REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET

DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE ABOU BEKR BELKAID TLEMCEN

Faculté de technologie

These

Présentée pour l'obtention du grade de DOCTORAT 3ème Cycle

En : Automatique

Spécialité : Automatique

Par : BENMANSOUR MALIK

Sujet : **Synthèse d'observateurs non-linéaires pour l'évaluation des compétences chirurgicales**

Soutenue publiquement le 12/09/2023 , devant le jury composé de :

Mme Benallel Mounira	Université de Tlemcen	MCA	Présidente
M Forestier Germain	Université de Haute-Alsace	Professeur	Examinateur
M Zemiti Cherif	Université de Montpellier	Maître de Conférence-HDR	Examinateur
M Malti Abed	Université de Tlemcen	MCA	Directeur
M Jannin Pierre	INSERM-Université de Rennes 1	DR-INSERM	Invité
M Ouejdi Salim	Université de Tlemcen	Professeur	Invité

À la mémoire de mon cher père,
dont le soutien, les sacrifices et les valeurs ont profondément marqué mon parcours.

À ma chère mère,
pour son soutien constant et ses encouragements tout au long de ce parcours.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Dr. ABED MALTI, for his trust in supervising this doctoral work. His rigorous scientific guidance, insightful advice, and constant support have been essential throughout this journey and have greatly contributed to the completion of this thesis.

My heartfelt thanks go to my parents, whose unconditional love, sacrifices, and unwavering support have been a constant source of strength. Their encouragement has been decisive in helping me pursue and accomplish this academic achievement.

I am deeply grateful to my colleagues Farah and Yasser, for their kindness, support, and stimulating discussions, which made this journey more enriching and less solitary.

I would also like to express my sincere appreciation to Pr. Pierre Jannin of the LTSI Laboratory, University of Rennes 1, for welcoming me during my research stay. This opportunity, made possible through his support and guidance, allowed me to enrich my work through a valuable combination of theoretical and applied research.

Finally, I would like to extend my appreciation to all those who, directly or indirectly, contributed to this work. Their support, in one way or another, has accompanied me throughout this doctoral journey.

Abstract

The outcomes of a patient's surgery are significantly influenced by a surgeon's technical skill level. The lack of safety during surgeries is prompting an ever-increasing demand for significant training enhancements in the surgical field. In fact, if a surgeon has poor technical skills during an operation, post-surgical complications may occur. Death may result from these complications. The most common cause of post-surgical complications, including re-operation and re-admission, is surgical technical errors. As a result, the need to improve the technical skill acquisition of surgeons through the development of novel and efficient automated methods is increasing from day to day.

In the recent years, machine learning and deep learning techniques have been applied to the assessment of surgical skills, with the goal of providing objective, accurate and efficient evaluations. These methods mostly involve the use of convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to analyze video and kinematic data collected during surgical procedures. By analyzing the patterns and movements of the surgical tools, the system can provide a quantitative assessment of the skill level of the surgeon. Several studies have shown promising results in the use of deep learning for surgical skills assessment, with potential applications in both surgical education and quality control.

In this thesis, we present a deep learning method based on a encoder-decoder architecture that acts as a nonlinear observer to provide an automated soft surgical skills evaluation based on a continuous scale. While this encoder-decoder architecture represents the main contribution, we also present other deep learning architectures based on Deep Feed Forward Neural Networks (DNNs), Recurrent Neural Networks (RNNs) and Long Short Term Memory Networks (LSTM). In all setups, we consider the surgical skill assessment task as a regression task. We trained and tested different models exclusively on surgical kinematic data recorded from surgeons with different expertise levels and compare the obtained regression results.

Keywords: Surgical skill assessment, deep learning, kinematic data, regression, encoder-decoder architecture, nonlinear observer.

Contents

List of Figures

Introduction

Technical skills in surgery [8] refer to the specialized abilities and manual dexterity required to perform surgical procedures effectively and safely. These skills encompass a wide range of techniques and competencies that surgeons develop through extensive training and practice. Technical skills in surgery involve precise manipulation of surgical instruments, suturing, tissue dissection, homeostasis (control of bleeding), and the ability to navigate and operate within the human body. The technical skill level of a surgeon has a significant effect on patient's surgical outcomes. The lack of safety during surgical operations is yielding from day to day to an increasing need of an important improvement in the surgery training. Indeed, post-surgical complications may occur if a surgeon delivers poor technical skills during an operation [9]. These complications may include death [10, 11]. Surgical technical errors are the most common reason for post-surgical complications including re-operation and re-admission [12, 13]. Consequently, inventing new effective approaches to improve surgeon's technical skill acquisition can positively affect the safety of the patient. The problem of poor or average acquired surgical skills may come from the ways followed by trainees so far to learn to perform surgical tasks: it consists on trying to replicate operations done by expert surgeons on cadavers or ex-vivo organs. The trainee first watches an expert surgeon performing a surgical task, then she/he tries her/his best to re-do the operation under the expert supervision [14, 15]. At the end, the trainee is assessed subjectively by the senior surgeon basing on many surgical criteria. Despite the fact that these surgical assessment methods are time and energy consuming for senior surgeons, they are also characterized by a certain subjectivity and a lack of accuracy. This subjectivity lies in the fact that every surgeon supervisor has her/his own appreciation of the quality of the operation. In an attempt to evaluate a trainee's skill level without using on an expert surgeon's judgement and thus increase objectivity of the process, objective structured assessment of technical skills (OSATS) was proposed and is currently adopted for clinical practice [16].

In this work, we present a novel approach for automated and objective surgical skill evaluation based on artificial neural networks that act as nonlinear observers which regresses (1) specific and personalized performance scores and (2) OSATS scores. These real valued scores deliver a strong feedback to the trainee according to several surgical criteria. The considered surgery type is robot-assisted surgery (RAS).

To enhance the surgical training programs, several High-Tech platforms have emerged, like surgical simulators based on virtual reality [17] or surgical systems like the DaVinci robot [18]. Until now, these systems do not include automatic surgical skills assessors software. Thus, several works on the topic have emerged that propose intelligent systems that can be included as a module in a surgical device to evaluate surgeons instantly and without any human intervention. For example, *Wang et al.* designed a deep CNN classify surgeons performance into three classes: novice, intermediate and expert [3]. The classification is made on the basis of DaVinci robot kinematic data that serve as input for the CNN. They also made an attempt for a scoring system based on a global score that reflect the global quality of the operation. However, the authors established arbitrary score thresholds to classify surgeons performance instead of grading it on a soft continuous scale. Another similar work made by *Hassan Ismail Fawaz et al.* [6] proposed a fully CNN for surgical skills classification. They also extended the FCNN to predict the OSATS scores. Interestingly, the latter extension to a regression model instead of a classification one enabled the authors to propose a technique that provides interpretability of the model's decision.

Most of the state-of-the-art methods rely on machine learning and deep learning models that have been trained with data from various surgical databases. Most of the works involving automated surgical skills assessment rely on classifying a surgeon's performance into discrete categories (e.g. novice, intermediate and expert) or into established scores thresholds. Also, many assessment methods based on artificial intelligence and especially on artificial neural networks were proposed to train surgeons and increase their surgical performance without any human intervention. These methods turned out to be very effective in classifying performance levels into discrete categories (for example: expert class, medium class and novice class). However, they do not deliver a strong feedback to the trainee since they classify performance into a certain category once the score reaches a certain threshold without taking into account intermediate scores. Consequently, the feedback suffers from a lack of accuracy and the trainee will not be able to have accurate information about her/his expertise level.

The intelligent system will allow senior surgeons who are currently in charge of the surgical training to save time and energy and focus on other surgical activities. Also, it helps improving objectivity and accuracy during a surgical performance evaluation. Concerning the literature, most of the previous works presented automated surgical skill assessment methods based on classification into discrete categories. There is a great lack of regression methods that helps to evaluate skills on a more efficient way that provide a more significant feedback to the surgeon. By this research, we contribute to reduce this deficit.

The research problem and questions can be stated as follows:

- Is there a method to reach high quality automated surgical evaluation that can be trusted in the surgical field?
- Does the method have the ability to score performance on a soft continuous scale instead of categorizing it?
- With the proposed database which contains surgical tasks performed on bench-top models only, can the trained models achieve high prediction performance ?
- The proposed database suffers from a lack of data samples. Is there an efficient data augmentation pipeline to overcome this problem and increase network generalizability ?

The objectives of this research work are as follows:

- Providing a new non-expensive and lightweight software that can be included in a surgical simulator to enhance surgical training.
- The proposed intelligent system has to be able to rate surgeons performance on a soft continuous scale in order to maximize feedback efficiency.
- We investigate the applicability of machine learning in analyzing motion signals of surgeons for objective evaluation of surgical skills.

In the chapter 1, we state the problem of the research. We show how can a neural network act as non linear observer to achieve the purpose of the surgical skill assessment on a continuous score scale. In the chapter 2, we present the domain literature. We talk about the previous works in the field of automated surgical skills assessment that relies on classification in discrete categories and other works that relies on regression on continuous score scales, using both video and kinematic data. In chapter 2, we present an analysis of surgical gesture flow based on kinematics. Also, we present the proposed database and explain every aspect of it, including the surgical tasks, the gestures, the time-series kinematic data. We also present the surgical criteria OSATS that serve as our basis for the automated assessment of surgical skills. In chapter four, we present the main contribution of this work which is a regression method of surgical skills assessment using time-series neural networks. In chapter five, we introduce another regression approach that consists in an encoder-decoder model that interprets (or describe) the input kinematic data by a sequence of OSATS scores.

To reach the aforementioned objectives, we propose several deep learning models to predict the surgeon's performance score on the basis of kinematic data that serve as input. The output score is on a continuous scale between 1 and 5 that reflects the quality of the work on one OSATS criteria. The models learn patterns from kinematic data of surgeons with various expertise levels (novice, medium and expert) that performed three basic tasks in surgery: knot-tying, needle-passing and suturing. First, we used a simple Deep Feed-forward Neural Network (DNN) for its implementation simplicity and for its low-computational demand. But since the inputs are time series data, we were convinced that using a Recurrent Neural Network (RNN) would be more effective because of its feedback connections. Indeed, the RNN yielded more accurate and plausible results but this type of neural network is subject to two considerable problems: the vanishing and exploding gradients. These problems occurs because the RNN is capable of capturing information in short time intervals and is unable to learn long-term dependencies. To overcome these problems, we employed a Long Short Term Memory Network which is an enhanced RNN and it is capable of capturing information widely spaced in time. The backward feedbacks allow the LSTM to take into account length-customizable previous entries. This specificity gives the LSTM an advantage over the RNN. In the context of surgery tasks, this enable us to take into account the evaluation of current gesture together with former executed gestures. At the end, results obtained with the LSTM surpassed all the previous results.Next, we went for Bidirectional LSTM which is a combination of two LSTM: One LSTM reads information in a forward way while the other one reads it in a backward way. This structure allows the networks to capture both past and future information about the sequence at every time step, providing an enhancement to the LSTM. After proposing some relatively small and lightweight models, we wanted to try another more complex approach involving capturing spatial information. Thus, we used a deep 2D Convolutional Neural Network (CNN) that is suitable for this task and it is well known for dealing with time-series. We propose two combined architectures of CNN-LSTM and CNN-BiLSTM to take advantage of the characteristics of both network types by capturing both spatial and temporal information from kinematic data. These last two architectures delivered the best results. Using the same architecture, we propose a dynamic observation of surgical skills with a neural observer model that has been trained online and offline.

Concerning the data, we relied on the JIGSAWS database [19] to make our experiments. It contains video and kinematic data of surgeons performing three basic tasks in surgery using the da Vinci surgical system. We used only kinematic data. We first employed raw kinematic data without performing any data pre-processing on the simple networks architectures (DNN,RNN and LSTM) and the obtained results were plausible but average. But when it came to more complex architectures (BiLSTM, CNN, CNN-LSTM and CNN-BiLSTM), generalization became more difficult for the network. Thus, the lack of kinematic data from the proposed database led us to perform a data augmentation pipeline in order to increase the number of samples. It consists on dividing a sequence of motion data into several sub-sequences along the time axis and along the variable axis. The networks hyperparameters are optimized in order to deliver prediction scores that correlate the most with ground truth scores that serve for testing the network generalizability.

