Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

ARTICLE TYPE

Deep Learning-Based Gesture Recognition for Surgical Applications: A Data Augmentation Approach

Sofía Sorbet Santiago[1] | Jenny Alexandra Cifuentes*[2,3]

[1]Department of Statistics, Universidad Carlos III de Madrid, Calle Madrid, 126, Getafe, Spain
[2]Department of Quantitative Methods, Faculty of Economics and Business Administration, ICADE, Comillas Pontifical University, Calle de Alberto Aguilera, 23, Madrid, Spain
[3]Institute for Research in Technology (IIT), ICAI School of Engineering, Comillas Pontifical University, Calle de Santa Cruz de Marcenado, 26, Madrid, Spain

Correspondence
*Corresponding author name, This is sample corresponding address. Email: jacifuentes@comillas.edu

Summary

Hand gesture recognition and classification play a pivotal role in automating Human-Computer Interaction (HCI) and have garnered substantial attention in research. In this study, the focus is placed on the application of gesture recognition in surgical settings to provide valuable feedback during medical training. A tool gesture classification system based on Deep Learning (DL) techniques is proposed, specifically employing a Long Short Term Memory (LSTM)-based model with an attention mechanism. The research is structured in three key stages: data pre-processing to eliminate outliers and smooth trajectories, addressing noise from surgical instrument data acquisition; data augmentation to overcome data scarcity by generating new trajectories through controlled spatial transformations; and the implementation and evaluation of the DL-based classification strategy. The dataset used includes recordings from ten participants with varying surgical experience, covering three types of trajectories and involving both right and left arms. The proposed classifier, combined with the data augmentation strategy, is assessed for its effectiveness in classifying all acquired gestures. The performance of the proposed model is evaluated against other DL-based methodologies commonly employed in surgical gesture classification. The results indicate that the proposed approach outperforms these benchmark methods, achieving higher classification accuracy and robustness in distinguishing diverse surgical gestures.

KEYWORDS:
Gesture Classification; Attention-based LSTM Neural Networks; Data Augmentation; Surgical Gestures ; Deep Learning

1 | INTRODUCTION

Minimally Invasive Surgery (MIS) has been a significant area of scientific research since the early 19th century, aiming to develop methods and provide solutions for surgical procedures with minimal incisions. MIS enables access to the patient's abdominal cavity or pelvis while reducing the need for large incisions and minimizing damage to surrounding tissues. This approach has demonstrated superior outcomes and shorter recovery times compared to traditional surgery [1]. However, the transition to laparoscopic surgery presents challenges in terms of knowledge transfer and the acquisition of specific psycho-motor skills, including depth perception, hand-screen coordination, and the loss of touch perception [2]. Traditionally, these skills were acquired through the use of human corpses and animal models. However, simulation-based approaches that replicate surgical procedures have gained popularity as effective training tools. These alternatives not only facilitate skill acquisition but also provide metrics for performance assessment, learning curve measurement, and operation monitoring. Over the past few decades, researchers have developed numerous objective metrics to classify training surgical trajectories and provide valuable feedback to trainees [3]. These metrics aim to differentiate gestures based on their geometric and kinematic features, as well as the expertise of the individuals performing the movements. This research field, known

as surgical gesture recognition, continues to evolve due to technical challenges associated with spatial complexity, repeatability, temporal variance, and changes in movement orientation.

Gesture recognition has found widespread application in Human-Computer Interaction (HCI), where interfaces enable intuitive and efficient interaction with virtual environments. HCI interfaces have proven particularly valuable in surgical training, enabling participants to simulate scenarios and interact with surgical environments[4]. Various tracking instruments, such as video cameras, accelerometers, electromagnetic devices, and sensors, are commonly employed to capture gesture trajectories in simulation environments[5]. These trajectories are typically represented as sequences of centroids of the rigid body, obtained by computing the centroidal position of the hand or instrument over time[6]. Consequently, dynamic-based analysis approaches are necessary to handle these trajectory sequences. However, dynamic gesture recognition using sensors poses several challenges that can impact algorithm performance. Studies have revealed that combining trajectories with different dynamic behaviors can reduce the accuracy of recognition algorithms[7]. While several Machine Learning (ML) based methods have been proposed to enhance gesture recognition, they often struggle with the dynamic and unstructured nature of surgical gestures, which can vary greatly in speed and complexity. Traditional models may not effectively handle the temporal dependencies and nuances of these movements. Moreover, conventional ML-based models often require substantial computational resources and may not generalize well across different surgical scenarios or gesture types without extensive training on large, varied datasets[8].

The proposed approach leverages the power of Long Short Term Memory (LSTM) models enhanced with attention mechanisms. Unlike traditional ML models, and within the context of Deep Learning (DL), LSTMs are adept at capturing long-term dependencies in time-series data, making them particularly suited for the temporal variability in surgical gesture data. The addition of attention mechanisms allows the model to focus dynamically on the most relevant segments of the data sequence, enhancing the model's ability to discern subtle differences in gesture execution that are required for accurate classification. This capability addresses the common drawbacks of insufficient sensitivity to the context and sequence structure that often hinder many existing DL approaches. To address the challenge of limited available data in this domain, an augmentation strategy is introduced, which generates new trajectories by applying controlled spatial transformations such as scaling, rotation, and translation to the existing ones. This data augmentation not only expands the diversity and size of the dataset but also helps to capture spatial dependencies of hand gestures, enhancing the learning process and improving the model's generalization ability. Finally, the performance of the LSTM-based model including an attention mechanism is evaluated, and compared to other DL-based strategies for surgical gesture classification. The primary objective is to assess the ability of the algorithm to extract sufficient features for accurate gesture classification by analyzing the coordinates of the laparoscopic instrument's rigid body over time.

This paper is structured as follows. In Section 2, an extensive review of the existing literature in the field of gesture recognition is provided, discussing the latest advancements and relevant studies. Section 3 describes the fundamentals of the base models proposed in this study. Section 4 presents the comprehensive methodology proposed in this study, outlining the data pre-processing techniques, the data augmentation strategy, and the implementation of a DL-based classification model utilizing an LSTM architecture with an attention mechanism. In Section 5, the experimental results obtained from the conducted trials are presented and analyzed in detail, including performance metrics and comparisons with baseline methods. Finally, Section 6 summarizes the main conclusions drawn from the research, highlighting the contributions made to the field of gesture recognition in surgical contexts and the efficacy of the proposed approach in accurately classifying diverse surgical gestures.

2 | LITERATURE REVIEW

Numerous studies have employed a diverse range of mathematical and computational techniques to tackle gesture classification. Despite these efforts, the field continues to be an active area of research due to the aforementioned challenges. Initially, researchers proposed basic measures based on global features to quickly and easily evaluate performance and classify surgical gestures[9]. This approach represents one of the most prevalent types of hand gesture classification techniques. The underlying concept behind these algorithms is to compute global quantities that enable the analysis of surgical skills. It is logical to assume that factors such as the length or duration of the instrument's path, the smoothness of the trajectory, or the total number of errors are related to the expertise level of the performer. Although these measures are straightforward to calculate and analyze, they do not produce high accuracy values and provide limited information and feedback regarding movement execution.

In contrast to global methods, there exist strategies that focus on analyzing gestures locally[10]. This approach involves decomposing the overall trajectory into smaller segments and independently analyzing these segments to create a qualitative description of the movement[8]. Local analysis techniques have been extensively explored in the context of surgical skill assessment[11]. Unlike global-based metrics, local metrics enable comparisons between experts and novices and across different types of gestures[12]. One of the key advantages of this approach is its suitability for classifying 3D movements[13] and incorporating the temporal dimension[12], leading to significant classification results. Various examples of local metrics include gradient direction[14], arc length[15], affine velocity[16], and signal decomposition[17]. Among the local algorithms widely used in the literature are Hidden Markov Models (HMM)[18], Dynamic Time Warping (DTW)[19], Longest Common Sub-Sequence (LCSS)[20], and Support Vector

Machines (SVM)[21]. However, these metrics can be computationally expensive and challenging to calculate[22], prompting the exploration of alternative methods to address these technical issues. Exploring further alternatives, genetic algorithms have been employed for gesture recognition, as demonstrated in the studies proposed in[23] and[24]. In[23], Kaluri et al. developed a system that employed a median filter for noise removal, followed by segmentation using the Modified Region Growing Algorithm (MRGA). The feature extraction and recognition phases were handled by an Adaptive Genetic Fuzzy Classifier (AGFC). This approach was innovative in integrating genetic algorithms with fuzzy logic to optimize the rule set generated by the classifier, enhancing the system's accuracy in distinguishing different sign language gestures. In[24], the study built on the previous work but introduced several refinements to enhance the robustness and accuracy of the gesture recognition process. This work employed an adaptive filter for more effective noise reduction and used a region growing algorithm for segmentation. Additionally, this framework was assessed against a SVM classifier, providing a comparative analysis that highlighted the strengths of genetic algorithms in handling complex gesture recognition tasks. These studies showcased the potential of genetic algorithms in optimizing the feature extraction process, enhancing the overall classification accuracy. However, despite their utility, genetic algorithms generally require significant computational resources and may not achieve the optimal solution efficiently, which can limit their practicality for surgical applications in real-time[25].

In this context, Artificial Neural Networks (ANNs) have achieved remarkable success in addressing the challenge of gesture classification, consistently demonstrating high accuracy rates in multi-level classification studies[26]. In a recent systematic review by Yanik et al.[27], a comprehensive overview of DL-based strategies for assessing surgical skills highlighted the numerous advantages offered by the implementation of ANNs. Firstly, ANNs possess powerful self-learning capabilities, eliminating the need for explicit programming of feature extraction algorithms. This inherent capability enables ANNs to automatically extract relevant features from input signals, relieving researchers from the burden of manual feature engineering. Furthermore, ANNs exhibit remarkable flexibility, allowing them to adapt and maintain robust performance even in the presence of slight variations in the input data. Their ability to capture intricate interactions and patterns positions them as one of the most potent techniques for analyzing gesture data, uncovering hidden information that may elude traditional methods. Notably, ANNs offer significant computational efficiency gains, substantially reducing the time required for classification tasks once the training process is completed. This efficiency makes ANNs highly practical and suitable for real-time gesture recognition applications[26]. In terms of architectures, while MultiLayer Perceptron Neural Networks (MLPNN) have been widely implemented, Recurrent Neural Networks (RNNs) are often preferred for modeling dynamic gestures due to their inherent advantages, particularly in capturing temporal dependencies. In this research domain, a notable study conducted by Bailador et al.[28] analyzed a database comprising 320 instances classified into eight distinct surgical gestures. The results demonstrated that when specific constraints were imposed on the dataset, the success rates on the training set reached an impressive 94%. Conversely, in the research developed by Mazomenos et al.[29], eight different participants performed three distinct gestures, which were classified using a single RNN-based architecture. The accuracy achieved was approximately 60%. These findings suggest that RNNs require a substantial number of training examples to achieve high accuracy rates[6]. Additionally, it is worth noting that RNNs exhibit less favorable performance metrics when dealing with long-duration temporal dependencies, as they are prone to the vanishing and exploding gradient problems during training. This inherent limitation can significantly degrade the accuracy of the models when processing longer input sequences, making them less effective for tasks that require capturing extensive temporal information[28].

