Oct;23:1-8.

[2] Gadkari PD, Dahikar GD, Ganjiwale RO. A review on carpal tunnel syndrome. Research Journal of Pharmacy and Technology. 2020;13(10):4961-5.

[3] Hosseini-Farid M, Schrier VJ, Starlinger J, Zhao C, Amadio PC. Carpal tunnel syndrome treatment and the subsequent alterations in tendon and connective tissue dynamics. Clinical Biomechanics. 2021 Aug 1;88:105440.

[4] Kondaveeti HK, Kumaravelu NK, Vanambathina SD, Mathe SE, Vappangi S. A systematic literature review on prototyping with Arduino: Applications, challenges, advantages, and limitations. Computer Science Review. 2021 May 1;40:100364.

[5] Rahman MA, Hossain MF, Hossain M, Ahmmed R. Employing PCA and t-statistical approach for feature extraction and classification of emotion from multichannel EEG signal. Egyptian Informatics Journal. 2020 Mar 1;21(1):23-35.

[6] Ramezan CA. Transferability of recursive feature elimination (RFE)-Derived feature sets for support Vector Machine Land Cover classification. Remote Sensing. 2022 Dec 8;14(24):6218.

[7] Tanaka S, Inui A, Mifune Y, Nishimoto H, Shinohara I, Furukawa T, Kato T, Kusunose M, Ehara Y, Takigami S, Kuroda R. Dynamic analysis of the median nerve in carpal tunnel syndrome from ultrasound images using the YOLOv5 object detection model. Applied Sciences. 2023 Dec 14;13(24):13256.

[8] Feng W, Zhang W, Meng M, Gong Y, Gu F. A novel binary classification algorithm for carpal tunnel syndrome detection using LSTM. In2023 IEEE 3rd International Conference on Software Engineering and Artificial Intelligence (SEAI) 2023 Jun 16 (pp. 143-147). IEEE.

[9] Nowak W, Znamirowska P, Szmigielska N, Zemsta K, Miśkiewicz J, Plata H, Pałatyńska M, Kulesza B. Risk factors for carpal tunnel syndrome. Journal of Pre-Clinical and Clinical Research. 2023.

[10] Labrín C, Urdinez F. Principal component analysis. InR for political data science 2020 Nov 17 (pp. 375-393). Chapman and Hall/CRC.

[11] Faysal JA, Mostafa ST, Tamanna JS, Mumenin KM, Arifin MM, Awal MA, Shome A, Mostafa SS. XGB-RF: A hybrid machine learning approach for IoT intrusion detection. InTelecom 2022 Jan 4 (Vol. 3, No. 1, pp. 52-69). MDPI.

[12] Sharma A, Bhuriya D, Singh U. Survey of stock market prediction using machine learning approach. In2017 International conference of electronics, communication and aerospace technology (ICECA) 2017 Apr 20 (Vol. 2, pp. 506-509). IEEE.

[13] Schmidt RM. Recurrent neural networks (rnns): A gentle introduction and overview. arXiv preprint arXiv:1912.05911. 2019 Nov 23.

[14] Smys S, Chen JI, Shakya S. Survey on neural network architectures with deep learning. Journal of Soft Computing Paradigm (JSCP). 2020 Jul 30;2(03):186-94.

[15] Han YK, Choi YB. Detection of emergency disaster using human action recognition based on LSTM model. IEIE Transactions on Smart Processing & Computing. 2020 Jun;9(3):177-84.

[16] Gautam N, Gupta SK, Shrivastava A. Leveraging wearable sensors and machine learning for posture-based detection of carpal tunnel syndrome. Engineering Research Express. 2024 Oct 15;6(4):045213.

[17] Elseddik M, Alnowaiser K, Mostafa RR, Elashry A, El-Rashidy N, Elgamal S, Aboelfetouh A, El-Bakry H. Deep learning-based approaches for enhanced diagnosis and comprehensive understanding of carpal tunnel syndrome. Diagnostics. 2023 Oct 14;13(20):3211.