We summarize our contributions as follows:

- We propose a novel deep neural network architecture to improve the accuracy and the objectivity of the current surgical skill evaluation program. The proposed model can analyze motion data of robotic end-effector tools to deliver instant feedback to the user.
- We propose a regression method that allows surgical skills rating on a soft continuous scale.
- We propose an encoder-decoder framework to reconstruct kinematic inputs and deliver regression scores outputs.
- We propose a encoder-decoder that acts like an nonlinear observer and allows a dynamic observation of surgical skills.

Chapter 1

Problem statement: Neural Networks as Nonlinear Observers for soft surgical skills assessment

1.1 Introduction

Artificial Neural networks are a type of machine learning algorithm that are widely used for a variety of tasks, including image and speech recognition, natural language processing, and autonomous systems. they have become increasingly popular in recent years. One key property of neural networks is their ability to function as nonlinear observers, which means that they are capable of transforming complex, nonlinear input data into more manageable, linear representations.

A nonlinear observer is a mathematical model that can estimate the state of a system based on input measurements. The observer uses a nonlinear function to map the input measurements to an estimate of the state. In the case of neural networks, the nonlinear function is represented by multiple layers of artificial neurons, which are connected through weighted connections. Nonlinear observers are important because many real-world systems are nonlinear in nature, and linear observers cannot always capture their behavior accurately. Neural networks, on the other hand, are able to learn the nonlinear relationships between inputs and outputs, and can thus provide more accurate predictions and representations of the data.

There are several types of neural networks that are commonly used as nonlinear observers, including feedforward networks, recurrent networks, and convolutional networks. Feedforward networks consist of multiple layers of artificial neurons that are connected in a feedforward manner, where the output of one layer is used as the input to the next layer. Recurrent networks, on the other hand, have connections that feed back into previous layers, allowing the network to maintain information over time. Convolutional networks are designed to process image data, and are particularly well-suited for tasks such as object recognition and image classification. The learning process in neural networks is based on optimization algorithms, such as gradient descent, which iteratively update the parameters of the network to minimize the error between the predicted outputs and the true outputs. This learning process allows the network to learn the nonlinear relationships between inputs and outputs, and thus function as a nonlinear observer.

Finally, neural networks are a powerful tool for nonlinear observation, and their ability to learn nonlinear relationships has made them a popular choice for a wide range of applications, including image and speech recognition, natural language processing, and autonomous systems. In the field of automated surgical skills assessment, neural networks can act as nonlinear observers to transform surgeon's surgical motion data into a set of scores that reflects the quality of the performance of the surgeon.

In this chapter, we show how a nonlinear observer can act like an artificial neural network. The combination of the two tools is called neural observers. We refer to the chapter 3 of this book [1] in order to present notions of neural observers for linear and nonlinear output cases. The considered systems are nonlinear discrete-time. At the end of the chapter, we present the neural observer problem for surgical skills assessment in the context of laparoscopic robot-assisted surgery. The considered states of the dynamic system are kinematic states variables from the JIGSAWS kinematic sequences [19] (see chapter 3).

1.2 Neural Observers for Discrete-Time Nonlinear Systems

In this section, the neural observer scheme is presented for unknown discrete-time multi-input and multi-output systems. First, we consider systems with linear output then systems with nonlinear output. To illustrate the applicability of the presented architectures, an application is illustrated: A neural observer for HIV model. But first, we define the notion of observers for dynamic systems in control theory.

1.2.1 Observers for dynamic systems

In control theory, an observer is a system that estimates the internal state of a dynamic system based on its input and output signals. Observers are commonly used when the internal state of the system cannot be measured directly, or when there are uncertainties in the system model. In the context of non-linear systems, observers are particularly useful because non-linear systems are often difficult to model and control. Mathematically, an observer for a non-linear system can be represented by a set of differential equations that estimate the internal state of the system based on its inputs and outputs. These differential equations are designed to minimize the difference between the estimated state and the actual state of the system over time. One popular type of observer for non-linear systems is the extended Kalman filter [20], which uses a non-linear model of the system and a Gaussian approximation to estimate the system state.

Consider a non-linear system described by the following set of equations:

$$\begin{aligned} x &= f(x, u) \\ y &= h(x) \end{aligned} \tag{1.1}$$

where $x \in \mathbb{R}^n$ is the system state, $u \in \mathbb{R}^m$ is the input, $y \in \mathbb{R}^p$ is the output, and f and h are non-linear functions.

An observer for this system can be represented by the following set of differential equations:

$$\begin{aligned} \hat{x} &= g(\hat{x}, u, y) \\ \hat{y} &= h(\hat{x}) \end{aligned} \tag{1.2}$$

where $\hat{x} \in \mathbb{R}^n$ is the estimated state, $\hat{y} \in \mathbb{R}^p$ is the estimated output, and g is a non-linear function that depends on the estimated state, input, and output. The observer is designed to estimate the internal state of the system based on its inputs and outputs, and is updated at each time step using the following update law:

$$\dot{\hat{x}} = g(\hat{x}, u, y) + L(y - \hat{y}) \tag{1.3}$$

where L is a feedback gain matrix that is designed to minimize the difference between the estimated state and the actual state of the system over time.

Observers for non-linear systems have a wide range of applications, including in robotics, aerospace engineering, and biomedical engineering. The use of observers for non-linear systems is an active area of research, with ongoing efforts to develop new and more effective observer designs for a variety of applications.

1.2.2 Nonlinear systems with linear output

In this section we assume reconstructing the state of a discrete-time nonlinear system with linear output. We consider estimating the state of a discrete-time nonlinear system 1.1, which is assumed to be observable, given by

$$\begin{aligned} x(k+1) &= F(x(k), u(k)) + d(k), \\ y(k) &= Cx(k) \end{aligned} \tag{1.4}$$

Where:

- $x \in \mathbb{R}^n$ is the state vector of the system.
- $u(k) \in \mathbb{R}^m$ is the input vector.
- $y(k) \in \mathbb{R}^p$ is the output vector.
- $C \in \mathbb{R}^{p \times n}$ is a known output matrix.
- $d(k) \in \mathbb{R}^n$ is a disturbance vector.
- $F(\bullet)$ is a smooth vector field with $f(\bullet)$ as entries.

For system 1.4, we propose a recurrent neural Luenberger type observer (or reccurent high order neural observer(RHONO)) with the following structure:

$$\begin{aligned} \hat{x}(k+1) &= W^{\intercal} z(\hat{x}(k), u(k)) + ge(k), \\ \hat{y}(k) &= C\hat{x}(k), \quad i = 1, \dots, n, \end{aligned} \tag{1.5}$$

with $g \in \mathbb{R}^p$, w and $z(x(k), u(k))$ being defined in the chapter 2 of [1].

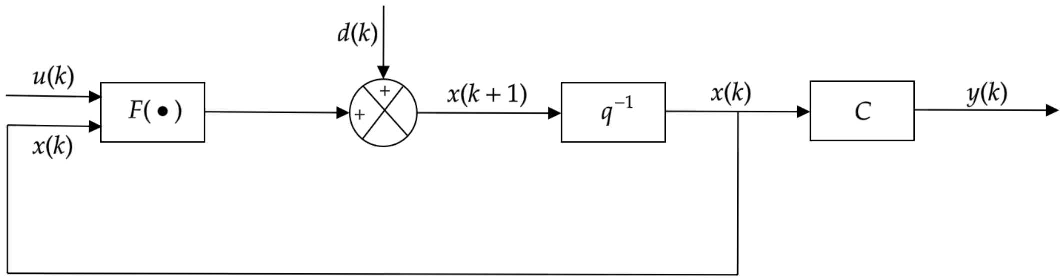

Figure 1.1: Schematic representation for a unknown discrete-time nonlinear system with linear output [1].

According to [21], the general discrete-time nonlinear system 1.4 can be approximated by a recurrent high order neural network (RHONN) using the following representation:

$$x(k+1) = W^{*\intercal} z(x(k), u(k)) + \epsilon_z, \tag{1.6}$$

Where x is the plant state vector and ϵ_z is a bounded approximation error, which can be reduced by increasing the number of the adjustable weights [21]. We name the desired weight vector as $W^* \in \mathbb{R}^L$, where L is the number of connections between layers of a neural network. It is the weight vector we want to reach such as the error $||\epsilon_z||$ reaches a minimum on a compact set $\Omega_z \subset \mathbb{R}^L$. W^* is constant but unknown. We define its estimate as W. Then the vectors of weights estimate $\tilde{W}(k)$ and the state observer $\tilde{x}(k)$ errors are respectively defined as:

$$\tilde{W}(k) = W(k) - W^* \tag{1.7}$$

and

$$\tilde{x}(k) = x(k) - \hat{x}(k). \tag{1.8}$$

Since W^* is constant, then

$$\tilde{W}(k+1) - \tilde{W}(k) = W(k+1) - W(k), \quad \forall k \in Z^+$$

The weight vectors are updated online with a decoupled Extended Kalman Filter (EKF) [20], described by

$$
\begin{aligned}
W(k+1) &= W(k) + \eta K(k)e(k), \\
K(k) &= P(k)H(k)M(k), \\
P(k+1) &= P(k) - K(k)H^{\intercal}(k)P(k) + Q(k),
\end{aligned} \tag{1.9}
$$

with

$$
M(k) = [R(k) + H^{\intercal}(k)P(k)H(k)]^{-1} \tag{1.10}
$$

The output error is defined by

$$
e(k) = y(k) - \hat{y}(k), \tag{1.11}
$$

and the state estimation error is defined by

$$
\tilde{x}(k) = x(k) - \hat{x}(k) \tag{1.12}
$$

Then the dynamics of 1.12 can be expressed as

$$
\tilde{x}(k+1) = x(k+1) - \hat{x}(k+1) \tag{1.13}
$$

Substituting (1.5) and (1.6) into (1.13) yields

$$
\tilde{x}(k+1) = \tilde{W}(k)z(\hat{x}(k), u(k)) + \epsilon_z' - gC\tilde{x}(k) \tag{1.14}
$$

with $\epsilon_z' = W^{*\intercal}z(\tilde{x}(k), u(k)) + \epsilon_z$ and $z(\tilde{x}(k), u(k)) = z(x(k), u(k)) - z(\hat{x}(k), u(k))$. On the other hand, the dynamics of (1.7) can be expressed as:

$$
\tilde{W}(k+1) = W(k+1) - W^* \tag{1.15}
$$

Using (1.9) in (1.28), it is possible to obtain

$$
\tilde{W}(k+1) = \tilde{W}(k) - \eta K(k)e(k) \tag{1.16}
$$

Figure 1.2 presents the proposed neural observer scheme.

Considering (1.9)-(1.29), the first main result of this section is established as the following theorem from chapter 3 of [1].

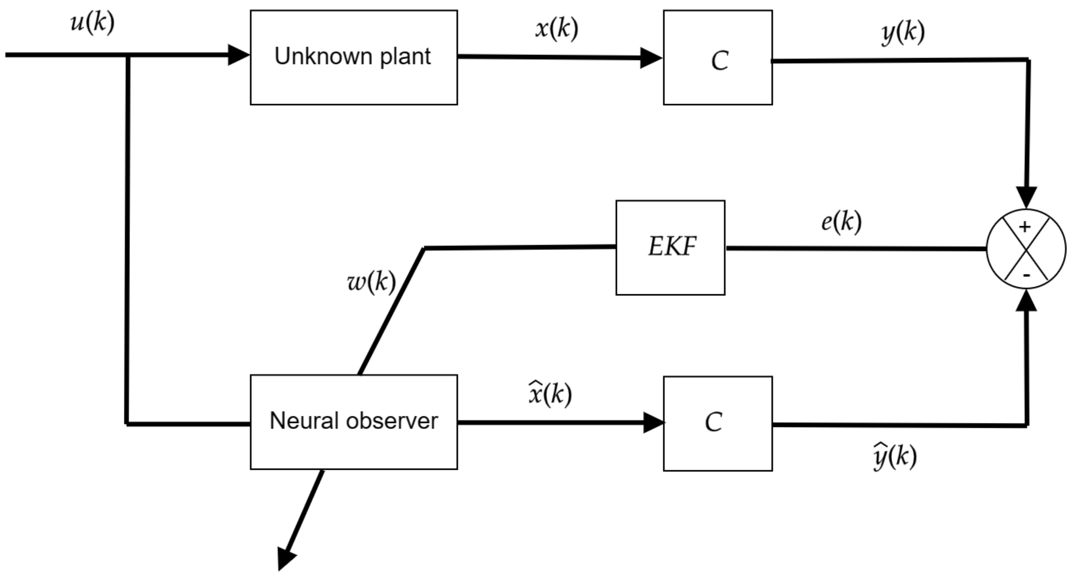

Figure 1.2: Neural observer scheme for linear output [1].