To address the challenges associated with RNNs in dynamic gesture classification, the Long Short Term Memory (LSTM) models were developed and have emerged as one of the preferred DL architectures. In a study developed by Cifuentes et al.[30], an experiment was conducted to compare the performance of LSTM and RNNs in modeling 3D medical gestures captured using a laparoscopic instrument. The dataset included gestures performed by 14 surgeons, consisting of 8 novices and 6 juniors. The results demonstrated that LSTM outperformed RNNs in accuracy, achieving 99.1% and 96% respectively. Similarly, Hasseb & Parasuraman[31] used a device equipped with movement sensors to acquire three different gestures, achieving results of up to 94%. To enhance the discriminative capability of LSTMs, several proposals have been made, including the adoption of Gated Recurrent Unit (GRU), Bidirectional Long Short-Term Memory Recurrent Neural Networks (BiLSTM), and Bidirectional Gated Recurrent Unit (BiGRU). These architectures often provide more abstract and useful representations. However, they still face challenges in capturing patterns when dealing with large intra-class and inter-class variability. The intra-class variability primarily arises from variations in user performance, while the inter-class variability is mainly due to the similarity among different gestures[5]. For instance, Hung et al.[32] tested a GRU architecture to classify two suturing gestures using image tracking. The experiment was conducted using a dataset comprising 122 training samples, 31 validation samples, and 31 test samples, achieving an accuracy of 86.67%. In the case of BiLSTM, this architecture leverages the idea that a gesture at a specific timestep may depend on both past and future context[33]. The research proposals described in[33] and[34] demonstrate the effectiveness of BiLSTM-based approaches.

Using this approach, an insightful investigation conducted by Lefebvre et al.[33] employed a dataset comprising 1540 distinct gestures, which were performed by a group of 22 individuals. This comprehensive dataset encompassed 14 different gestures, each executed with a commendable level of expertise and repeatability, as they were repeated five times. To capture the intricate nuances and subtleties of these gestures, state-of-the-art accelerometer and gyroscope sensors were employed, ensuring the capture of highly detailed and accurate movement data. Building upon this dataset, the BiLSTM-RNN approach was expertly applied, leading to successful outcomes. In fact, the mean classification rate achieved for

a selected subset of 616 test gestures reached a value of 95.57%. While the BiLSTM-RNN approach demonstrated impressive results in gesture classification, Convolutional Neural Networks (CNNs) could offer complementary advantages. Their superior spatial feature extraction capabilities and computational efficiency make them an attractive alternative for further enhancing the accuracy and performance in gesture recognition tasks, especially with complex sensor data [35,36]. In this research line, in [35], a CNN algorithm was employed on two distinct video-based datasets. The first dataset comprised 1239 training, 411 validation, and 431 testing videos, encompassing 83 distinct gestures. The second dataset consisted of 1050 training and 482 test videos, featuring 25 gesture classes. The achieved accuracy values for these datasets were 94.04% and 83.82%, respectively. Moreover, in a notable contribution, Gadekallu et al. implemented a novel crow search-based convolutional neural networks model in gesture recognition within the HCI domain. Utilizing a publicly available hand gesture dataset, the study applied a one-hot encoding technique for data pre-processing, followed by the crow search algorithm to optimize hyper-parameters for CNN training. This method effectively excluded irrelevant parameters, significantly enhancing the classification accuracy of hand gestures. The model reported a high training and testing accuracies of 99.9%, underscoring its superiority over traditional models in HCI scenarios [37]. Additional examples in the literature demonstrate the efficacy of CNN-based approaches, such as [38], where CNN algorithms were utilized to accurately recognize various fingertip positions through image processing, achieving a minimum accuracy of 99.90%. However, the aforementioned approaches encounter challenges when dealing with a large number of diverse gestures. The primary issue is that the accuracy of recognition methods tends to decrease as the number of classification levels increases. This reduction in performance is attributed to both inter- and intra-variability among the gestures. Inter-variability refers to the differences in gesture execution between different individuals, while intra-variability describes the variations in gesture performance across different attempts by the same individual. These variabilities introduce a level of complexity that standard CNN architectures might struggle to manage effectively [39].

To overcome this challenge, researchers have explored the combination of RNNs and CNNs, leading to the development of innovative architectures like Long Short Term Memory - Fully Convolutional Networks (LSTM-FCN). As elucidated by the authors, LSTM facilitates time-dependent learning of complex information, while FCNs enable efficient gesture prediction by extracting abstract spatial features. Notably, in [39], five users performed 50 different types of gestures using a WiFi data glove sensor-based approach, achieving an average accuracy of 98.9% in the testing set. Similarly, other studies, such as [40], demonstrated relative high accuracy (98%) by combining depth and skeleton information to recognize 14 distinct gesture types. While FCNs are efficient in gesture prediction by extracting abstract spatial features, they often lack the capability to capture complex temporal dependencies, which is crucial in gesture classification. On the other hand, TCNs (Temporal Convolutional Networks) combine the feature extraction efficiency of CNNs with a specialized structure for handling temporal sequences, offering a more balanced and robust approach for dynamic gesture recognition. In this context, a notable study proposed in [41] investigated the impact of label granularity on the performance and generalization of robotic activity systems during surgical gesture recognition tasks, specifically examining TCNs. This research made significant strides by comparing the performance of TCNs at various levels of the surgical hierarchy using only kinematic data. As a result, it was also found that models trained on aggregated data from multiple tasks significantly enhanced performance. Particularly, the inclusion of a small portion of target task data in the training set markedly improved the accuracy of surgeme classification models.

Recent advancements in the field have included the proposal of alternative strategies, specifically involving Graph Neural Networks (GNNs), as detailed in [42,43,44]. Notably, [43] introduces an efficient graph convolutional network tailored for dynamic hand gesture recognition. This model stands out for its enhanced spatiotemporal feature learning capabilities, demonstrating competitive performance alongside current state-of-the-art deep learning-based approaches. A key strength of GCNNs, as exemplified in this model, is their graph-based learning framework. This framework is adept at effectively representing and processing relational dependencies among diverse data points or features. Such a capability is particularly beneficial in handling data characterized by complex, graph-like structures. This approach offers a more comprehensive understanding of the data, a significant advancement over the traditional sequential processing methods employed by TCNs. Upon these advances in GCNNs for gesture recognition, the integration of attention mechanisms is proposed to be incorporated, which is anticipated to further enhance the performance of these techniques. By focusing on key features within complex data structures, attention mechanisms enable the model to prioritize and weigh the most relevant information more effectively. This targeted approach, combined with other architectures that model temporal dependencies such as LSTM RNNs, is expected to yield more accurate recognition of dynamic hand gestures.

3 | GESTURE CLASSIFICATION FRAMEWORK USING DEEP LEARNING STRATEGIES

DL encompasses a set of Machine Learning (ML) algorithms that facilitate the modeling of data through nonlinear transformations [45]. DL algorithms have found applications in various domains, including gesture classification. Within this context, this paper focuses on analyzing DL-based architectures that effectively leverage the temporal dependencies present in data sequences. While conventional neural architectures assume independent samples, time series data, on the other hand, exhibit correlations between the data point at time t and its past and future counterparts. Overlooking these relationships would result in the loss of valuable information. Hence, it becomes imperative to incorporate algorithms that process data in a sequential manner. DL-based algorithms offer the advantages of robustness against noise and automatic discovery of nonlinear relationships

within data[46]. Nevertheless, working with time-series data entails a meticulous process, as modeling sequences poses challenges, and even minor fluctuations in trends or seasonality can impact model performance[46]. Among DL-based strategies, RNNs are particularly adept at handling sequential data, making them highly relevant for applications such as gesture classification, where temporal dependencies significantly influence performance. Unlike conventional neural architectures which assume independent data samples, RNNs consider the correlations between consecutive data points in a time series, a critical feature for capturing the nuances of gesture sequences. This consideration prevents the loss of valuable temporal information, ensuring that each data point contributes contextually to the overall pattern recognition. Given their significance in handling temporal dependencies, RNNs will be extensively explored in this section. Additionally, this section will introduce a specialized data augmentation strategy designed for multi-dimensional data, providing a detailed examination of both the theoretical foundations and practical applications of these techniques in gesture classification.

3.1 | Recurrent Neural Networks for Time Series

RNNs evolved as an extension of MLPNNs, providing a significant advantage in handling sequences of varying lengths. Their effectiveness lies in their ability to capture both short-term and long-term dependencies within data. Notably, RNNs have proven to be robust performers, even without extensive pre-processing of the data[47]. In this approach, the network takes d-dimensional data sequences of length n as input, represented as $x_1, x_2, ..., x_n$. The fundamental concept underlying RNNs is that the d-dimensional input at time t, denoted as x_t, should possess knowledge of previous inputs. This is achieved through the utilization of a "hidden state", which serves as a form of local memory, preserving relevant patterns from previous time steps to inform subsequent updates[47]. Consequently, past activations can contribute to modeling the current behavior[48]. Mathematically, the hidden state h at time t is a function f of the sequence value at time t and the hidden state from the preceding time step $(t-1)$:

$$h_t = \begin{bmatrix} 0, & t = 0 \\ \tanh\left(W_{xh}x_t + W_{hh}h_{t-1}\right) & \text{otherwise} \end{bmatrix}, \tag{1}$$

where $\tanh$ represents a non-linear function, such as the sigmoid function. W_{xh} denotes the input-hidden matrix, and W_{hh} represents the hidden-hidden matrix. Additionally, W_{hy} is defined as the hidden-output matrix. Consequently, the output probabilities are learned from the hidden states using the following expression:

$$y_t = W_{hy}h_t \tag{2}$$

The basic architecture of an RNN is depicted in Figure 1, which has demonstrated successful implementation in various applications. However, it has been observed that training becomes challenging when time sequences exhibit long-term dependencies, primarily due to the issue of vanishing gradients[49]. To address this problem, more advanced models such as LSTM and GRU networks were developed[50]. The subsequent section will provide detailed explanations of prominent LSTM architecture, frequently referenced in the literature.

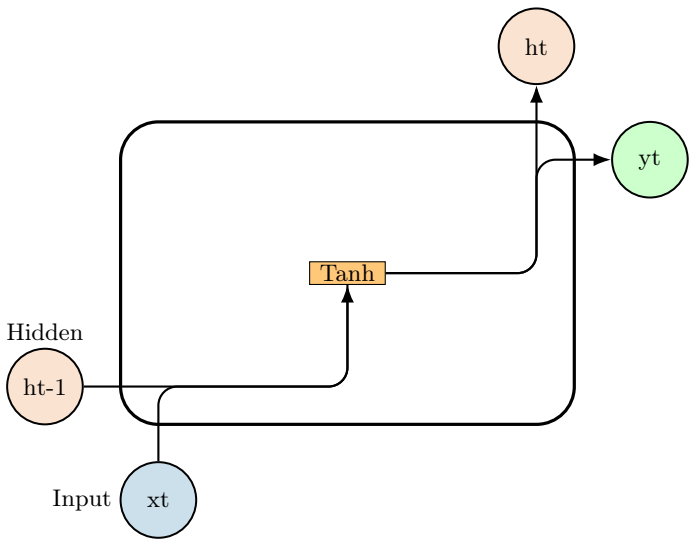


Figure 1 RNN Network Representation

3.2 | Vanilla Long Short Term Memory

LSTM networks were developed as an extension of the basic components of RNNs, specifically to address the vanishing gradient problem and improve the memory capacity for retaining past data [46]. The fundamental LSTM architecture incorporates key components that significantly improve the predictive capabilities of RNNs in time-series analysis. In this approach, the input vector of length n is represented as x_t, and the hidden state of the k-th layer in a multi-layer LSTM is denoted as $h_i^{(k)}$. LSTMs utilize a cell state, denoted as $c_t^{(k)}$, to bolster their ability to store long-term information. This is achieved through various operations on previous cell states, including "forgetting" and "updating" mechanisms, which allow the network to selectively retain relevant information and discard less important data. The architecture of LSTMs includes three main components: the input gate i, the forget gate f, and the output gate o. Each of these gates controls the flow of information into, out of, and within the cell state, respectively. The input gate regulates the amount of new information to be stored in the cell state c, while the forget gate determines which existing information is retained or discarded. The output gate controls the amount of information that is exposed to the next layer of the LSTM or the final prediction. These gating mechanisms endow LSTMs with the ability to learn and adapt over long sequences, making them particularly effective for time-series data. The computation of these intermediate gates can be expressed in equation 3. By employing this memory storage approach, LSTM models effectively mitigate gradient instability issues. This is accomplished by allowing states in different temporal layers to share more similarity through long-term memory, which results in reduced disparity in gradients concerning incoming weights [47].