Theorem 1 *For system (1.4), the RHONO (1.5), trained with EKF-based algorithm (1.9), ensures that the ith (i=1,2,. . . ,n) estimation error (1.12 and the output error (1.11) are semi-globally uniformly ultimately bounded (SGUUB); moreover, the RHONO weights remain bounded.*

The theoretical proof of this theorem can be found in chapter 3 in [1]. We will demonstrate bounded error experimentally.

In our work presented in 5, we use the ADAM optimizer [22] to update the weights of the network instead of using a EKF. The EKF and the ADAM optimizer are two different mathematical algorithms used in the fields of control theory and machine learning, respectively. Although they serve different purposes, there are some similarities between them: Both the EKF and ADAM optimizer involve optimization of a function in order to reduce estimation error. Also, both the EKF and ADAM optimizer are iterative algorithms that perform updates to the estimated state or model parameters based on a set of measurements or observations.

1.2.3 Nonlinear systems with nonlinear output

In this section we assume reconstructing the state of a discrete-time nonlinear system with nonlinear output. We consider estimating the state of a discrete-time nonlinear system 1.3, which is assumed to be observable, given by

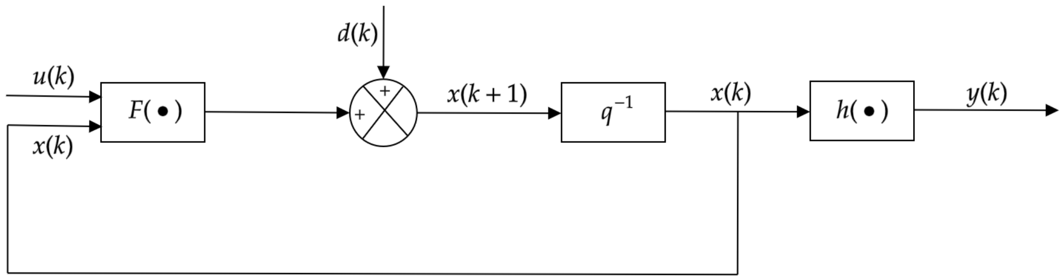

Figure 1.3: Schematic representation for a unknown discrete-time nonlinear system with nonlinear output [1]

$$\begin{aligned} x(k+1) &= F(x(k), u(k)) + d(k), \\ y(k) &= h(x(k)) \end{aligned} \tag{1.17}$$

Where:

- $x \in \mathbb{R}^n$ is the state vector of the system.
- $u(k) \in \mathbb{R}^m$ is the input vector.
- $y(k) \in \mathbb{R}^p$ is the output vector.
- $h(\bullet)$ is a known nonlinear output function which is Lipschitz in $x(k)$.
- $d(k) \in \mathbb{R}^n$ is a disturbance vector.
- $F(\bullet)$ is a smooth vector field with $f(\bullet)$ as entries.

For system 1.17, we propose a recurrent neural Luenberger type observer (RHONO) with the following structure:

$$\begin{aligned} \hat{x} &= [\hat{x_1}(k) \quad \dots \quad \hat{x_i}(k) \quad \dots \quad \hat{x_n}(k)]^\intercal, \\ \hat{x}(k+1) &= w^\intercal z(\hat{x}(k), u(k)) + ge(k), \\ \hat{y}(k) &= h(\hat{x}(k)), \end{aligned} \tag{1.18}$$

with $g_i \in \mathbb{R}^p$, w_i and $z_i(x(k), u(k))$ as defined (2.6) and (2.7), respectively.

The general discrete-time nonlinear system 1.17 can be approximated by a RHONN using the following representation:

$$x(k+1) = W^{*\intercal} z(x(k), u(k)) + \epsilon_z, \tag{1.19}$$

Variables are the described the same as in the linear output case. x is the plant state vector and ϵ_z is a bounded approximation error, which can be reduced by increasing the number of the adjustable weights. We name the desired weight vector as $W^* \in \mathbb{R}^L$. It is the weight vector we want to reach such as the error $||\epsilon_z||$ reaches a minimum on a compact set $\Omega_z \subset \mathbb{R}^L$. W^* is constant but unknown. We define its estimate as W. Then the vector of weights estimate $\tilde{W}(k)$ and the state observer $\tilde{x}(k)$ errors are respectively defined as:

$$\tilde{W}(k) = W(k) - W^* \tag{1.20}$$

and

$$\tilde{x}(k) = x(k) - \hat{x}(k). \tag{1.21}$$

Since W^* is constant, then

$$\tilde{W}(k+1) - \tilde{W}(k) = W(k+1) - W(k), \quad \forall k \in Z^+$$

The weight vectors are updated online with the same decoupled EKF as for the linear output, described by

$$\begin{aligned} W(k+1) &= W(k) + \eta K(k) e(k), \\ K(k) &= P(k) H(k) M(k), \\ P(k+1) &= P(k) - K(k) H^\intercal(k) P(k) + Q(k), \end{aligned} \tag{1.22}$$

with

$$M(k) = [R(k) + H^\intercal(k) P(k) H(k)]^{-1} \tag{1.23}$$

The output error is defined by

$$e(k) = y(k) - \hat{y}(k), \tag{1.24}$$

and the state estimation error is defined by

$$\tilde{x}(k) = x(k) - \hat{x}(k) \tag{1.25}$$

Then the dynamics of 1.25 can be expressed as

$$\tilde{x}(k+1) = x(k+1) - \hat{x}(k+1) \tag{1.26}$$

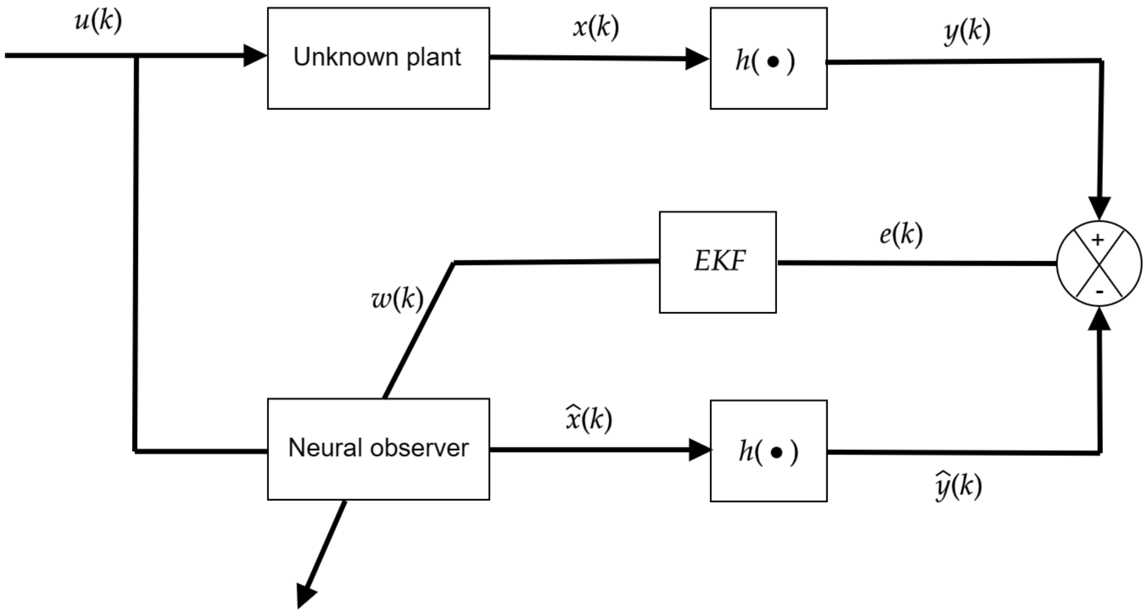

Figure 1.4: Neural observer scheme for nonlinear output [1]

Substituting (1.18) and (1.19) into (1.26) yields

$$\tilde{x}(k+1) = \tilde{W}(k)z(\hat{x}(k), u(k)) + \epsilon'_z - gh(\tilde{x}(k)) \tag{1.27}$$

with $\epsilon'_z = W^{*\intercal} z(\tilde{x}(k), u(k)) + \epsilon_z$ and $z(\tilde{x}(k), u(k)) = z(x(k), u(k)) - z(\hat{x}(k), u(k))$. On the other hand, the dynamics of (1.20) can be expressed as:

$$\tilde{W}(k+1) = W(k+1) - W^* \tag{1.28}$$

Using (1.9) in (1.28), it is possible to obtain

$$\tilde{W}(k+1) = \tilde{W}(k) - \eta K(k)e(k) \tag{1.29}$$

Figure 1.4 presents the proposed neural observer scheme for a nonlinear output.

Then, considering (1.17) and (1.18), it is possible to extend Theorem 1 for an unknown nonlinear system with nonlinear output as follows:

Theorem 2 *For system (1.17), the RHONO (1.18), trained with EKF-based algorithm (1.22), ensures that the ith (i=1,2,...,n) estimation error (1.25 and the output error (1.24) are* *semi-globally uniformly ultimately bounded (SGUUB); moreover, the RHONO weights remain bounded.*

The theoretical proof of this theorem can be found in the third chapter 3 in [1].

Considering our case, the state of the system is composed of kinematic data from surgical training and a soft evaluation score. Therefore, we consider the output of our system as linear.

1.3 Examples of application

The following sections show examples of application of neural observers on real-life scenarios. They have been taken from chapter 3 of [1].

1.3.1 Neural network observer for Human Immunodeficiency Virus (HIV) model

Biomedical systems are extremely expensive, challenging, and occasionally impossible to measure. In order to help doctors better comprehend the immunological markers in patients with viral infectious illnesses, state estimation by neural networks may be crucial. Since it can cause the acquired immunodeficiency syndrome, HIV has received the most attention among the various types of viral diseases over the past 30 years (AIDS). The World Health Organization (WHO) estimates that 33 million persons worldwide were HIV positive as of the end of 2007 in its global review of the AIDS epidemic. Around 2 million people died of AIDS in that same year. To combat the infection, there is, however, no recognized cure at this time. Due to the need for new techniques to give immune function markers in HIV patients, clinicians need additional information regarding the infection in patients. A burgeoning field has been observers in HIV infection. Most of these research have taken into account continuous time measurements, however these measurements are only made occasionally each year. In addition, one flaw in these systems is that the observer design must be modeled. As a result, we estimate a model for the system as well as the measurement of the state variables using the proposed RHONO in this application.

HIV Model

A neural observer is applied to a to a HIV model. The HIV model is composed of:

- The concentration of infected (T^i) and uninfected (T) cell.

- The number of viral cells in the blood stream V.

Using the Euler method, the HIV model can be represented in discrete time as:

$$\begin{array}{c} T(k+1) = T(k) + \delta s(k) - \delta \mu_T T(k) + \delta_r \frac{T(k)V(k)}{C+V(k)} - \delta \epsilon(k) k_v T(k)V(k), \\ T^i(k+1) = T^i(k) + \delta \epsilon(k) k_v T(k)V(k) - \delta \mu_T^i T^i(k) - \delta_r \frac{T^i(k)V(k)}{C+V(k)}, \\ V(k+1) = V(k) + \delta N r \frac{T^i(k)V(k)}{C+V(k)} - \delta k_T T(k)V(k) + \delta g_v \frac{V(k)}{C+V(k)}, \\ y(k) = h(T(k), T^i(k), V(k)) \end{array} \tag{1.30}$$

Where:

- s is the source of new CD4 cells produced by the Timo.
- μ_T is the death speed of the uninfected CD4.
- μ_T^i is the death speed of the infected cells.
- k_v is the speed of the uninfected cells becoming cells with virus.
- k_T is the speed of lymphocytes CD8 eliminating the virus.
- r is the maximum proliferation of the CD4 cells population.
- N is the number of free viruses produced by the infected cells.
- C is the semi-constant of the proliferation process.
- c is the total daily drug dosage in chemotherapy.
- b is the half-constant saturation value of an external source of virus.
- g_v is the level under which other cells (that are not lymphocytes) are free of the virus in the blood.
- δ is the sampling period.
- h is the output matrix, given by $h = [1 \quad 1 \quad 0]$.

RHONO for the HIV Model

In this section the discrete-time RHONO for unknown nonlinear systems with linear output is designed to estimate the model dynamics and state measurements for an HIV model.

The RHONO is applied to HIV dynamics, whose nonlinear dynamics is considered unknown. To estimate the state, which is defined by the number of viral cells in the blood torrent ($V = x_3$), the online concentration of infected cells ($T^i = x2$), and the online concentration of uninfected cells ($T = x1$). For this case T^i and T measurements are considered known, then $y(k) = [x1(k) \quad x2(k)]^\intercal$; we use a sampling time of 20 minutes. The neural dynamic used for this state estimation is given by:

$$\begin{gathered}
x_1(k+1) = w_{11}(k)S(\hat{x}_3(k)) + w_{12}(k)S(\hat{x}_1(k))^2S(\hat{x}_2(k)) + w_{13}(k)S(\hat{x}_1(k))S(\hat{x}_3(k))S(u(k)) \\
+w_{14}(k)S(\hat{x}_1(k))S(\hat{x}_2(k)) + g_1e(k), \\
\\
\hat{x}_2(k+1) = w_{21}(k)\hat{x}_1(k)\hat{x}_3(k) + w_{22}(k)\hat{x}_2(k) + w_{23}(k)S(\hat{x}_3(k))S(u(k))\hat{x}_2(k)\hat{x}_3(k) + g_2e(k), \\
\\
\hat{x}_3(k+1) = w_{31}(k)S(\hat{x}_2(k))S(\hat{x}_3(k))S(u(k)) + w_{32}(k)S(\hat{x}_1(k))S(\hat{x}_3(k)) \\
+w_{33}(k)S(\hat{x}_1(k))S(\hat{x}_3(k))^2 + g_2e(k) \\
\\
\hat{y}(k) = [\hat{x}_1(k) \quad \hat{x}_2(k)]^\intercal
\end{gathered} \tag{1.31}$$

where $\hat{x1}$, $\hat{x2}$, and $\hat{x3}$ are the estimates of concentration of uninfected cells (T), concentration of infected cells (T^i), and the number of viral cells in the blood stream (V), respectively. The input $u(k) = c$ is the total daily drug dosage in chemotherapy.

1.3.2 Other examples

In addition, we mention briefly some other examples of applicability of neural observers. First, *Lakhal et al.* [23] treated the topic of neural network observers for nonlinear systems. They proposed a neural observer with a three-layer feedforward neural network which is trained extensively with the error backpropagation learning algorithm. The neural observer is intended to reconstruct the unavailable state variables of an induction motor. Next, *Li et al.* [24] proposed a novel torque observer for interior permanent magnet synchronous machine drives based on a deep neural network. The proposed observer was built without the use of motor model and parameters. Finally, *Ghosn et al.* [25] compared several neural networks architectures that act like observers and compared them to the classical EKF. They evaluated their method on a kinematic bicycle mode.

1.4 Neural observers for surgical skills assessment

Now that the neural observer notion has been presented and its applicability illustrated with a real life scenario problem, in this section, we discuss its utility in the field of automated surgical skills assessment. Automated surgical skills assessment is a promising area of research that has the potential to improve the quality and efficiency of surgical training. However, one challenge that researchers face is the lack of a physical model in some surgical procedures, such as endovascular interventions. In our work, where surgical data are acquired from the da Vinci surgical system (see chapter 3), the physical model of the latter is too complex to establish. Indeed, The da Vinci robot is a complex system that uses a combination of advanced hardware and software to provide surgeons with precision control and enhanced visualization during minimally invasive surgeries. The robot's physical model includes multiple articulated arms and specialized surgical instruments, all of which are tightly integrated with the robot's software control system (the robot is described in chapter 3). The complexity has made it challenging for researchers to develop automated surgical skills assessment tools that rely on this technology. Moreover, the da Vinci surgical system is still receiving improvements and modifications each year [26]. Thus, its physical model is still undergoing several changes. In these cases, automated assessment tools must rely on other sources of information, such as video data or sensor readings, to assess the surgeon's skills. Fortunately, such information is available: There are publicly available surgical datasets containing kinematic and video data from surgeons with various expertise level that have performed several basic surgical tasks. Moreover, the data is annotated with performance scores on the basis of surgical criteria. We present the dataset (JIGSAWS) that we used for experiment in chapter 3. With the complexity of systems physical models and the availability of annotated surgical datasets, one approach to automated surgical skills assessment is the use of deep neural networks as nonlinear observers. Figure 1.5 depicts the diagram of a discrete-time nonlinear system connected to a neural observer for surgical skills assessment.At each instant k, the input $u(k)$ represents the user's input while manipulating the da Vinci tools, creating motion of instruments that is recorded as a kinematic sequence with the da Vinci software control system.

Afterthat, the kinematic sequence $u(k)$ goes through a hypothetical surgical skills evaluation model and produces a state variable $x(k)$ containing the surgical performance score that is not measurable. However, this score is observed in the output $y(k)$ through the observation matrix. $y(k)$ may contain only the perforamnce score or only the kinematic sequence at the last time step, depending of the dynamic evaluation approach that is followed (offline/offline or offline/online, see chapter 5). In parallel, the kinematic sequence $u(k)$ goes through a neural observer that provides estimates $\hat{x}(k)$ of states $x(k)$ and thus, the estimate $\hat{y}(k)$ of the output $y(k)$. The output estimation error $e(k)$ is computed and minimized through a training algorithm that delivers the necessary changes $w(k)$ that are the weights updates of the neural observer. The process is repeated until $e(k)$ reaches a minimum, meaning that the neural observer succeeds at estimating adequate surgical performance scores and kinematic variables states.

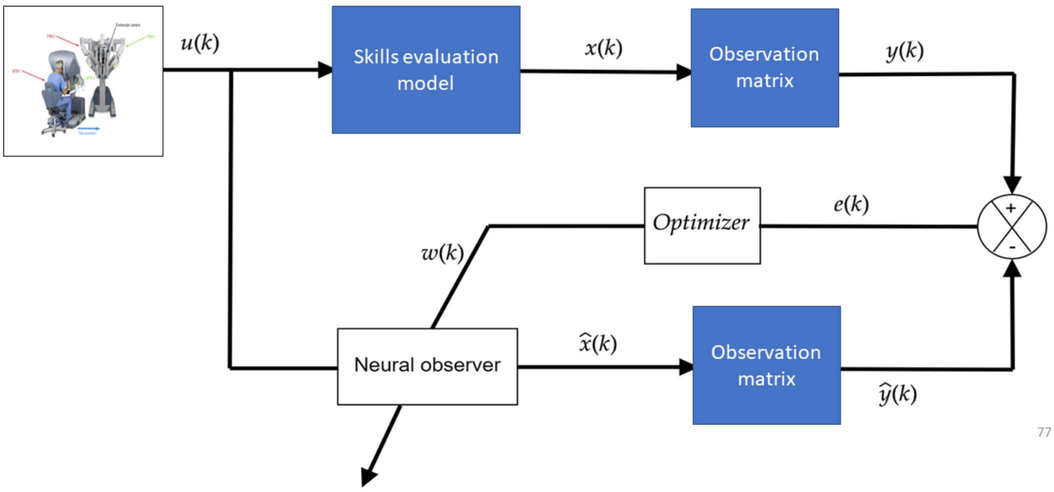

Figure 1.5: An illustration of dynamic system related to a neural observer for automated surgical skills assessment.

1.5 Conclusion

In this chapter, we have described the notion of observers and neural observers and shown an example of their applicability in real-life problems. In addition, we discussed how a neural observer can be applied for automated surgical skills assessment and pointed out why such methods are mandatory when the physical model of the used system is too complex to establish because of the constant changes made in the said system. Therefore, we can conclude that neural networks as nonlinear observers are well-suited for modeling complex and non-linear systems. By learning the nonlinear relationships between inputs and outputs, neural networks can provide accurate representations of nonlinear systems, even in cases where the underlying relationships are complex or not well understood. In our work presented in chapter 5, we propose two approaches for a dynamic observation of surgical skills: (1) an offline/offline for training and testing the neural observers and (2) an offline/online approach in order to match the neural observer model depicted in figures 1.4 and 1.5. Like shown in figures, we replace the EKF by an ADAM optimizer algorithm in order to update weights of the neural observer and to minimize the prediction error.

Chapter 2

State of the art

2.1 Introduction

Since the advent of powerful computers and machine learning programming libraries, machine learning has found widespread application in the fields of surgical skill assessment and surgical gesture recognition over the past ten years. Using a wide range of approaches, including Hidden Markov Models (HMM), Support Vector Machines (SVM), and Artificial Neural Networks (ANN), numerous works have been produced for the purpose of assessing surgical skills. The ANN is currently the most effective machine learning technique, as demonstrated by a number of studies and their findings. Particularly Convolutional Neural Networks (CNNs), Long-Short Term Memory (LSTM), Deep Feed-forward Neural Networks (DDNs), and Recurrent Neural Networks (RNNs) and their derivatives and variants. While the majority of the published works concentrate on the classification of surgical skills and gesture recognition, a few employed regression techniques.

We consider classification-based methods and regression-based methods as two distinct surgical skills assessment approaches and divide the literature in two sections, one for each approach. On the one hand, classification-based contributions deal with skill level categorization by establishing algorithms that finds functions that help divide a given dataset into classes (expertise levels in the case of surgical datasets) based on various parameters. Therefore, classification algorithms estimate discrete surgical skills levels and do not take into account in-between levels that might exist among assessed subjects. On the other hand, regression-based contributions employ methods that finds correlations between dependent and independent variables by finding functions that map an input variable x to a continuous output variable y. Unlike classification algorithms, regression algorithms help predict continuous variables such as OSATS scores and GEARS scores (see chapter 3). Thus, the assessment of surgical skills is not limited to assign subjects performance to categories and can provide a more relevant feedback based on continuous rating scales. In regard to these significant differences, we believe there is a clear boundary between the two approaches.

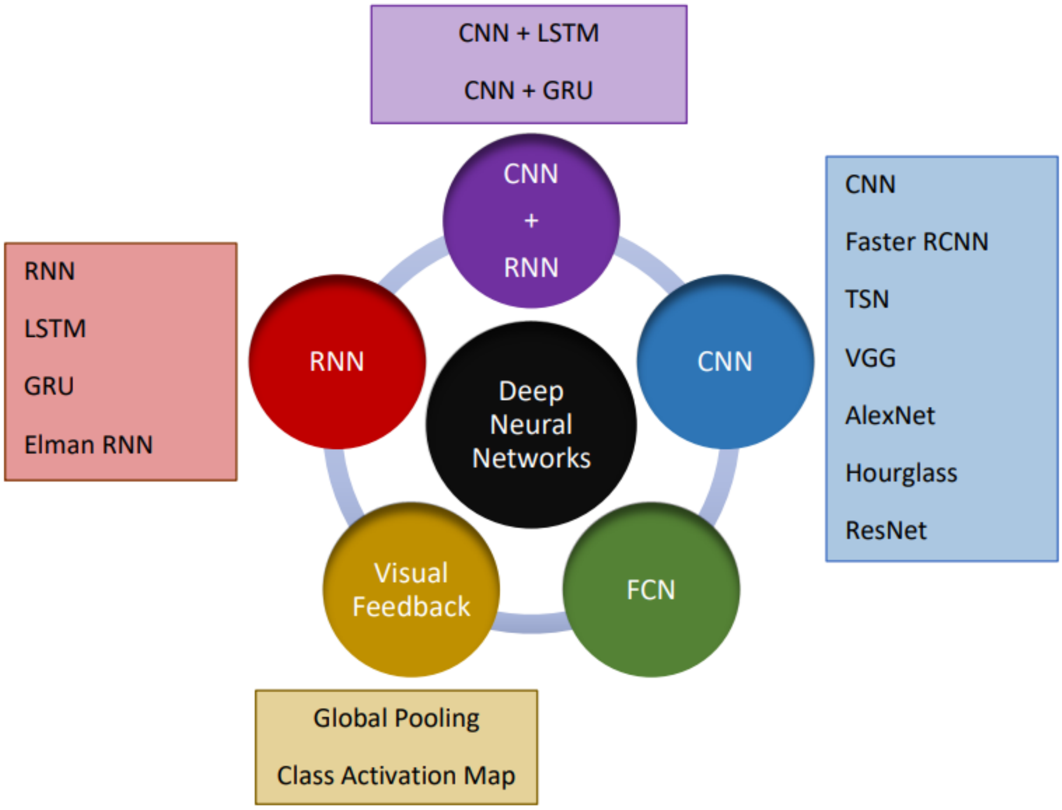

Figure 2.1: DNNs employed for surgical skill assessment. Circles other than the yellow represent the architectures, namely convolutional neural networks (CNN), recurrent neural networks (RNN), fully-connected networks (FCN), and CNN-RNN variates, whereas the yellow circle is for visualization. The color-coded boxes illustrate the specific frameworks used [2]

Concerning the data, kinematic and video-based data are two of the most common types of information gathered for surgical skill assessment. The data collection is not done in the operating room but during training sessions like Pelvi-trainer platforms [27] with in-silico, ex-vivo or virtual reality exercises. To collect motion-related data, sensors are placed at the trainee's tooltips, hands, or other body parts to collect kinematic data. Using these sensors, data for features like surgical tools location, velocity, and rotation can be directly collected. Due to the sensor's potential to obstruct trainee movement, this method of data collection is intrusive. Some robotic surgical systems, like the DaVinci robot [28], propose built-in applications to record kinematic data that capture positions, velocities and rotation of the robot manipulators. For virtual training purposes, the kinematic data can also be collected virtually without the limitations of physical sensors with software like EndoVis [29]. Video-based data, in contrast to kinematic data, is collected in a non-invasive manner. However, significant post-processing is required to extract useful information from video-based data. The most common used dataset by the gathered studies is the JIGSAWS dataset [19] (see chapter 3).