$$\begin{matrix}\text{Input Gate:}\\ \text{Forget Gate:}\\ \text{Output Gate:}\\ \text{New C.-State:}\end{matrix}\begin{bmatrix} i \\ f \\ o \\ c \end{bmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \tanh \end{pmatrix} W^{(k)} \begin{bmatrix} h_t^{(k-1)} \\ h_{t-1}^{(k)} \end{bmatrix} \tag{3}$$

where sigm represents the sigmoid function, $\tanh$ denotes any non-linear function, and $W^{(k)}$ describes the weight matrix. Once the intermediate values are computed, the cell state or memory unit $c^{(k)}t$ is updated. This involves selectively ignoring a portion of the previous memory $c(t-1)^{(k)}$ with a weight of f and incorporating additional content c with a weight of i. The updates to the long-term memory can be expressed using Equation 4. Simultaneously, the hidden state is also updated using Equation 5, where $\odot$ denotes the element-wise product. For visual reference, Figure 2 illustrates the architecture of a Vanilla LSTM.

$$c_t^{(k)} = f \odot c_{t-1}^{(k)} + i \odot c \tag{4}$$

$$h_t^{(k)} = o \odot \tanh\left(c_t^{(k)}\right) \tag{5}$$

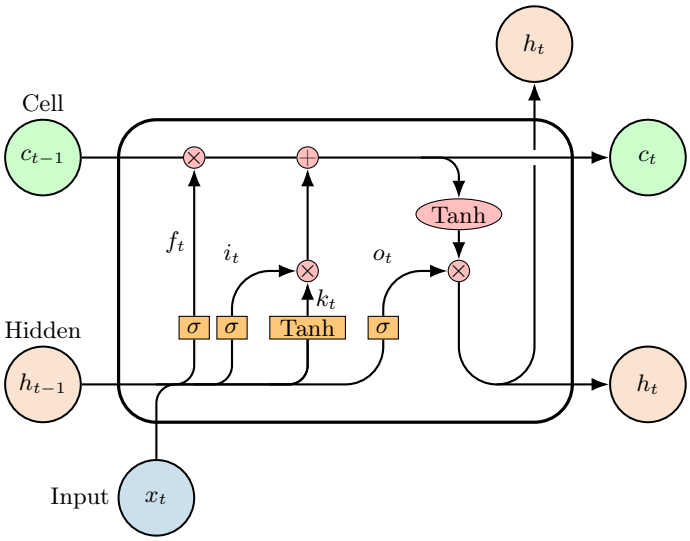


Figure 2 LSTM Network Representation

Within this field of research, recent investigations have put forth several methodologies aimed at augmenting the capabilities of conventional LSTM networks. Among these techniques, the integration of attention mechanisms has emerged as a particularly promising approach, demonstrating notable advancements in both accuracy and efficiency for such networks [51].

3.3 | Attention Mechanisms

The concept of attention mechanism originally emerged in the field of neural machine translation, where its purpose was to automatically translate text from one language to another. In traditional sequence-to-sequence models, the input sentence is encoded into a fixed-length vector and then decoded to generate the output sentence. However, this fixed-length encoding poses challenges when dealing with long input sentences, potentially leading to the loss of relevant information and resulting in inaccurate translations. To address this issue, the attention mechanism enables the decoder to selectively focus on different segments of the input sentence. During the decoding process, the attention mechanism computes a weighted sum of the encoding vectors of the input sentence, where the model learns these weights automatically. By using the attention mechanism, the decoder can effectively pay more attention to the important parts of the input sequence, leading to better translation results.

In the time series classification field, attention mechanisms have demonstrated their effectiveness in capturing intricate dependencies and patterns within the data [51]. Similar to machine translation, time series classification often involves handling lengthy input sequences. In such cases, attention mechanisms can effectively model temporal dependencies by focusing on the most relevant segments of the series at each step of the classification process. Recognizing these advantages, the LSTM model proposed in this study incorporates an attention mechanism. The proposed forecasting approach is depicted in Figure 3, where the LSTM is initially trained on a set of time series with a length of k, enabling the acquisition of the hidden state at each time step.

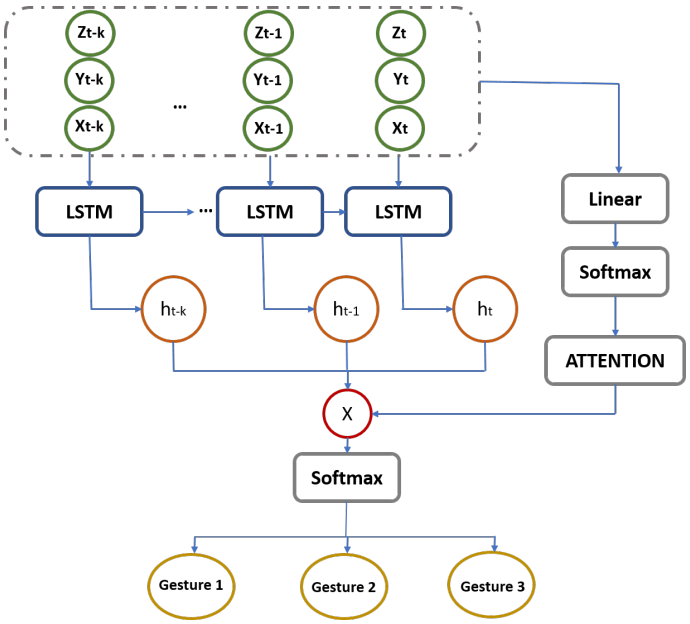


Figure 3 Overview of each stage involved in the proposed classification approach.

3.4 | Data Augmentation Strategies

Overfitting occurs when a model's ability to generalize is compromised, meaning that a high level of accuracy on the training dataset does not necessarily translate to good performance on the testing dataset. This problem becomes more pronounced when the available dataset is small or when dealing with complex relationships. NNs are generally susceptible to overfitting, but there are techniques to mitigate this issue, such as regularization, parameter sharing, and data augmentation, among others [47]. Data augmentation is a strategy employed to artificially expand the training dataset, making it appear as if the new data is generated from the same underlying distribution [52]. However, to generate meaningful data within the specific domain of study and adhere to valid transformations, domain expertise is required. This requires careful design, implementation,

and validation of the transformation set for each new domain, limiting its reusability and generality [53]. Intuitively, data augmentation is valuable for increasing the quantity and diversity of the data and providing the model with knowledge about invariances within the data domain [54]. Data augmentation offers a primary advantage over other methods to address overfitting, as it targets the root of the problem by enriching the training dataset [55]. While data augmentation techniques have been extensively studied in domains such as images, speech, and computer vision [56], their exploration in the context of 3D kinematic data remains comparatively limited.

Most approaches for modeling 3D kinematic data are based on time series strategies, which have been explored in the context of time series classification [57], time series forecasting [58], and anomaly detection over time [59]. However, these methods are still under development due to the inherent challenges posed by time series data, such as temporal dependencies among variables. In fact, the modeling of multivariate time series remains underdeveloped, as current models are still unable to fully capture the complex dynamics of variables over time [60]. In time domain methods, changes are typically applied directly to the input time series. Common strategies include introducing noise patterns, such as Gaussian noise, spikes, or slope-like trends [60]. Frequency domain strategies, on the other hand, involve transformations applied to the original data in the frequency domain, although they are less prevalent than time domain methods. For example, Gao et al. [61] proposed interfering with the input signal by performing transformations on its amplitude and phase spectrum, using the resulting series to train a convolutional neural network. Lastly, time-frequency domain methods, despite being extensively studied for time series analysis, are still in early stages of development for data augmentation. For instance, Steven Eyobu et al. [62] used short Fourier transformations to extract time-frequency features and train an LSTM network for classification, achieving promising results in terms of performance. However, these strategies face a challenge when applied to 3D kinematic data, as they do not consider the spatial correlation among the three temporal series that represent the spatial trajectories over time. Consequently, alternative methodologies have been developed that incorporate spatial transformations while preserving the overall shape of the trajectory and the inherent progression of the sequence. The most prevalent transformations in this context include scaling, rotation, and translation of the 3D trajectories.

4 | MATERIALS AND METHODS

4.1 | Data

Data collection for this study involved capturing the spatial position of a laparoscopic instrument during the execution of three distinct pre-defined trajectories, considering a sampling rate of 30 samples per second. Participants were tasked with performing navigation movements while avoiding a set of pegs using the laparoscopic instrument within an Endo-trainer simulator (see Figure 4). Each participant completed a total of 20 attempts for each trajectory type, with an even distribution between their left and right hand, resulting in 10 attempts per hand. The arrangement of the pegs varied within each trajectory type, leading to distinct movement requirements compared to the other two trajectories. As a result, the initial database comprised 600 different trials. These trajectories were specifically designed to train and assess the spatial perception of trainees when using laparoscopic instruments. The number of trials was determined based on several factors, including participant availability, the dynamic nature and diversity of the gestures, and the need for data quality and consistency. Each gesture within the trials encompassed diverse depth and curvature characteristics, representative of the varied forms and execution patterns inherent in hand gesture recognition. Despite the modest size, the dataset is considered sufficiently representative due to the comprehensive range of depth perceptions critical in laparoscopic surgery covered by the trials. Moreover, the implementation of a subsequent data augmentation strategy, coupled with the capabilities of the proposed deep learning models, particularly the LSTM-based approach with an attention mechanism, enables effective learning and pattern extraction from this initial dataset, maximizing the utility of each trial.

The acquired dataset contains several pieces of relevant information for the study. Firstly, the participant ID served as a unique identifier for each of the 10 participants. The trajectory label indicated one of the three distinct trajectory types, while the hand marker specified whether the gesture was executed with the right or left hand. Additionally, for each participant, a trial ID distinguished each of the 10 different attempts performed for each trajectory and hand combination. Lastly, the XYZ-coordinates represented the 3D position of the instrument at each time step.

4.2 | Proposed Methodology

Figure 5 provides an overview of the methodology followed in this study. The section is structured to offer a comprehensive explanation of the pre-processing steps required for cleaning and preparing the 3D sequences. Subsequently, an in-depth description of the employed data augmentation strategy is presented. Finally, this section discusses the modeling step, encompassing the practical implementation details of the NN-based models used in this research.

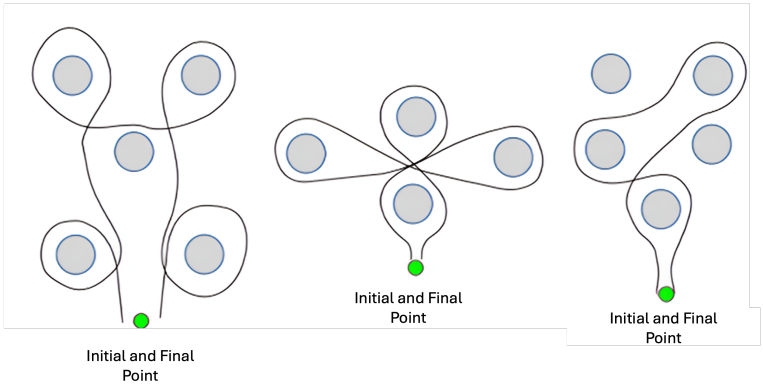


Figure 4 Shape of the three different trajectories in the experiment.