Figure 2.1, taken from a deep learning methods survey proposed by *Yanik et al* [2], shows the different DNN types employed for surgical skill assessment in the literature. Figure 2.2 represent the general framework for the surgical skills assessment process: A variety of machine learning (ML) algorithms are fed kinematic or video data from various surgical tasks performed in a variety of settings. A trained model is developed as a result. These models can then be fed new test data to give an evaluation of surgical skill. In the sections that follow, we will present a few of these earlier classification-based and regression-based works in the field of Computer Assisted Surgery (CAS) that made use of the aforementioned machine learning techniques and data types.

2.2 Classification-based methods

Human Activity Recognition (HAR) and Surgical Gesture Recognition (SGR) are two fields of study that are concerned with understanding and classifying human movements. HAR involves the identification and characterization of human activities based on sensor data from devices such as accelerometers and gyroscopes. This technology is used in various applications such as health monitoring, sports training, and security surveillance. On the other hand, SGR focuses on the recognition and classification of surgical gestures performed by a surgeon during a medical procedure. The aim is to provide real-time feedback to the surgeon and improve surgical training. Despite the differences in their applications, HAR and SGR share several similarities. Both fields require the analysis of complex motion data, often captured in noisy and unstructured environments. The processing of this data requires the use of sophisticated algorithms and machine learning techniques. In HAR, these algorithms are used to recognize activities such as walking, running, and cycling. In SGR, they are used to recognize gestures such as cutting, suturing, and dissecting. In both cases, the recognition of these activities or gestures is critical for providing useful feedback to the user and improving their performance.

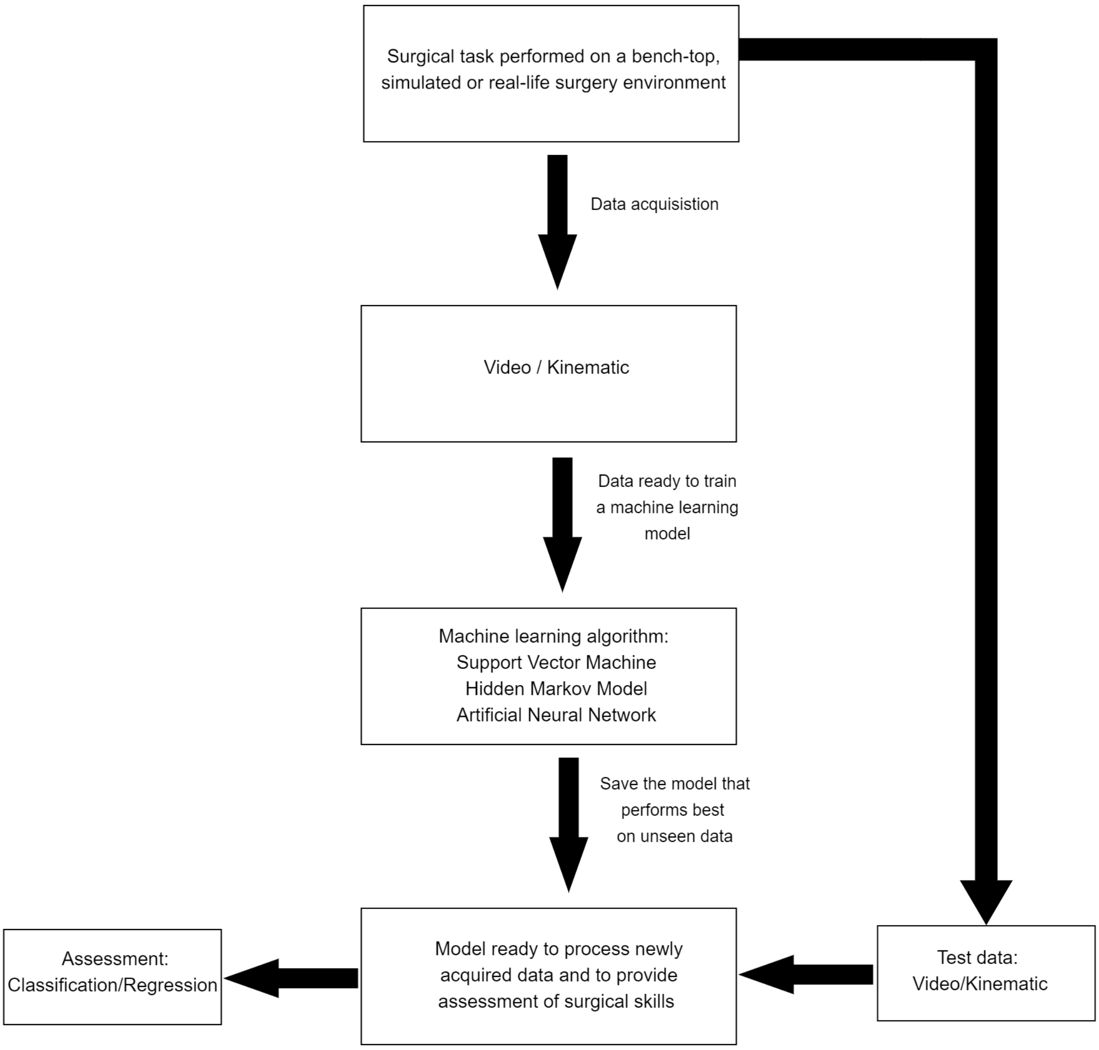

Figure 2.2: General framework for the surgical skills assessment process.

2.2.1 Using video data as inputs

On the one hand, *Andru P. Twinanda et al.*[30] proposed a method for surgical phase recognition that uses a CNN called EndoNet to automatically learn features from cholecystectomy videos. In a similar work, *Colin Lea et al.*[4] proposed a segmentation and an surgical action recognition method using a Spatiotemporal CNN. Figure 2.6 depicts a diagram of their model which contains three components: spatial, segmental and temporal components. Although the authors of these two contributions have shown effectiveness of their methods in phase recognition and fine-grained action segmentation, surgical skill evaluation is not involved, while we solely focus on the latter. Another work involving surgical phase recognition has been made by *Manish Sahu et al.*[31]. They proposed a pre-trained CNN with a transfer learning method for surgical workflow detection and estimation. *Dandan Zhang et al.*[32] developed A bidirectional multi-layer independently RNN for surgical gesture recognition combined with a Deep Convolutional Neural Network (DCNN) model based on the VGG architecture [33] (whose architecture is shown in figure 2.5) for spatial feature extraction from surgical video frames . The authors used the JIGSAWS dataset to validate their method that covers the surgical gesture recognition. Finally, *Lingling Tao et al.*[34] presented six techniques based on a temporal approach for segmentation and recognition of gestures in robotic surgery including Hidden Markov Models and Semi HHMs. All these works involves surgical gestures recognition or action segmentation but do not propose a surgical skill assessment method.

On the other hand, *Hei Law et al.*[35] proposed an automated method for surgical skill evaluation by tracking surgical instruments using a Hourglass Network. The evaluation relies on the five GEARS criterion [36]. However, surgeons are classified by adopting scores thresholds. *Amy Jin et al.*[37] introduced an automatic surgical skill assessment approach based on tool tracking and analyzing tool movements in surgical videos, using region-based convolutional neural networks. This method was the first to not only detect presence but also spatially localize surgical tools in real-world laparoscopic surgical videos. *Isabel Funke et al.*[38] proposed a 3D convolutional neural networks have to classify snippets (a batch of few consecutive frames extracted from surgical video) from the JIGSAWS dataset into three expertise classes.

2.2.2 Using kinematic data as inputs

Robert Dipietro et al.[39] proposed a work on recognizing surgical activities from robot kinematic data using LSTM and BiLSTM. The same first authors developed an architecture based on unsupervised learning that combines an RNN encoder-decoder and mixture density networks (MDNs) to model the conditional distribution over future motion given past motion, showing the possibility to learn meaningful representations of surgical skill motion, without supervision, by learning to predict the future.

In another work, *Robert DiPietro et al.*[40] made a comparison between four RNN architectures (simple RNNs, long short-term memory, gated recurrent units, and mixed history RNNs) for the purpose of recognizing surgical activities from kinematic data. In [41], *Ilker Gurcan et al.* claimed that RNNs and LSTM are not effective in capturing the relationship

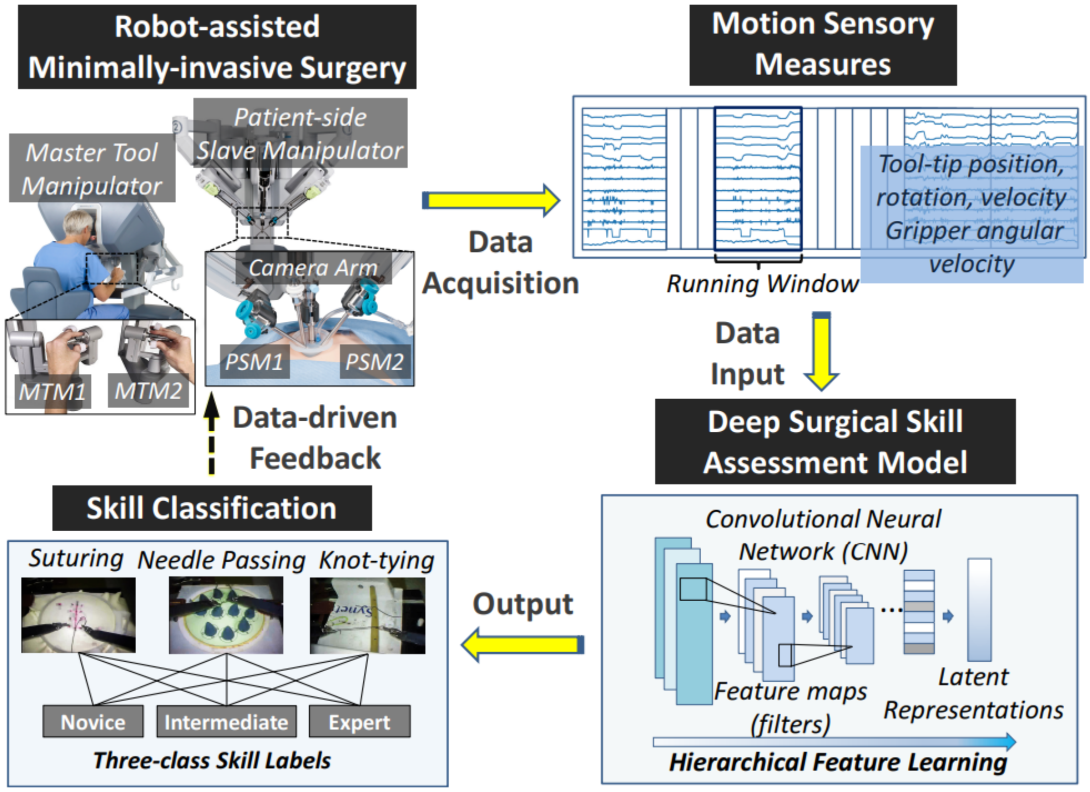

Figure 2.3: Diagram of the framework proposed by *Wang et al.*[3]

of features with different temporal scales, leading to sub-optimal recognition performance of surgical activities containing complex motions at multiple time scales. Thus, a Multiscale recurrent neural networks (MS-RNN) that combines the strength of both wavelet scattering operations and LSTM has been developed for the purpose of surgical activities recognition. *Laura Sbernini et al.*[42] proposed a work that focuses on manual dexterity by considering it as one of the most important surgical skills. A system is designed to track surgeon's hand movements during simulated open surgery tasks. Then, their manual expertise is evaluated using and an artificial neural network (ANN) classifier. Although the ANN achieved very good results, we believe that the inclusion of recurrent neural network or a CNN would yield better results than a simple ANN. Another work on surgical gesture recognition applied a Time Delay Neural Network (TDNN) to JIGSAWS kinematic data to introduce temporal modeling in gesture recognition [43]. [44] proposed a multi-class SVM for surgical gestures dictionary learning and classification which can be used to effectively analyze complex surgical gestures recorded by the Da Vinci robotic surgical system. *Filippo Cavallo et al.* [45] developed and evaluated a machine learning platform for the construction of a holistic biomechanical model of the surgeon and of the instruments used for minimally invasive surgery that uses a Markov chain for gesture analysis. The machine learning method is able to recognize expertise level of surgeons but only after setting an expert surgeon performance as a reference. An interesting work done by *Wang et al.*[3] involves an in-real-time surgical skill assessment method by building a deep CNN that can reliably interpret skills within 1-3 seconds window. Figure 2.3 and 2.4 depict their proposed general framework for the window-time-based surgical skill assessment and the proposed architecture of the CNN, respectively.

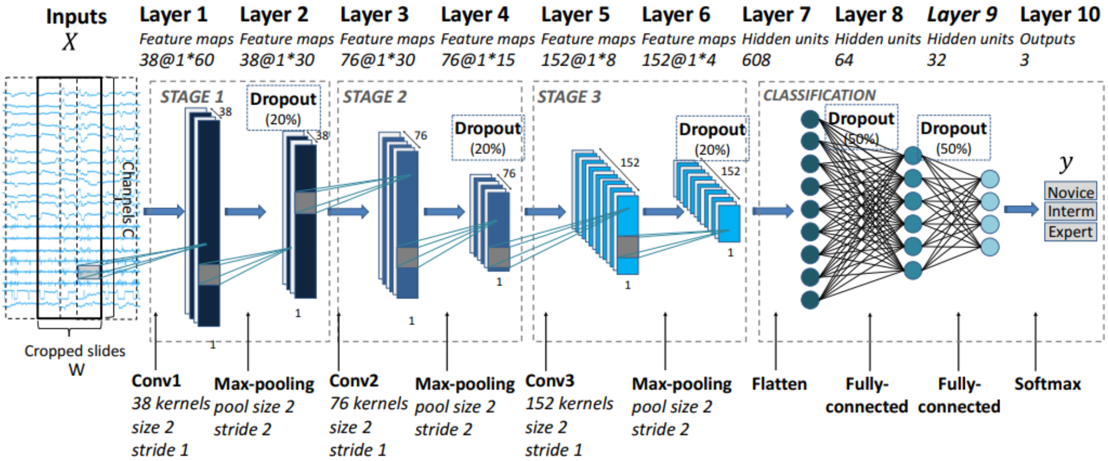

Figure 2.4: Architecture of the CNN proposed by *Wang et al.* [3]

The same authors presented a combined architecture of a CNN and a Gated Recurrent Unit (GRU) for online trainee skill analysis and task recognition [46]. The CNN component learns spatial abstract representations within the interval of input frame while the GRU component learns the temporal dynamics of multiple channels at each time step in raw motion data. Consequently, the model can characterize the nature of surgery motion relative to both the surgeon level of expertise and operation task. Although these works contribute to both surgical skill classification and task recognition domains, they still do not propose a robust method to truly grade a surgical performance by giving an exact score, instead of categorizing it. *Wang et al.*[3] trained their CNN to assess surgeons using the Global Rating Score (GRS) of the JIGSAWS dataset, but they did not set the original GRS outputs for a regression approach and they opted instead for a GRS classfication approach by establishing scores thresholds.

Jason D Kelly et al.[47], took advantage of bidirectional LSTM that is able to read input time series data in a forward direction and in a backward direction as well, in order to classify surgeon expertise level from the Basic Laparoscopic Urologic Skills dataset (BLUS [48]). This work has proven the effectiveness of the BiLSTM in surgeon level categorization. Hence, we

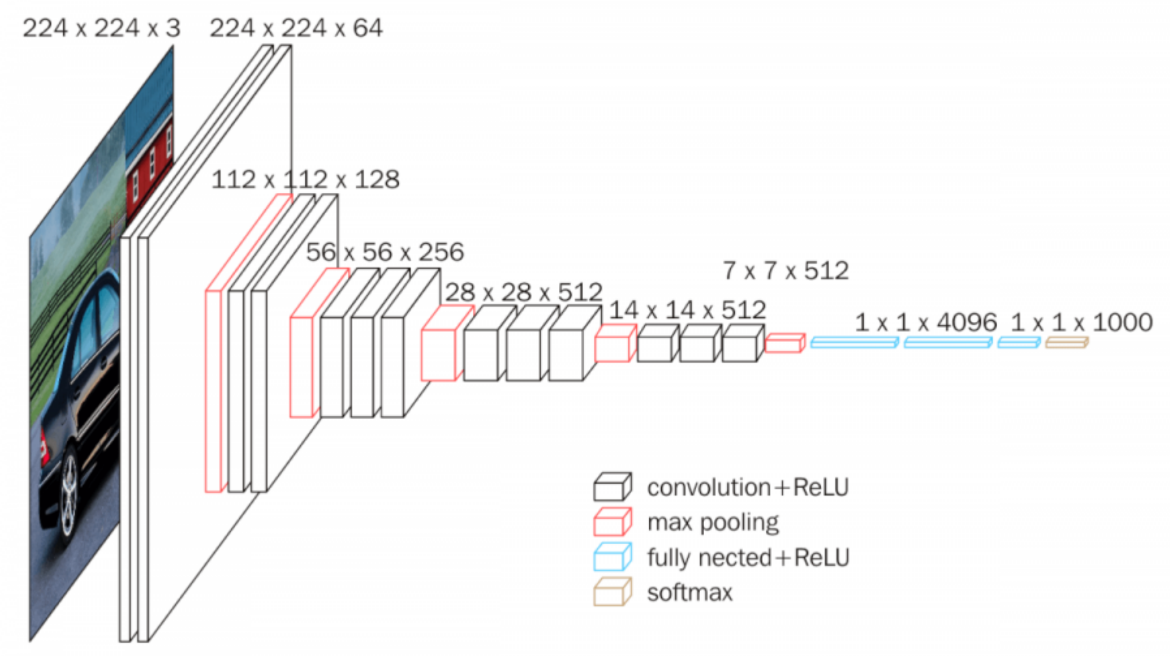

Figure 2.5: Architecture of the VGG convolutional neural network.

decided to include a BiLSTM block in our model architecture but for the purpose of predicting the exact surgical performance score. However, in our model, the BiLSTM component does not learn patterns from the input kinematic data directly but rather from latent states that have been encoded by a CNN component onto a latent space. *Piyush Poddar et al.* [49] provided a descriptive structure for nasal septoplasty by automatically segmenting it into higher-level meaningful activities called strokes. The sequence of strokes has been used to train a SVM classifier to distinguish between novice and expert surgeons. However, the best classification accuracy they obtained is 73%, which remains pretty low for a classification task and there is no real-valued score for an exact performance evaluation. Another work [50] using SVMs combined to a logistic regression method for robotic minimally invasive surgery skill assessment provided an evaluation method on six important movement features, leading to a strong surgical performance evaluation scheme.

2.3 Regression-based methods

Classification and regression are both important techniques in the field of surgical skill assessment. Classification involves the identification and recognition of specific parts of the surgical flow, while regression focuses on the evaluation of overall surgical skill and the identification of factors that contribute to successful outcomes. One advantage of classification

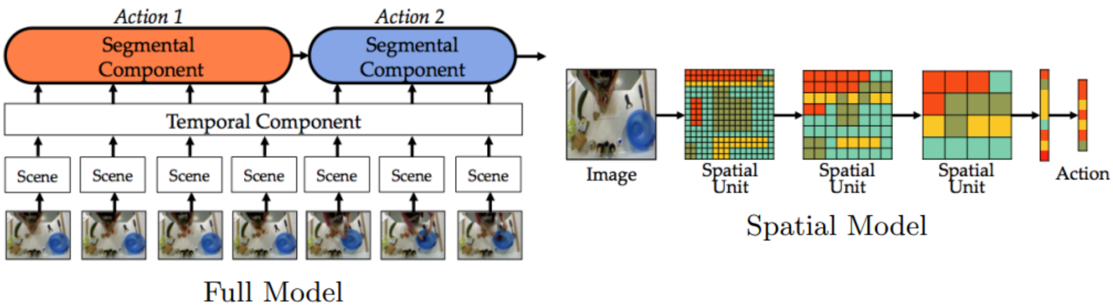

Figure 2.6: Model proposed in [4]. It contains three components. The spatial, temporal, and segmental units encode object relationships, how those relationships change, and how actions transition from one to another. (right) The spatial component of the model.

is that it allows for the identification of specific areas in the surgical flow that may need improvement, such as identifying where a surgeon may be struggling with a particular step. This can be useful in targeted interventions to improve a surgeon's performance. However, one potential disadvantage is that it may not provide a comprehensive evaluation of overall surgical skill, as it focuses on individual steps rather than the overall technique. Regression analysis, on the other hand, can provide a more comprehensive evaluation of surgical skill, as it considers multiple factors that contribute to successful surgical outcomes. This can lead to a more accurate assessment of a surgeon's overall skill level and help to identify areas for improvement. One potential disadvantage of regression analysis is that it may require larger datasets to be effective, which may not always be available in surgical training settings. Overall, both classification and regression analysis have their advantages and disadvantages in surgical skill assessment. However, regression analysis can bring more precision to identifying specific areas for improvement in a surgical gesture, leading to more targeted interventions and ultimately improved surgical outcomes.

2.3.1 Using video data as inputs

Shuja Khalid et al.[5] proposed a BiLSTM autoencoder to assess surgical video clips by categorizing them based on the surgical step being performed and the level of the surgeon's competence. The authors worked on both classification of skill level and regression using OSATS scores [16]. They used BiLSTM with an attention mechanism relying on video data. Figure 2.7 represents a diagram of their end-to-end autoencoder model. While most of the prior works focused on surgical skill assessment using simulated datasets, *Daochang Liu et al.*[51] used a real clinical dataset which consists of in-vivo laparoscopic surgeries. An objective and automated framework based on neural network (Multi Layer Perceptron) is proposed to predict surgical skills using a regression method achieving a Spearman's correlation of 0.55 with the ground truth of overall technical skill.

2.3.2 Using kinematic data as inputs

Karl-Friedrich Kowalewski et al.[52] presented a contribution that compares several machine learning algorithms (decision forest, neural networks, boosted decision tree) for skill classification, phase recognition but also OSATS scores predictions on continuous scales. In [53], three machine learning algorithms have been trained, involving SVMs in order to evaluate surgical performance and predict clinical outcomes of patients who have had a robot-assisted radical prostatectomy. Finally, *Hassan Ismail Fawaz et al.*[6] proposed a 1D convolutional neural network to assess surgeons directly from the kinematic data of the JIGSAWS dataset.

The framework is composed of a Fully Convolutional Network (FCN) and a Class Activation Map component that highlights the fractions of the surgical kinematic trial that contributed highly to classify the surgeon as novice, intermediate or expert. In addition, the authors proposed a regression model to predict OSATS scores [54, 16] on a continuous scale. Figure 2.8 shows the architecture of their proposed FCN. The authors did not apply a data augmentation pipeline. Thus, their model is trained with raw kinematic data with variable length. In addition, they have used a 1D CNN with convolutional filters sliding only over the temporal axis of the kinematic data sample. Similarly, *Zia et al.* [7] proposed a method that relies only on robot kinematic data to investigate the utilization of various holistic features for automated skill assessment and proposed a weighted feature fusion method for enhancing score prediction performance. These holistic features are Sequential Motion Texture (SMT), Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT) and Approximate Entropy (ApEn). Experiments are performed with JIGSAWS kinematic data for skill classification and exact OSATS score prediction. Figure 2.9 depicts their proposed framework for robotic surgical skills assessment.