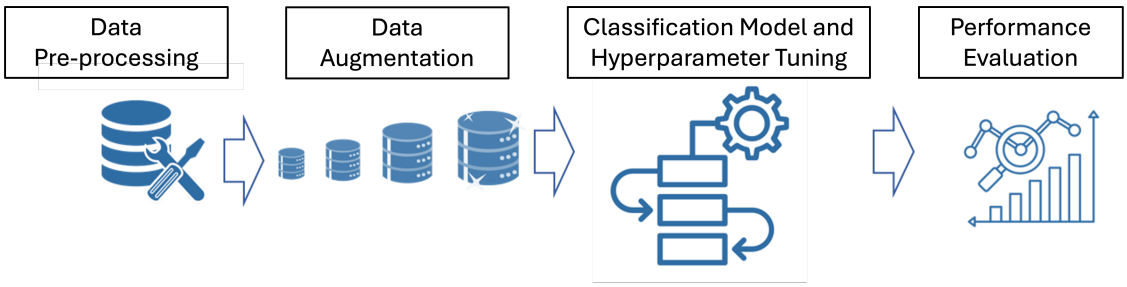


Figure 5 Proposed Methodology Overview: Data Pre-processing, Data Augmentation, and Deep Learning-Based Classification

4.2.1 | Data Pre-processing

The data in this study presented two main challenges: missing values and high-frequency noise. The position coordinates of the instrument at each time step were obtained using six sensors that triangulated the position of the rigid body. However, in some cases, this information was missing, resulting in missing values, or the sensors detected values that were deemed impossible based on the experiment's configuration, resulting in outliers. To address these issues, a strategy was developed that leveraged the temporal nature of the data. Time series data exhibit a high correlation between data points at time steps $t-1$ and $t+1$ with the data point at time step t. Since the number of missing points between two acquired data points was minimal in this study, a linear interpolation method was employed to estimate the missing values at time step t for each axis. This involved computing a linear function between the known values at time steps $t-1$ and $t+1$ to estimate the value at t. By performing this interpolation, the missing values were replaced, improving the overall completeness of the dataset. Linear interpolation is a widely used technique in the literature for handling missing values, particularly when the missing points are limited within a certain range ([63],[64]). During this stage, it was observed that certain trajectories exhibited high-frequency noise, which could negatively impact the performance of the neural network models.

To mitigate the noise and improve the quality of the trajectories, a filtering technique known as Savitzky-Golay filter was applied. This filter is effective in smoothing out trajectories by fitting a polynomial curve to the data [65]. The implementation of the Savitzky-Golay filter involves two key parameters: the polynomial order and the window size of the smoothing kernel. To identify the optimal configuration, various window sizes, ranging from 5 to 19, and polynomial orders between 2 and 5, were tested in this study. The selection of the window size and polynomial order for the filter was based on achieving a balance between effectively smoothing the data and preserving the essential characteristics of the original signal. To this end, a criterion focused on minimizing the deviation between the original trajectory data and the filtered output was adopted. This approach facilitated a comparative analysis of the trajectories filtered across varying window sizes and polynomial orders. The aim was to identify the configuration that exhibited the least deviation from the original dataset. The experimental results led to the selection of a window size of 17 and a polynomial order of 2 as the most effective combination for the data. This specific configuration, a window size of 17, was chosen because it allowed for sufficient smoothing of the high-frequency noise, while still retaining significant details in the trajectory. The polynomial order of 2 was found to be optimal in fitting the data closely enough to mitigate noise, yet not so rigidly as to introduce distortions to the data. These values represent a carefully considered compromise, selected after testing various combinations for their impact on noise reduction and fidelity to the original data.

4.2.2 | Data Augmentation

Upon completing the pre-processing of the trajectories, the next step involved implementing the data augmentation strategy. This approach aimed to generate new trajectories by leveraging the existing ones, but with the caveat of enforcing appropriate constraints to ensure the meaningfulness of the samples. Additionally, the mathematical operations applied to the sequences were tailored to suit the specific domain under investigation. The proposed method in this study involved scaling, rotation, and translation to augment the data while preserving the spatial relationships in the trajectories. Consequently, the variations generated are representative of realistic gesture alterations while preserving the integrity of the original gesture patterns. The general transformation applied to each 3D sequence $r(t) = [x(t), y(t), z(t)]$ was formulated using scaling, translation, and rotation terms, as depicted in Equations 6 and 7. These equations defined the precise mathematical operations to modify the sequences and generate the augmented samples.

$$r_T(t) = A_{\text{scaling}} \times A_{\text{rotation}} \times r(t) + A_{\text{translation}} \tag{6}$$

where:

$$\begin{bmatrix} x_T(t) \\ y_T(t) \\ z_T(t) \end{bmatrix} = \begin{bmatrix} A_{sx} & 0 & 0 \\ 0 & A_{sy} & 0 \\ 0 & 0 & A_{sz} \end{bmatrix} \times \begin{bmatrix} \mathrm{C}(\gamma)\mathrm{C}(\beta) & -\mathrm{S}(\gamma)\mathrm{C}(\alpha) + \mathrm{C}(\gamma)\mathrm{S}(\beta)\mathrm{S}(\alpha) & \mathrm{S}(\gamma)\mathrm{S}(\alpha) + \mathrm{C}(\gamma)\mathrm{S}(\beta)\mathrm{C}(\alpha) \\ \mathrm{S}(\gamma)\mathrm{C}(\beta) & \mathrm{C}(\gamma)\mathrm{C}(\alpha) + \mathrm{S}(\gamma)\mathrm{S}(\beta)\mathrm{S}(\alpha) & -\mathrm{C}(\gamma)\mathrm{S}(\alpha) + \mathrm{S}(\gamma)\mathrm{S}(\beta)\mathrm{C}(\alpha) \\ -\mathrm{S}(\beta) & \mathrm{C}(\beta)\mathrm{S}(\alpha) & \mathrm{C}(\beta)\mathrm{C}(\alpha) \end{bmatrix} \times \begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix} + \begin{bmatrix} A_{tx} \\ A_{ty} \\ A_{tz} \end{bmatrix} \tag{7}$$

The resulting coordinates for time t after the spatial transformation are denoted as $r_T(t) = [x_T(t), y_T(t), z_T(t)]^T$. To perform the transformation, scaling factors A_{sx}, A_{sy}, and A_{sz} are applied to each axis x, y, and z, respectively. The angles α, β, and γ associated with the x, y, and z axes are used to compute the cosine (C) and sine (S) functions. In addition to scaling, rotation, and the associated angles, translation parameters A_{tx}, A_{ty}, and A_{tz} are used to consider the spatial shifts along each axis. Together, these parameters define the spatial transformation applied to the 3D sequences. The variation range of the transformation parameters is determined based on the standard deviation of the trajectory values. Specifically, the minimum and maximum values for scaling, translation and rotation are evaluated within a range of plus or minus three times the standard deviation. This range provides controlled and diverse variations to the transformed trajectories. By leveraging these equations and considering the statistical properties of the trajectory values, the proposed data augmentation strategy ensures that the generated samples exhibit controlled variations in scaling, rotation, and translation.

Augmenting the dataset with these spatial transformations significantly enhances its diversity, addressing the challenge of limited data availability in the specialized field of surgery. Surgical gestures, influenced by individual surgeon proficiency and the complexity of movements, can vary greatly. By introducing a broader spectrum of scenarios through augmentation, the model to be trained is better equipped to learn and accurately classify different gestures. Furthermore, this strategy of including both original and augmented data in the training set is relevant in mitigating the risk of overfitting, ensuring that the model generalizes across different yet realistic variations of surgical movements, rather than merely memorizing specific gestures.

4.2.3 | Modelling

The primary objective of this paper is to introduce a technique for hand gesture classification based on LSTM architecture with attention mechanisms. The proposed approach's performance is compared with several DL models, including Vanilla LSTM, Vanilla GRU, BiLSTM, and BiGRU. The evaluation is conducted on three distinct data arrangements, namely A-I, A-II, and A-III. In A-I, the complete dataset was used as input, while in A-II, only the sequences performed with the right hand were included. This comparison aimed to assess whether the models' performance was affected by the dynamic differences introduced when using the non-dominant hand. The A-III arrangement was employed to determine if the models could distinguish between each trajectory and the hand used, using the entire database. This experiment sought to evaluate if the NNs could accurately detect these dynamics, resulting in favorable evaluation metrics. Otherwise, it would suggest that processing the dominant hand information separately would yield better results.

Before presenting the results, certain considerations were made. NNs typically require quantitative features to be scaled and qualitative response variables to be encoded. In this study, one-hot encoding was utilized to encode the response categories as dummy variables, while min-max scaling was applied to the quantitative features. The scaling process transformed the training data into the range of $(0, 1)$ and applied the same transformation to the testing set. It should be noted that all tested models required sequences of the same length. Although the acquisition frequency was consistent across all trials, the speed varied depending on the participant, resulting in sequences of different lengths. To address this, sequences shorter than the maximum length were padded with zeros.

To optimize the models, a comprehensive grid-based search was conducted to identify the best combination of parameter values. The three parameters tuned for all models were the number of epochs, batch size, and the number of units per layer. The range of values tested for epochs encompassed [100, 200, 300], carefully chosen to strike a balance between preventing overfitting and ensuring convergence. Batch size, which directly impacted the number of training samples considered during each update, also played a crucial role in the convergence of the neural

networks. Multiple batch sizes were experimented with, including [32, 64, 128]. Lastly, the values for the number of neurons in the hidden layers were explored in this paper, encompassing [32, 64, 128]. Altogether, a total of 27 distinct combinations of these parameters were meticulously employed to train the neural networks, enabling the identification of the most effective set of hyperparameters for achieving superior performance. Furthermore, it is important to note that the process of tuning these parameters was conducted using cross-validation with 5 folds. This approach was chosen to ensure a more robust and generalizable model by evaluating its performance across different subsets of the data.

Other parameters were set to commonly used values in the literature. Dropout, which excludes a percentage of units during each iteration to prevent overfitting, was set to 20%. The Adam optimization algorithm with a learning rate of 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1e-08$ was chosen for its computational efficiency and ability to handle large and noisy datasets. The training set was divided into train and validation sets, with a hold-out strategy utilizing 10% of the training set for validation. For discrete-valued outputs, softmax activation with cross-entropy loss functions was commonly employed. These considerations and parameter settings were implemented to ensure a rigorous evaluation and optimization process for the DL models used in this study.

5 | EXPERIMENTAL RESULTS

Results described in this section are based on the methodology previously presented. All the models were compiled using Python 3.8 and Tensor Flow 2.4.0, and a processor Intel(R) Core(TM) i5-5200U.

5.1 | Pre-processing results

This section focuses on presenting and explaining the key results related to the pre-processing steps. Figure 6 provides an example of the raw sequences from the first trial of each participant for the second trajectory. In this particular example, one notable issue observed among various signal problems is the presence of outliers, which are clearly visible.

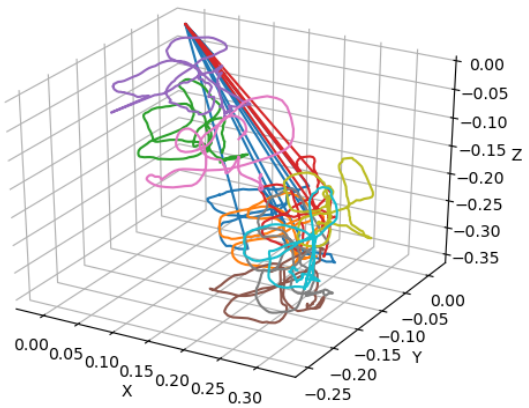


Figure 6 Raw data of each participants' first trial for the second trajectory

To provide an overview of the pre-processing strategy's application and the implementation of the data augmentation strategy, visual comparisons are presented in Figures 7-9. These figures illustrate the coordinates before and after the pre-processing methodology for each axis. Notably, higher levels of noise are observed in the left-hand trials compared to the dominant-handed trials. However, by applying the pre-processing steps, significant noise reduction is achieved, leading to distributions that closely resemble those of the right-handed trials. These results underscore the effectiveness of the pre-processing strategy in reducing noise and aligning the distributions of left-handed and right-handed trials.