Best results were obtained using a combination of these 4 features. Figure 2.9 the general framework of their proposed method. Regarding our previous works based on regression, we used several types of neural networks and compared to their results.

2.4 Positioning our work

In [55], we used a Deep feed-forward Neural Network (DNN) architecture to assess all surgeons of the JIGSAWS dataset on the three tasks separately, using real performance scores as a ground truth. Despite the good results obtained with this approach, we knew that results can be greatly improved by changing the type of the utilized neural network. Since the kinematic data that we are dealing with represent dynamic sequences of kinematic variables that depend on time (time series), then we wanted to incorporate the notion of time in our approach. A suitable type of neural network that seemed to be effective for such type of data is the recurrent neural network (RNN). Therefore, in [56], we proposed to use a RNN architecture to provide a dynamic evaluation of the performed surgery tasks. RNN are well known for their capability to allow feedback connections between states at different time steps. Indeed, the results obtained using the same amount of data samples for the training and the test steps as in [55] are more plausible and match more accurately our expectations. However, RNN are known to be vulnerable to two major problems possibly encountered during the training step: the vanishing gradient and the exploding gradient [57]. Indeed, long term dependencies in time series data may cause these problems. Besides, a kinematic data of a surgical task is composed of gestures (or sub-tasks) as it can be checked in [19] and these gestures are largely spaced over time in a kinematic data sample. So, with the lack of memory, a RNN is not capable to learn correlations between largely time-spaced events in a data sample. Consequently, we used for our third work [58] a Long Short Term Memory (LSTM) Network architecture which represents an evolution of a traditional RNN and it is suited to deal with time series data and to overcome the problems encountered with the RNNs. The LSTMs, indeed, have feedback connections and in addition, a notion of memory allowing it to learn long term dependencies. These backward feedback allow the LSTM to take into account length-customizable previous entries. This specificity gives LSTMs advantages over the RNNs and in the context of surgery tasks, this enable us to take into account the evaluation of current gesture together with former executed gesture. The results obtained using this architecture are a bit different from those obtained with classic RNNs but seem more logical since the whole surgical task is considered, as explained previously. Finally, we propose a combination of a CNN and a BiLSTM architectures [59] that behaves like an encoder-decoder to assess surgical skills from JIGSAWS kinematic data on a continuous scale based on each of the six OSATS criteria. The kinematic is first fed to the CNN with across three convolution stages and is encoded into a latent space. Then, it is decoded by the BiLSTM to a soft OSATS score reflecting the quality of the surgical task on the selected OSATS criteria. Our previous contributions are described in detail in chapters 4 and 5.

2.5 Discussion

Tables 2.1, 2.2 and 2.3 present a summary for this literature chapter. We have selected the following important criteria to compare the experimental setups established by the presented studies:

- The utilized machine learning method.
- The data type used for training and testing the machine learning model whether is kinematic or video.
- the considered task whether is a regression task based on output real scores or a classification task based on output categories.
- The adopted validation scheme that plays an important role in the degree of reliability of the trained model.
- The dataset that served as experiment ground.

First of all, we notice that video and kinematic data are the only two types of data utilized in these contributions, with the kinematic data being largely more used than the video data. The reason that might be behind this is that sensors are able record hand and tools motion directly, as opposed to video-based data and also, kinematic data can be fed rawly to machine learning models. In the contrary, an additional step of pre-processing is required to locate tools or hands in video-based data by performing image segmentation.

Finally, we notice that the majority of contributions consider surgical skill assessment as a classification task. Among the gathered studies, 20 involves classifying surgeons into expertise categories and only 11 proposed regression on scores established for performance grading like OSATS (see figure 2.10). Regression, in contrast to classification, requires a sophisticated ground-truth scoring system to annotate trials of surgeons on a continuous scale for a given surgical task , which accounts for the limited regression studies. However, these scoring systems are limited and may not be accessible to the general public.

Author (Year)	Machine learning method	Data type	Task
Law *et al.* (2017) [35]	Hourgalss Network (CNN) + SVM	Video	Regression (GEARS)
Funke *et al.* (2019) [38]	3DCNN	Video	Classification (N,I,E)
Doughty *et al.* (2018) [60]	Two streams CNN	Video	Classification (N,I,E)
Fawaz *et al.* (2019) [6]	FCN	Kinematic	Classification (N,I,E) & regression (OSATS)
Wang & Fey (2020) [3]	CNN	Kinematic	Classification (N,I,E,GRS)
Wang & Fey (2020) [46]	CNN-GRU	Kinematic	Classification (N,I,E)
Benmansour *et al.* (2018) [55]	DNN	Kinematic	Regression on real scores
Kelly *et al.* (2020) [47]	BiLSTM	Kinematic & video	Classification (N,E)
Ahmidi *et al.* (2015) [61]	SVM	Kinematic	Classification (N,E)

Table 2.1: Overview of the contributions involving automated and objective surgical skills assessment

Author (Year)	Machine learning method	Data type	Task
Benmansour *et al.* (2023) [59]	CNN-BiLSTM	Kinematic	Regression (OSATS)
Mahtab *et al.* (2016) [50]	Logisitc regression & SVM	Kinematic	Classification (N,E)
Liu *et al.* (2019) [51]	ANN	Video	Regression (OSATS, COF)
Benmansour *et al.* (2018) [56]	RNN	Kinematic	Regression on real scores (sequence-to-sequence)
Benmansour *et al.* (2019) [58]	LSTM	Kinematic	Regression on real scores (sequence-to-sequence)
Zia *et al.* (2018) [62]	ApEn	Video & accelerometer data	Classification (OSATS)
Zia & Essa (2018) [7]	ApEn	Kinematic	Classification (N,I,E) & regression (OSATS)
Kowalewski *et al.* (2019) [48]	DNN	Kinematic	Classification (N,I,E) & regression (OSATS)
Vaughan & Gabrys (2020) [63]	BiLSTM, ResNet, FCN, CNN	kinematic	Classification (N,I,E)

Table 2.2: Overview of the contributions involving automated and objective surgical skills assessment (continued)

Author (Year)	Machine learning method	Data type	Task
Li *et al.* (2019) [64]	RNN-based attention network	Video	Classification
Nguyen *et al.* (2019) [65]	CNN-LSTM	Kinematic	Classification (N,I,E)
Castro *et al.* [66]	CNN	Kinematic	Classification (N,I,E)
Anh *et al.* (2019) [67]	CNN, CNN-LSTM, Encoder-Decoder	Kinematic	Classification (N,I,E)
Khalid *et al.* (2020) [5]	BiLSTM (attention)	Kinematic	Classification (N,I,E) regression (OSATS)
Zhang *et al.* (2020) [68]	CNN	Kinematic	Classification (N,I,E)
Kim *et al.* (2019) [69]	Temporal CNN	Video	Classification
Jin et al. (2018) [37]	Region-based CNN	Video	Classification (N,E)
Mazomenos *et al.* (2018) [70]	VGG & AlexNet	Video	Regression

Table 2.3: Overview of the contributions involving automated and objective surgical skills assessment (continued)

2.6 Conclusion

In this chapter, we presented the most recent and the most relevant contributions in the topic of automated surgical skills assessment and surgical gesture recognition. We have shown that these works split into two categories: surgical skills classification methods (that covers most of the literature) and surgical skills regression methods. Among these contributions, we find video-based assessment methods but also kinematic-based assessment methods. CNN-based and RNN-based models represent the majority of the machine learning methods employed. We believe that automated surgical skills assessment will significantly benefit from the availability of large, high-quality, publicly accessible annotated datasets. In the next chapter, we will present the contents and the terminology of all the elements of the JIGSAWS dataset that we used to train our models.

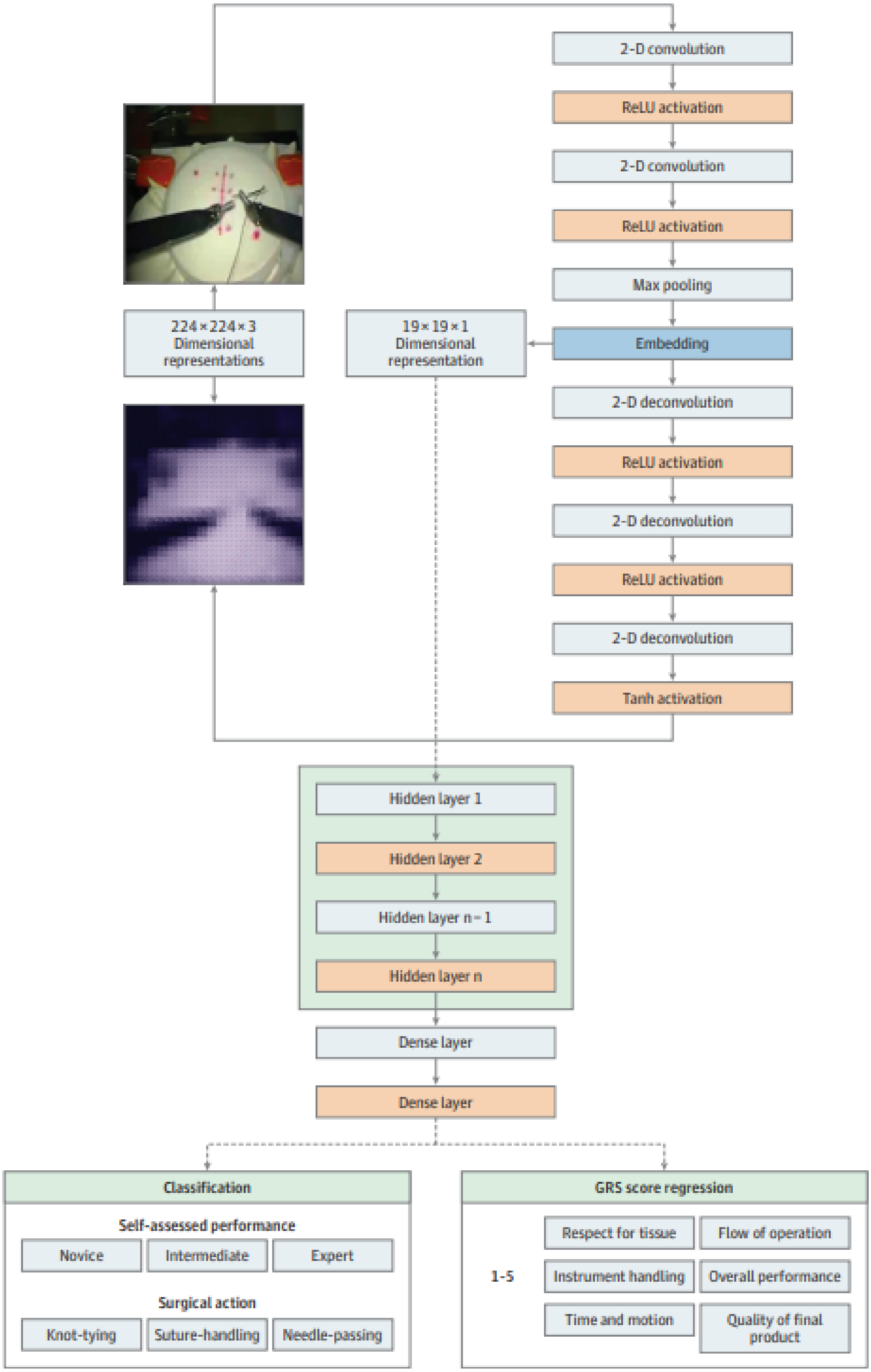

Figure 2.7: The model proposed by *Khalid et al.*[5]. It can be used as a classifier to predict surgical actions and self-reported skill and also as a regression model, which predicts scores on the Likert-scale for each category of the Global Rating Scale (GRS).