The final stage of the pre-processing procedure involved the application of the Savitzky-Golay filter to all the signals to effectively reduce high-frequency noise. As depicted in Figure 10, this noise often disrupted the smoothness of the gesture realization, particularly noticeable in the Z-axis. The Savitzky-Golay filter demonstrated its ability to correct these disruptions, effectively preserving the essential shape of the trial while eliminating the unwanted noise, resulting in more accurate and reliable data for subsequent analysis. Figures 11 and 12 provide valuable insights

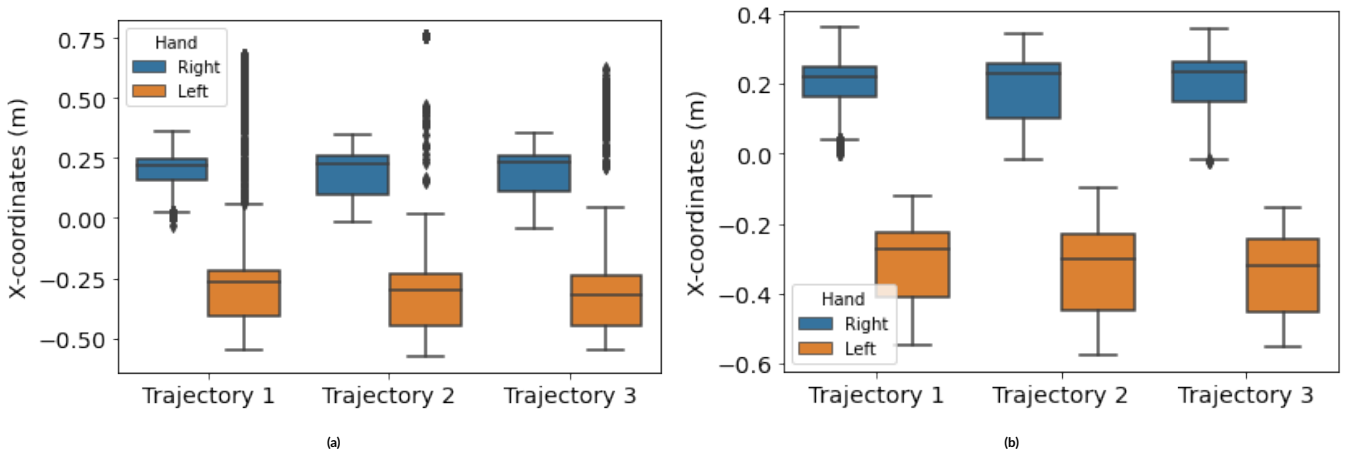


Figure 7 (a) Original distribution of the X coordinates for each hand and trajectory. (b) Distribution of the X coordinates after pre-processing for each hand and trajectory.

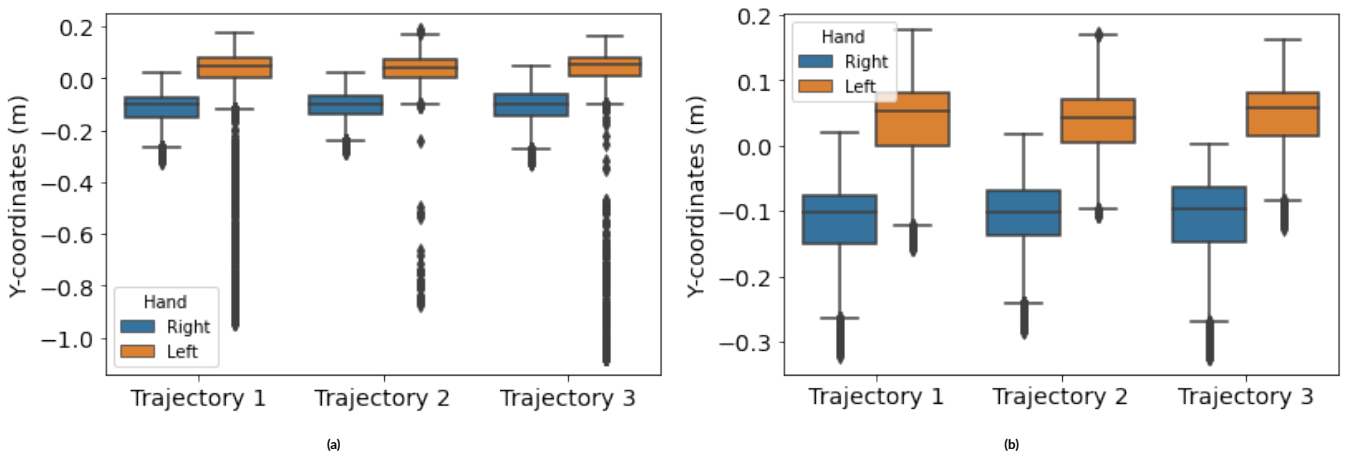


Figure 8 (a) Original distribution of the Y coordinates for each hand and trajectory. (b) Distribution of the Y coordinates after pre-processing for each hand and trajectory.

into the inter- and intra-variability of the trajectories, as well as the quality of samples based on hand dominance. Figure 11 focuses on intra-variability, showcasing the consistency when trajectories are performed by the same subject using the same hand, resulting in minimal variation. In contrast, when the gestures are executed with the non-dominant hand, intra-variability increases. Figures 12a to 12b illustrate inter-variability, where each realization is performed by a different participant. As expected, the variability between trials is higher when considering multiple subjects, compared to the aforementioned intra-variability. Notably, gestures executed with the dominant hand (right hand) exhibit a clearer and more defined shape, while those performed using the non-dominant hand (left hand) demonstrate greater variety in instrument orientation and a wider range of execution coordinates. It is also noteworthy that the first and second trajectories exhibit a closer similarity in shape compared to the third trajectory, which may have implications when analyzing the model results.

5.2 | Data Augmentation results

Initially, each participant performed 10 trials per trajectory and hand, resulting in a potential total of 40 trials per trajectory, hand, and participant after the application of the data augmentation strategy. The subsequent Figures 13-15 exemplify the augmented trajectories for each trajectory type and hand. These illustrations demonstrate that the shape of the original trials is preserved despite the applied transformations. Importantly, these augmented trajectories introduce new information to the models, providing insight into the potential range of coordinate values that could have been acquired.

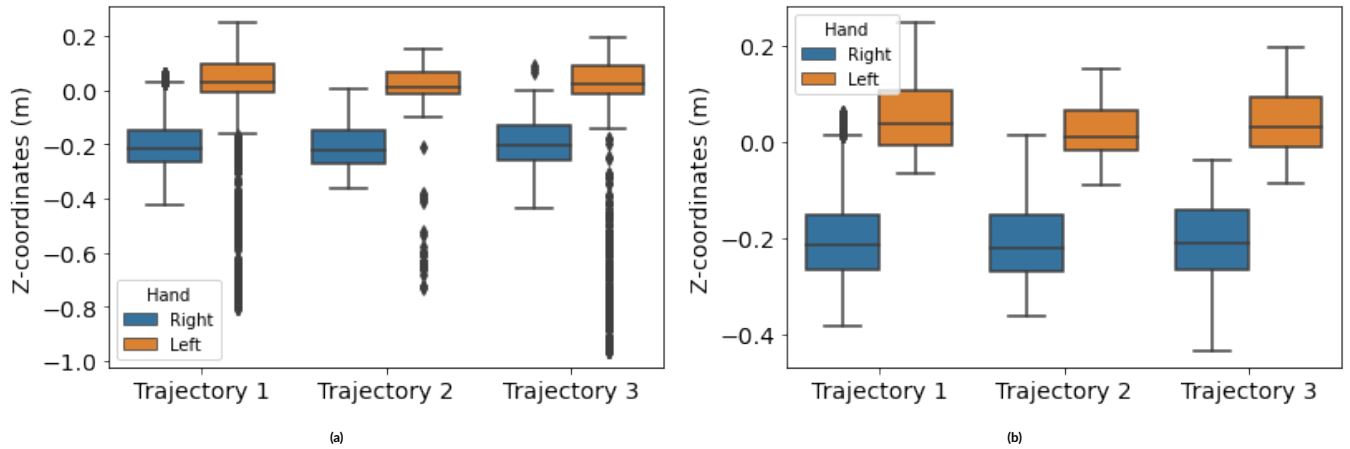


Figure 9 (a) Original distribution of the Z coordinates for each hand and trajectory. (b) Distribution of the Z coordinates after pre-processing for each hand and trajectory.

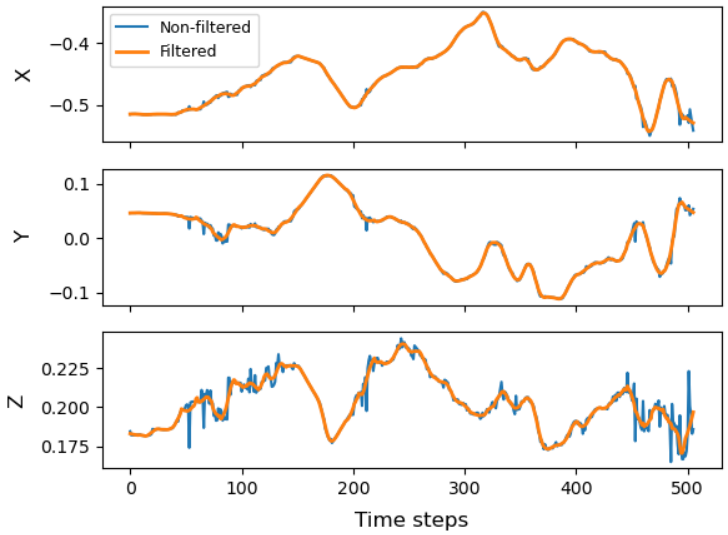
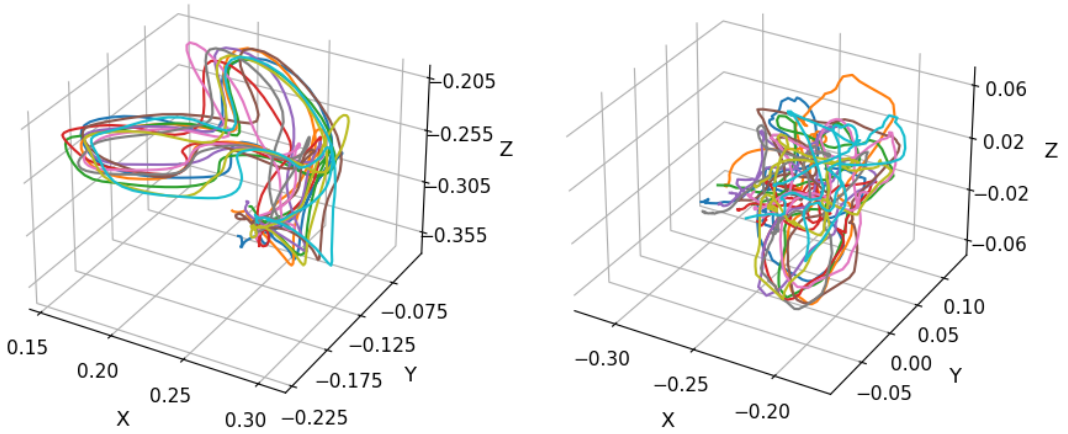


Figure 10 Smoothing procedure for one trial of a participant, second trajectory and left hand, comparing the non-filtered trajectory to the filtered one, by axis, using a window size of 17.

5.3 | Classification Model

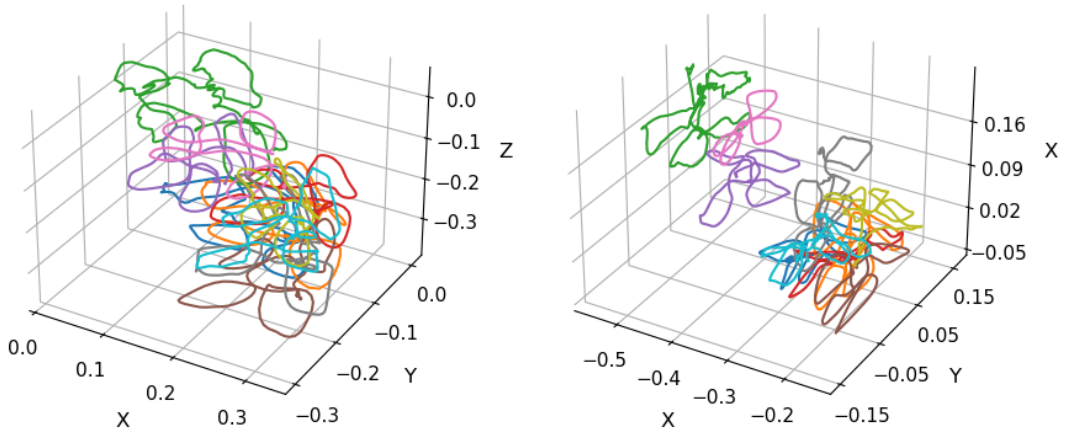
The dataset analyzed in this study for classification purposes exhibited a crucial characteristic: it was completely balanced, meaning that each class had an equal number of instances. This balance enabled a random train-test split, where 70% of the data was allocated for training and the remaining 30% for testing. As discussed in Section 4.2.3, three distinct data arrangements were investigated to evaluate the impact of using a non-dominant hand on the modeling performance. In this way, this section presents the analysis of the results in two distinct steps. Firstly, for the proposed architecture, a comprehensive examination is conducted, considering various hyper-parameter configurations utilized in training the algorithm for each data arrangement. The selection of the best configuration for each A-I, A-II, and A-III is based on the evaluation of performance metrics. The configurations of the models encompass variations in batch size, the number of units in the hidden layer, and the number of epochs. In terms of notation, T1, T2, and T3 represent the three different trajectories, while T1-L or T2-R indicate T1 performed with the left hand and T2 performed with the right hand, respectively. The second step of the analysis involves a comparison with baseline methodologies, showcasing the final configuration that yielded the best results for each architecture. This comparison allows for an assessment of the effectiveness of the proposed architecture relative to other strategies reported in the literature and the identification of the optimal configuration.

Table 1 provides an overview of the overall results achieved by the LSTM with attention mechanism across the three different data arrangements.



(a) Right hand (b) Left hand

Figure 11 Comparison between the right and left hand among the 10 trials of a participant for the third trajectory.

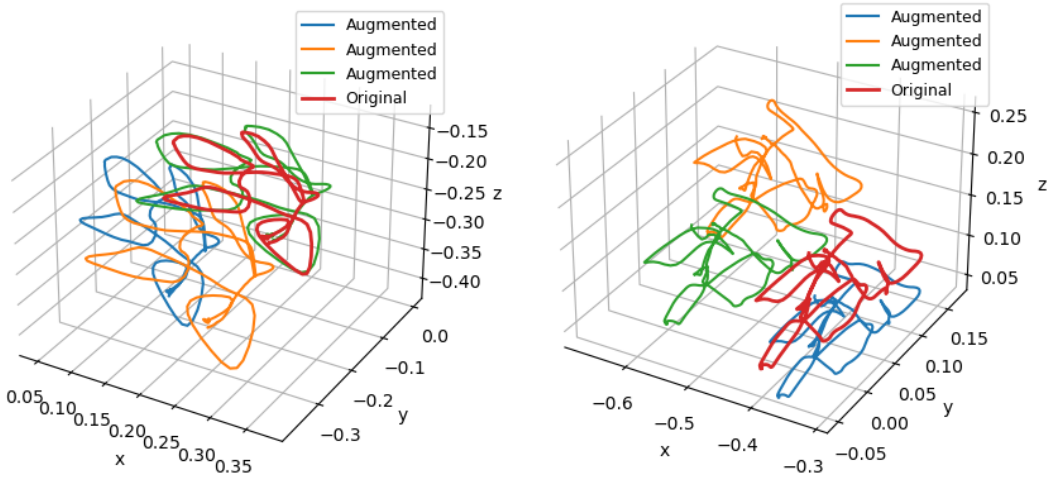


(a) Right hand (b) Left hand

Figure 12 Comparison between right and left hand for the first trajectory. Each realization belongs to a different participant.

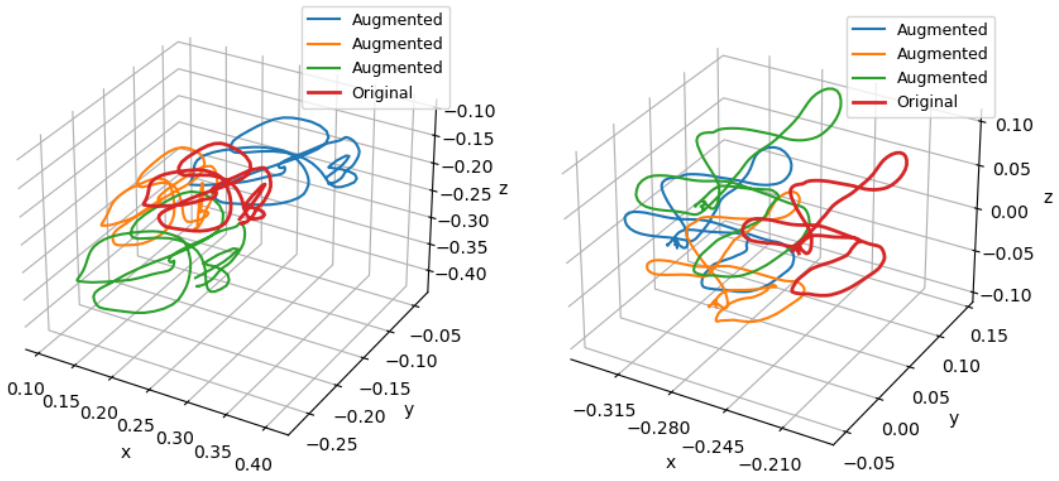
In this study, several strategies have been implemented, which serve as baselines for comparison in the field. These strategies include LSTM, BiLSTM, GRU, and BiGRU models. The LSTM architecture, in particular, has been evaluated, and the results are presented in Table 2. Notably, the performance of LSTM varies across different data arrangements (A-I, A-II, and A-III). The largest differences between epochs are observed for A-I, where an increase in accuracy is evident with larger epochs. Additionally, larger batch sizes also lead to improved results in this particular case. For A-I, the best accuracy value achieved was 95.64%, utilizing a combination of parameters with 300 epochs, 128 units in the hidden layer, and a batch size of 128. Conversely, the lowest accuracy was obtained for A-III, where the selected combination included 300 epochs, a batch size of 32, and 128 neurons in the hidden layer. The A-II model demonstrated an accuracy of 95.47%, with 100 epochs and a batch size and units set to 128.

Subsequently, the results of the different configurations used for BiLSTM are presented in Table 3. Similar to the LSTM architecture, the accuracy of the BiLSTM model varies across different data arrangements. For A-I, the highest accuracy value of 94.41% was achieved using a configuration with a batch size of 128, 128 units in the hidden layer, and 200 epochs. Interestingly, this same result was obtained with a different configuration of 300 epochs, a batch size of 32, and 64 units in the hidden layer. For A-II, using a batch size of 32 resulted in a slightly lower accuracy of 94.13%. Finally, for the third data arrangement (A-III), the configuration with 300 epochs, a batch size of 32, and 32 units in the hidden layer achieved an accuracy of 93.73%. Notably, in general, the accuracy rates tended to slightly increase with the number of epochs for all data arrangements. Furthermore, for most of these models, the lower accuracy rates were found when the number of hidden neurons was set to 32. It is important to



(a) Right hand (b) Left hand

Figure 13 Data augmentation strategy applied to one trial of the first trajectory and: (a) right hand, (b) left hand.



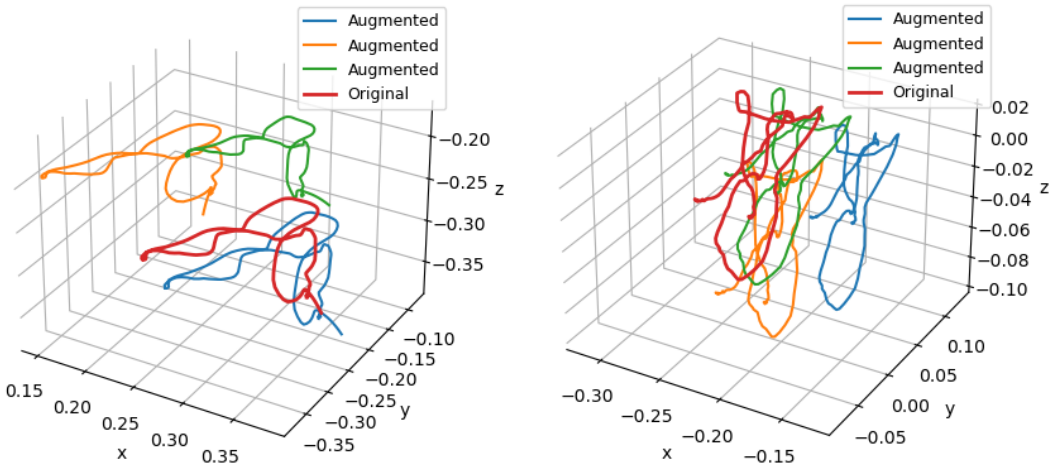
(a) Right hand (b) Left hand

Figure 14 Data augmentation strategy applied to one trial of the second trajectory and: (a) right hand, (b) left hand.

note that the performance of the BiLSTM model closely resembles that of the LSTM architecture, and the choice of hyperparameters significantly influences the accuracy of the model.

The results obtained for the GRU model with different variations of hyperparameter configurations are summarized in Table 4. For A-I, the highest accuracy of 95.23% was achieved with 200 epochs, 128 units in the hidden layer, and a batch size of 128. Similarly, for A-II, the model attained a comparable accuracy of 94.67% using 100 epochs, 32 hidden layer units, and a batch size of 64. The lowest accuracy percentage was observed for A-III, reaching 94.14% with 300 epochs, 64 units in the hidden layer, and a batch size of 128. As it can be seen, the number of epochs seems to influence the accuracy performance. Models with more than 200 epochs generally exhibited higher accuracy rates, especially when combined with larger units and batch sizes. Conversely, using only 100 epochs resulted in lower accuracy.

Finally, the results for the BiGRU configuration are presented in Table 5. For A-I, the highest accuracy of 94.96% was achieved with 200 epochs, 64 units in the hidden layer, and a batch size of 128. On the other hand, A-II obtained an accuracy of 93.33% with 300 epochs and 32 for both batch size and number of units. The lowest accuracy value was observed for A-III, reaching 92.78% with 200 epochs, 64 units in the hidden layer, and a batch size of 64.



(a) Right hand (b) Left hand

Figure 15 Data augmentation strategy applied to one trial of the third trajectory and: (a) right hand, (b) left hand.

		100 epochs			200 epochs			300 epochs		
		Batch size			Batch size			Batch size		
		32	64	128	32	64	128	32	64	128
A-I	32 units	93.99	89.28	94.60	91.19	92.22	95.29	95.81	94.73	95.81
	64 units	95.30	91.38	95.15	94.86	94.55	95.71	96.91	96.86	**97.64**
	128 units	94.85	89.94	94.77	91.89	92.99	94.89	95.63	95.08	97.06
A-II	32 units	93.96	92.99	93.88	92.18	93.80	92.46	92.54	92.19	94.98
	64 units	95.10	94.55	95.19	93.92	94.04	95.56	96.10	94.75	**96.61**
	128 units	94.54	94.14	94.04	93.64	92.68	92.07	93.52	94.34	95.75
A-III	32 units	75.69	94.90	95.54	94.28	90.89	94.92	96.08	**96.39**	91.55
	64 units	82.20	95.82	93.87	93.86	90.43	93.15	94.22	94.04	89.56
	128 units	75.64	91.46	95.06	95.37	92.44	92.59	92.24	93.01	90.25

Table 1 Accuracy (%) of all the explored attention-based LSTM models for A-I, A-II and A-III

		100 epochs			200 epochs			300 epochs		
		Batch size			Batch size			Batch size		
		32	64	128	32	64	128	32	64	128
A-I	32 units	88,56	92,64	92,23	93,32	92,51	93,05	91,01	92,51	92,37
	64 units	90,74	75,89	92,10	92,37	93,73	94,41	93,46	93,05	94,96
	128 units	90,33	92,51	83,92	91,14	88,42	95,10	91,96	91,42	**95,64**
A-II	32 units	92,00	92,00	92,00	92,53	92,80	93,33	92,53	92,53	93,33
	64 units	89,87	91,20	92,53	91,47	91,20	93,33	93,33	92,27	93,07
	128 units	89,60	92,27	**95,47**	87,20	92,80	94,13	94,40	93,60	93,33
A-III	32 units	87,47	87,60	89,51	87,74	89,92	90,87	90,33	88,96	91,69
	64 units	89,92	90,33	91,14	88,15	91,14	92,10	88,01	90,74	89,92
	128 units	88,28	90,19	90,60	88,69	90,05	90,19	**92,92**	91,28	91,69

Table 2 Accuracy (%) of all the explored Vanilla LSTM models for A-I, A-II and A-III

		100 epochs			200 epochs			300 epochs		
		Batch size			Batch size			Batch size		
		32	64	128	32	64	128	32	64	128
A-I	32 units	90,74	90,87	91,01	91,28	93,60	92,92	91,28	92,64	92,10
	64 units	91,14	85,97	93,46	93,05	91,55	91,83	**94,41**	92,92	93,60
	128 units	91,01	92,23	90,74	91,55	91,28	**94,41**	92,37	93,05	93,19
A-II	32 units	89,60	90,67	90,40	93,07	90,93	93,07	91,20	91,73	91,73
	64 units	86,67	93,60	88,27	90,40	90,40	92,80	92,00	92,00	91,73
	128 units	88,53	86,40	90,67	92,27	93,07	92,53	**94,13**	90,40	92,53
A-III	32 units	88,28	87,47	92,10	83,52	90,46	89,78	**93,73**	90,46	91,96
	64 units	90,74	91,83	91,14	90,05	91,83	91,01	90,60	92,23	92,23
	128 units	88,69	90,87	91,14	91,96	92,51	91,42	92,92	92,64	92,64

Table 3 Accuracy (%) of all the explored Bi-LSTM models for A-I, A-II and A-III

		100 epochs			200 epochs			300 epochs		
		Batch size			Batch size			Batch size		
		32	64	128	32	64	128	32	64	128
A-I	32 units	91,69	93,73	93,05	91,83	91,83	90,33	92,92	92,37	92,51
	64 units	90,87	92,37	90,05	93,46	93,60	93,19	93,32	93,87	94,55
	128 units	91,96	91,01	93,87	94,28	94,01	**95,23**	94,28	94,41	94,82
A-II	32 units	92,53	**94,67**	92,53	89,33	92,27	92,00	91,73	92,27	92,53
	64 units	89,87	92,53	91,20	93,87	94,40	91,47	90,40	91,20	93,07
	128 units	89,87	91,73	89,60	92,00	93,33	92,53	92,00	91,47	92,00
A-III	32 units	87,60	84,60	91,14	87,87	92,37	91,96	89,37	90,46	91,83
	64 units	87,60	91,14	92,37	92,37	90,60	92,64	89,37	89,92	**94,14**
	128 units	87,47	90,05	92,51	88,15	91,01	93,05	90,60	90,60	92,37

Table 4 Accuracy (%) of all the explored GRU models for A-I, A-II and A-III

		100 epochs			200 epochs			300 epochs		
		Batch size			Batch size			Batch size		
		32	64	128	32	64	128	32	64	128
A-I	32 units	93,32	89,65	92,92	86,78	94,28	91,01	91,42	93,87	92,10
	64 units	90,60	92,78	93,87	91,01	92,51	**94,96**	93,19	92,92	94,41
	128 units	90,19	91,69	93,05	92,23	92,64	92,51	92,64	93,19	92,78
A-II	32 units	90,40	92,00	92,27	90,67	92,53	92,53	**93,33**	89,60	92,53
	64 units	89,33	93,07	92,53	90,40	91,47	93,07	90,93	92,00	92,00
	128 units	89,07	90,13	89,87	91,47	92,80	91,73	90,40	90,13	92,00
A-III	32 units	88,56	88,83	89,92	90,60	90,87	90,05	89,51	90,05	88,83
	64 units	91,28	91,42	91,69	89,37	**92,78**	90,60	91,01	91,28	92,51
	128 units	89,78	90,87	90,74	87,19	91,96	91,96	91,55	90,87	88,42

Table 5 Accuracy (%) of all the explored Bi-GRU models for A-I, A-II and A-III

In addition, Table 6 provides a comprehensive comparison of the performance achieved by various architectures, including the LSTM, Bidirectional LSTM, GRU, Bidirectional GRU, and our proposal, LSTM with attention mechanisms. Across all data arrangements, the proposed strategy consistently outperformed the other architectures. These results yield several important insights. Firstly, bidirectional models (BiLSTM and BiGRU) did not significantly enhance accuracy compared to LSTM and GRU for most data arrangements. This suggests that incorporating both past and

future information may not be crucial for achieving good classification results with the given dataset. Also, comparing A-I and A-II, it can be observed only an approximate 1% of difference in performance for the best models, with A-I (combining both left and right information) having the higher accuracy. Contrary to the initial hypothesis that the left hand might introduce noise, these results indicate that the models with A-I configuration can effectively capture information from each trajectory regardless of the hand used. For the third data arrangement (A-III), the proposed LSTM-based model once again emerged as the best strategy, achieving an accuracy of 96.34%.

	Accuracy (%)		
Model	**A-I**	**A-II**	**A-III**
LSTM	95.64	95.47	92.92
BiLSTM	94.41	94.13	93.73
GRU	95.23	94.67	94.14
BiGRU	94.96	93.33	92.78
attention-based LSTM	**97.64**	**96.61**	**96.39**

Table 6 Summary of the accuracy values for all the models and data arrangements.

6 | CONCLUSIONS

This study aimed to improve gesture recognition using 3D motion capture data by incorporating a data augmentation strategy. By leveraging spatial transformations such as scaling, rotation, and translation, this strategy effectively expanded the dataset's diversity and size without requiring additional data collection. A key advantage of the data augmentation strategy was its ability to preserve spatial relationships among the three axes when generating augmented samples. This ensured that the transformed trajectories remained contextually meaningful for gesture recognition. By carefully controlling the variations through suitable constraints and statistical considerations, the augmentation process allowed exploration of a wider range of spatial configurations.

The data augmentation's controlled variations were particularly beneficial for training DL models, including attention mechanism-based LSTMs. The integration of attention mechanisms was motivated by their ability to dynamically weigh the importance of different time steps in the input sequence, allowing the models to focus on the most relevant information for accurate predictions. This proved valuable for gesture recognition, involving complex and dynamic movements with varying significance over time. The exposure to a more diverse set of inputs enabled the models to learn robust and discriminative features, leading to improved classification performance. Additionally, the augmented dataset helped prevent overfitting and enhanced the models' generalization ability, making them more effective at handling unseen data.

Moreover, the data augmentation strategy was necessary to address the constraints of the limited dataset size, particularly in configurations with fewer samples. By generating additional trajectories, the augmented dataset facilitated the training of more complex models like the proposed attention-based LSTM, which require larger data volumes to achieve optimal performance. This proved especially vital for arrangements where distinguishing between different trajectories and hand usage was challenging. Overall, the integration of attention mechanism-based LSTMs and the data augmentation strategy offered a powerful approach to gesture recognition from 3D motion capture data. The attention mechanisms allowed the models to dynamically focus on relevant information, while the data augmentation strategy enriched the dataset and improved the learning process. The superior performance of the proposed attention-based model across all data arrangements highlights the effectiveness of this integrated approach in enhancing gesture recognition accuracy.

Moving forward, the scope of future work includes the expansion of the dataset to enhance the robustness and applicability of the findings. It is planned to increase the number of trials and participants, which will allow for the validation and refinement of the model under a broader array of conditions. This expansion is likely to include a more diverse array of surgical scenarios and varying levels of participant expertise, thereby improving the generalizability of the approach across different real-world applications. Additionally, the integration of synthetic data generation techniques such as Generative Adversarial Networks (GANs) will be more extensively explored. GANs have been shown to be capable of generating realistic synthetic data that can be indistinguishable from real data, which could provide a more varied and extensive dataset for training the models. By having the dataset enhanced with GAN-generated synthetic data, not only is an increase in the volume of training data anticipated, but also the introduction of complex, scenario-specific data points that could challenge and refine the predictive capabilities of the models. This approach is expected to simulate a broader variety of surgical movements, thereby enriching the training dataset and pushing the boundaries of what can be learned and predicted by the DL models.

Financial disclosure

None reported.

Conflict of interest

The authors declare no potential conflict of interests.

References

1. Radojcić B, Jokić R, Grebeldinger S, Meljnikov I, Radojić N. *History of minimally invasive surgery*. 62. Elsevier Inc. . 2009

2. Chmarra MK, Kolkman W, Jansen FW, Grimbergen CA, Dankelman J. The influence of experience and camera holding on laparoscopic instrument movements measured with the TrEndo tracking system. *Surgical Endoscopy and Other Interventional Techniques* 2007; 21(11): 2069–2075. doi: 10.1007/s00464-007-9298-5

3. Lam K, Chen J, Wang Z, et al. Machine learning for technical skill assessment in surgery: a systematic review. *NPJ Digital Medicine* 2022; 5(1): 24.

4. Gupta S, Bagga S, Sharma DK. Hand Gesture Recognition for Human Computer Interaction and Its Applications in Virtual Reality. *Advanced Computational Intelligence Techniques for Virtual Reality in Healthcare* 2020: 85–105.

5. Li C, Xie C, Zhang B, Chen C, Han J. Deep Fisher discriminant learning for mobile hand gesture recognition. *Pattern Recognition* 2018; 77: 276–288. doi: 10.1016/j.patcog.2017.12.023

6. Stern H, Shmueli M, Berman S. Most discriminating segment - Longest common subsequence (MDSLCS) algorithm for dynamic hand gesture classification. *Pattern Recognition Letters* 2013; 34(15): 1980–1989. doi: 10.1016/j.patrec.2013.02.007

7. Cardenas EJE, Chavez GC. Multimodal hand gesture recognition combining temporal and pose information based on CNN descriptors and histogram of cumulative magnitudes. *Journal of Visual Communication and Image Representation* 2020; 71: 102772.

8. Forestier G, Petitjean F, Senin P, et al. Surgical motion analysis using discriminative interpretable patterns. *Artificial intelligence in medicine* 2018; 91: 3–11.

9. Cotin S, Stylopoulos N, Ottensmeyer M, Neumann P, Rattner D, Dawson S. Metrics for laparoscopic skills trainers: The weakest link!. In: Springer. ; 2002: 35–43.

10. Reiley CE, Lin HC, Yuh DD, Hager GD. Review of methods for objective surgical skill evaluation. *Surgical endoscopy* 2011; 25: 356–366.

11. Plouffe G, Cretu AM. Static and dynamic hand gesture recognition in depth data using dynamic time warping. *IEEE Transactions on Instrumentation and Measurement* 2016; 65(2): 305–316. doi: 10.1109/TIM.2015.2498560

12. Somoyeh B, Khurshid A, Ehsan T. Using two-third power law for segmentation of hand movement in robotic assisted surgery. In: ; 2015; Boston, Massachusetts, USA: 1–6

13. Abend WBE, Morasso P. Human arm movement trajectory formation. *Brain* 1982; 105: 331–348.

14. Feng KP, Yuan F. Static hand gesture recognition based on HOG characters and support vector machines. *Proceedings - 2013 2nd International Symposium on Instrumentation and Measurement, Sensor Network and Automation, IMSNA 2013* 2013: 936–938. doi: 10.1109/IMSNA.2013.6743432

15. Cifuentes J, Pham MT, Boulanger P, Moreau R, Prieto F. Towards a classification of surgical skills using affine velocity. *IET Science, Measurement and Technology* 2018; 12(4): 548–553. doi: 10.1049/iet-smt.2017.0373

16. Qing C, Georganas ND, Petriu EM. Real-time vision-based hand gesture recognition using haar-like features. *Conference Record - IEEE Instrumentation and Measurement Technology Conference* 2007. doi: 10.1109/imtc.2007.379068

17. Cifuentes J, Pham MT, Moreau R, Boulanger P, Prieto F. Medical gesture recognition using dynamic arc length warping. *Biomedical Signal Processing and Control* 2019; 52: 162–170. doi: 10.1016/j.bspc.2019.04.022

18. Rosen J, Hannaford B, Richards CG, Sinanan MN. Markov modeling of minimally invasive surgery based on tool/tissue interaction and force/-torque signatures for evaluating surgical skills. *IEEE Transactions on Biomedical Engineering* 2001; 48(5): 579–591. doi: 10.1109/10.918597

19. Choi HR, Kim TY. Directional dynamic time warping for gesture recognition. *ACM International Conference Proceeding Series* 2017(January): 22–25. doi: 10.1145/3145511.3145526

20. Vlachos M, Kollios G, Gunopulos D. Discovering similar multidimensional trajectories. In: ; 2002: 673-684

21. Zeng M, Xiao L, Li R. A K-NN and sparse representation based method for gesture recognition. *Proceedings - 2013 IEEE International Conference on High Performance Computing and Communications, HPCC 2013 and 2013 IEEE International Conference on Embedded and Ubiquitous Computing, EUC 2013* 2014: 2325–2329. doi: 10.1109/HPCC.and.EUC.2013.334

22. Cifuentes J, Boulanger P, Prieto F. Tool Gesture and Motion Classification using Gated Recurrent Neural Networks. 2019: 1–25.

23. Kaluri R, Reddy P. Sign gesture recognition using modified region growing algorithm and adaptive genetic fuzzy classifier. *International Journal of Intelligent Engineering and Systems* 2016; 9(4): 225–233.

24. Kaluri R, Pradeep Reddy C. A framework for sign gesture recognition using improved genetic algorithm and adaptive filter. *Cogent Engineering* 2016; 3(1): 1251730.

25. Moustris GP, Hiridis SC, Deliparaschos KM, Konstantinidis KM. Evolution of autonomous and semi-autonomous robotic surgical systems: a review of the literature. *The international journal of medical robotics and computer assisted surgery* 2011; 7(4): 375–392.

26. Maung THH. Real-time hand tracking and gesture recognition system using neural networks. *World Academy of Science, Engineering and Technology* 2009; 38: 470–474. doi: 10.5281/zenodo.1333642

27. Yanik E, Intes X, Kruger U, et al. Deep Neural Networks for the Assessment of Surgical Skills: A Systematic Review. 2021.

28. Bailador G, Roggen D, Tröster G, Triviño G. Real time gesture recognition using Continuous Time Recurrent Neural Networks. *BODYNETS 2007 - 2nd International ICST Conference on Body Area Networks* 2007. doi: 10.4108/bodynets.2007.149

29. Mazomenos E, Watson D, Kotorov R, Stoyanov D. Gesture Classification in Robotic Surgery using Recurrent Neural Networks with Kinematic Information. *8th Joint Workshop on New Technology for Computer/Robot Assisted Surgery (CRAS 2018)* 2018: 6–7.

30. Cifuentes J, Boulanger P, Pham MT, Prieto F, Moreau R. Gesture Classification Using LSTM Recurrent Neural Networks. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS* 2019(2): 6864–6867. doi: 10.1109/EMBC.2019.8857592

31. Haseeb MAA, Parasuraman R. Wisture: RNN-based Learning of Wireless Signals for Gesture Recognition in Unmodified Smartphones. 2017: 1–10.

32. Hung* A, Aastha n, Nguyen J, Aron K, Damerla V, Liu Y. DEEP-LEARNING BASED COMPUTER VISION TO AUTOMATE IDENTIFICATION OF SUTURING GESTURES. *Journal of Urology* 2020; 203(Supplement 4): e506-e506. doi: 10.1097/JU.0000000000000878.08

33. Lefebvre G, Berlemont S, Mamalet F, Garcia C. BLSTM-RNN based 3D gesture classification. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 2013; 8131 LNCS: 381–388. doi: 10.1007/978-3-642-40728-4$_48$

34. Zhang D, Wang R, Lo B. Surgical Gesture Recognition Based on Bidirectional Multi-Layer Independently RNN with Explainable Spatial Feature Extraction. 2021.

35. Köpüklü O, Gunduz A, Kose N, Rigoll G. Real-time hand gesture detection and classification using convolutional neural networks. *Proceedings - 14th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2019* 2019. doi: 10.1109/FG.2019.8756576

36. Ozdemir MA, Kisa DH, Guren O, Akan A. Hand gesture classification using time–frequency images and transfer learning based on CNN. *Biomedical Signal Processing and Control* 2022; 77: 103787.

37. Gadekallu TR, Alazab M, Kaluri R, Maddikunta PKR, Bhattacharya S, Lakshmanna K. Hand gesture classification using a novel CNN-crow search algorithm. *Complex & Intelligent Systems* 2021; 7: 1855–1868.

38. Alam MM, Islam MT, Rahman SMM. Unified Learning Approach for Egocentric Hand Gesture Recognition and Fingertip Detection. 2021.

39. Tang Z, Liu Q, Wu M, Chen W, Huang J. WiFi CSI gesture recognition based on parallel LSTM-FCN deep space-time neural network. *China Communications* 2021; 18(3): 205–215. doi: 10.23919/JCC.2021.03.016

40. Ikram A, Liu Y. Skeleton based dynamic hand gesture recognition using LSTM and CNN. *ACM International Conference Proceeding Series* 2020: 63–68. doi: 10.1145/3421558.3421568

41. Hutchinson K, Reyes I, Li Z, Alemzadeh H. Evaluating the Task Generalization of Temporal Convolutional Networks for Surgical Gesture and Motion Recognition using Kinematic Data. *IEEE Robotics and Automation Letters* 2023.

42. Guo R, Li H, Zhang C, Qian X. A tree-structure-guided graph convolutional network with contrastive learning for the assessment of parkinsonian hand movements. *Medical Image Analysis* 2022; 81: 102560.

43. Peng SH, Tsai PH. An Efficient Graph Convolution Network for Skeleton-Based Dynamic Hand Gesture Recognition. *IEEE Transactions on Cognitive and Developmental Systems* 2023.

44. Liao J, Xiong P, Liu PX, Li Z, Song A. Enhancing Robotic Tactile Exploration With Multireceptive Graph Convolutional Networks. *IEEE Transactions on Industrial Electronics* 2023.

45. Heaton J. *Introduction to Neural Networks for Java*. 99 . 2008.

46. Lee T, Singh VP, Cho KH. Deep Learning for Time Series. 2021: 107–131. doi: 10.1007/978-3-030-64777-3$_9$

47. Neapolitan RE, Jiang X. *Neural Networks and Deep Learning* . 2018

48. Maraqa M, Abu-Zaiter R. Recognition of Arabic Sign Language (ArSL) using recurrent neural networks. *1st International Conference on the Applications of Digital Information and Web Technologies, ICADIWT 2008* 2008: 478–481. doi: 10.1109/ICADIWT.2008.4664396

49. Bengio Y, Simard P, Frasconi P. Learning Long-Term Dependencies with Gradient Descent is Difficult. *IEEE Transactions on Neural Networks* 1994; 5(2): 157–166. doi: 10.1109/72.279181

50. Cho K, Merrienboer vB, Bahdanau D, Bengio Y. On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. 2015: 103–111. doi: 10.3115/v1/w14-4012

51. Zhao B, Xing H, Wang X, Song F, Xiao Z. Rethinking Attention Mechanism in Time Series Classification. *Information Sciences* 2023.

52. Zhang C, Bengio S, Hardt M, Recht B, Vinyals O. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM* 2021; 64(3): 107–115. doi: 10.1145/3446776

53. DeVries T, Taylor GW. Dataset augmentation in feature space. *5th International Conference on Learning Representations, ICLR 2017 - Workshop Track Proceedings* 2019: 1–12.

54. Cubuk ED, Zoph B, Mane D, Vasudevan V, Le QV. AutoAugment: Learning Augmentation Policies from Data. *Cvpr 2019* 2019(Section 3): 113–123.

55. Shorten C, Khoshgoftaar TM. A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data* 2019; 6(1). doi: 10.1186/s40537-019-0197-0

56. Zhang Y, Gao J, Zhou H. Breeds Classification with Deep Convolutional Neural Network. *ACM International Conference Proceeding Series* 2020: 145–151. doi: 10.1145/3383972.3383975

57. Fawaz HI, Forestier G, Weber J. Deep learning for time series classification: a review.. *Data Mining and Knowledge Discovery* 2019; 33(4): 917—963.

58. Han Z, Zhao J, Leung H, Ma K, Wang W. A review of deep learning models for time series prediction.. *IEEE Sensors Journal* 2019: 1.

59. Gamboa J. Deep learning for time-series analysis. 2017.

60. Wen Q, Sun L, Yang F, et al. Time Series Data Augmentation for Deep Learning: A Survey. 2021: 4653–4660. doi: 10.24963/ijcai.2021/631

61. Gao J, Song X, Wen Q, Wang P, Liang S, Xu H. Robusttad: Robust time series anomaly detection via decomposition and convolutional neural networks. *MileTS'20: 6th KDD Workshop on Mining and Learning from Time Series* 2020: 1–6.

62. Eyobu SO, Han DS. Feature representation and data augmentation for human activity classification based on wearable IMU sensor data using a deep LSTM neural network.. *Sensors* 2018; 18(9): 2892.

63. Terry W, Lee J, Kumar A. Time series analysis in acid rain modeling: Evaluation of filling missing values by linear interpolation. *Atmospheric Environment (1967)* 1986; 20(10): 1941-1943. First International Conference on Atmospheric Sciences and Applications to Air Quality Part IIdoi: https://doi.org/10.1016/0004-6981(86)90335-5

64. Yen N, Chang J, Liao J. Analysis of interpolation algorithms for the missing values in IoT time series: a case of air quality in Taiwan.. *J Supercomput* 2020; 76: 6475—6500.

65. Schafer RW. What Is a Savitzky-Golay Filter? [Lecture Notes]. 2011(July): 111–117.

How to cite this article: Sorbet S.,and J. Cifuentes (2023), Deep Learning-Based Gesture Recognition for Surgical Applications: A Data Augmentation Approach, *Expert Systems.*, *2023;XX*.