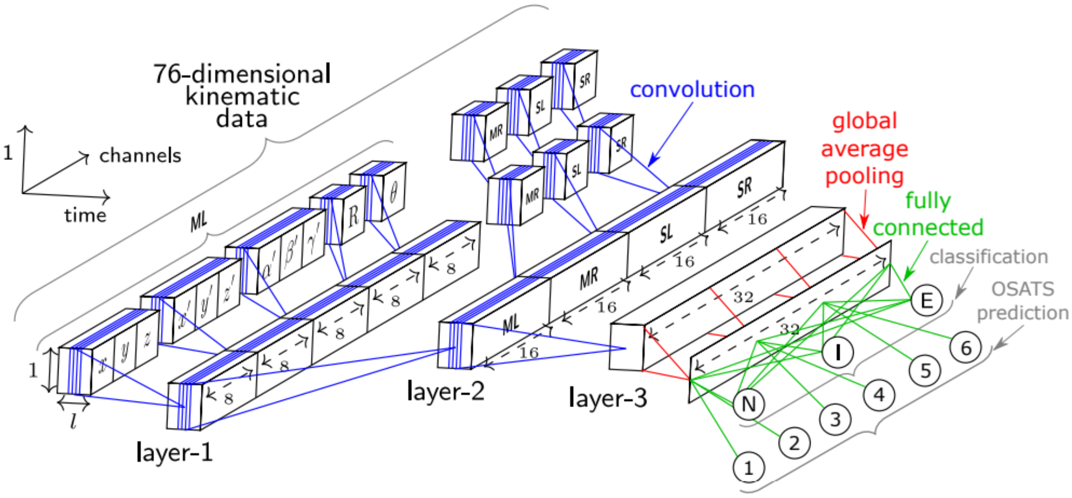

Figure 2.8: Fully Convolutional Network for surgical skill evaluation from kinematic data proposed by *Fawaz et al.* [6]

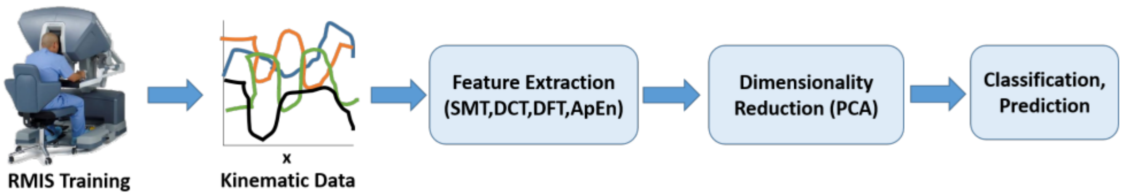

Figure 2.9: Flow diagram of the proposed framework for robotic surgical skills assessment by *Zia et al.* [7]

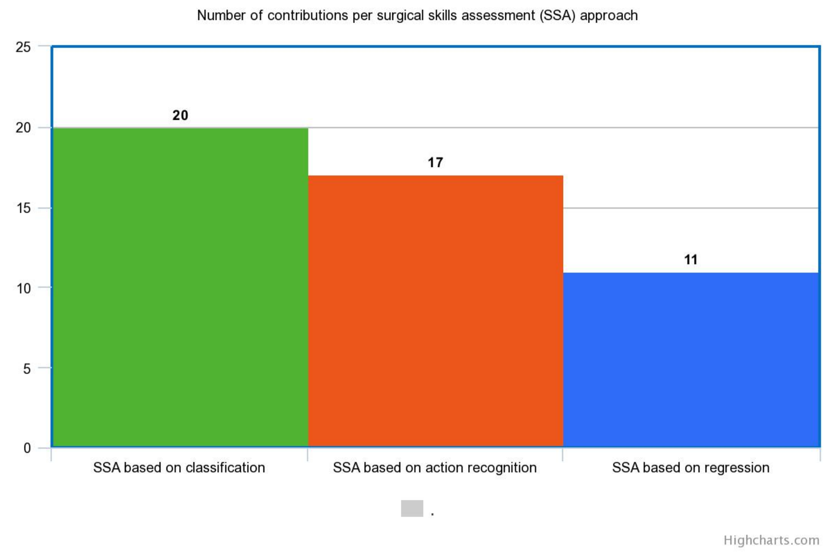

Figure 2.10: A histogram counting numbers of contributions per surgical skills assessment approach

Chapter 3

Surgical gesture flow based on kinematics

3.1 Introduction

The surgical gesture workflow is a process that can use kinematic data to analyze and understand the motion of surgical instruments during a procedure. The kinematic data, which includes information about the position, velocity, and acceleration of the instruments, is collected using sensors such as accelerometers and gyroscopes. This data is then processed and analyzed to understand the patterns and movements of the instruments during the procedure.

One of the key elements of the surgical gesture workflow is the identification and classification of different surgical gestures. This is typically done by analyzing the kinematic data and looking for specific patterns or movements that are characteristic of different gestures. For example, a suturing gesture may be characterized by a specific pattern of motion in the instruments, while a knot-tying gesture may be characterized by a different pattern of motion.

Once the different gestures have been identified and classified, the next step is to analyze the performance of the gestures in terms of accuracy, consistency, and efficiency. This can be done by comparing the kinematic data from the procedure to a set of predefined standards or benchmarks like scoring systems such as Objective Structured Assessment of Technical Skills (OSATS) [16], Global Evaluative Assessment of Robotic Skills (GEARS) [71] and Global Operative Assessment of Laparoscopic Skills (GOALS) [72].

Another important aspect of the surgical gesture workflow is the use of kinematic data to train and improve the skills of surgeons. By analyzing the kinematic data from a procedure, it is possible to identify areas where a surgeon may be struggling and provide targeted training to improve their skills. This can be done by providing the surgeon with feedback on their performance, or by providing them with simulated training exercises that are designed to improve their skills in specific areas.

In this chapter, we will define kinematic data in the context of surgery, present the existing surgical performance scoring systems and give an overview of available surgical datasets that have been used to validate the existing works in surgical skills assessment.

3.2 Surgical kinematic data

Surgical kinematic data refers to information about the motion of surgical instruments during a procedure. This data can be collected using sensors such as accelerometers, gyroscopes, and cameras, and can provide detailed information about the position, velocity, and acceleration of the instruments. This information can be used to analyze the movement of the instruments during the procedure, which can provide valuable insights into the performance of the surgery and the skill of the surgeon.

One of the main uses of surgical kinematic data is to evaluate the performance of surgical gestures. A surgical gesture is a specific movement or action performed by the surgeon during the procedure. For example, a suturing gesture may involve a specific pattern of motion in the instruments as the surgeon ties a knot, while a needle passing gesture may involve a different pattern of motion as the surgeon moves the needle through tissue. By analyzing the kinematic data, it is possible to identify and classify different surgical gestures, and to evaluate the performance of each gesture in terms of accuracy, consistency, and efficiency.

Surgical kinematic data can also be used to train and improve the skills of surgeons. By analyzing the data, it is possible to identify areas where a surgeon may be struggling, and to provide targeted training to improve their skills. For example, if a surgeon is having difficulty with a specific surgical gesture, they may be provided with simulated training exercises that are designed to improve their skills in that area. Additionally, the data can be used to provide feedback to the surgeon during the procedure, allowing them to make real-time adjustments to their technique.

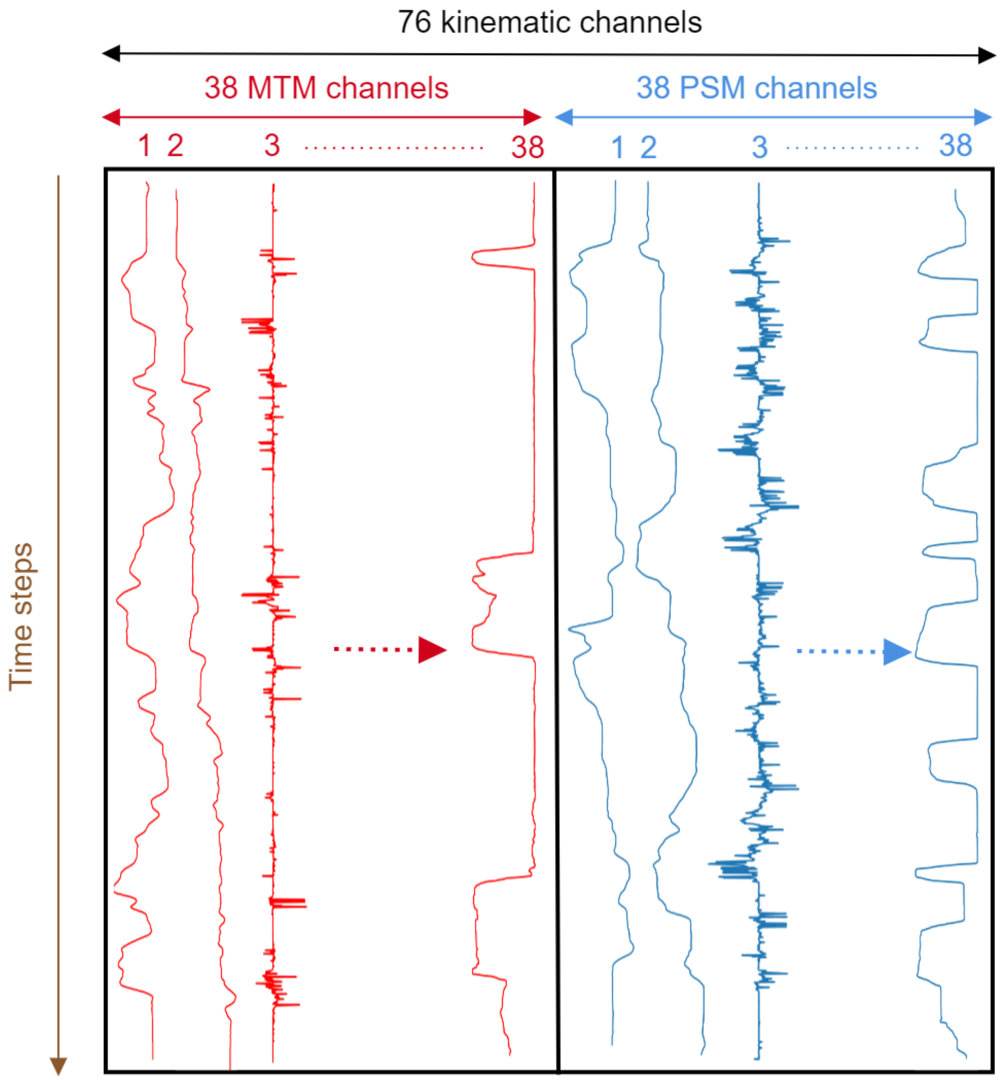

Figure 3.1: Representation of the kinematic sequence for one trial of one subject performing one task in JIGSAWS. Red and blue curves represents signals of different kinematic variables of the MTM and the PSM of the DaVinci, respectively.

3.3 The JHU-ISI Gesture and Skill Assessment Working Set (JIGSAWS)

The JHU-ISI Gesture and Skill Assessment Working Set (JIGSAWS) is a dataset that was created by researchers at the Johns Hopkins University and the Information Sciences Institute [19]. It consists of kinematic data that was collected from surgeons performing a variety of surgical tasks using the Da Vinci surgical system (DVSS) [28]. The dataset includes data from expert, intermediate and novice surgeons, and covers a wide range of procedures, including suturing, needle passing, and knot tying. The JIGSAWS dataset is used to support research on surgical skill assessment and training, as well as to evaluate and improve the design of robotic surgical systems. It includes motion data of the robot's instruments and video data of the task, which can be used to analyze the motion of the surgical instruments, the motion of the surgical site, and the motion of the surgeon's hands. It also includes information about the task itself, such as the type of procedure, the type of instrument being used, and the phase of the task. The JIGSAWS dataset is publicly available and has been widely used in the field of surgical robotics research. In the following sub-sections, we present the DaVinci surgical system and the content of the JIGSAWS dataset.

3.4 The DaVinci surgical system (DVSS)

The DaVinci Surgical System is a robotic surgical platform that is designed to assist surgeons in performing complex procedures. It consists of a surgeon's console, which is where the surgeon sits and controls the system, and a patient-side cart, which contains the robotic arms that perform the surgery. The system is designed to provide the surgeon with enhanced dexterity, precision, and control during the procedure. The DVSS uses a combination of advanced robotics, miniaturization, and computer-enhanced technology to enable the surgeon to perform complex procedures with greater precision and control than would be possible with traditional laparoscopic or open surgical techniques. The system also features high-definition, 3D imaging that allows the surgeon to see inside the patient's body in real-time, providing a greater level of detail and accuracy than would be possible with traditional 2D imaging. The DVSS is used for a wide range of procedures, including urologic, gynecologic, and general surgical procedures. The system is comprised of a master-side console with two master tool manipulators (MTMs) that are operated by the surgeon, a patient-side robot with three patient-side manipulator (PSMs) and a camera arm with a stereo endoscope (see figure 3.2). During surgery, the surgeon tele-operates the PSMs by manipulating the MTMs. The systems also includes a research interface called Application Programming Interface (API) that awllows data capture in real-time.

The use of surgical robots, such as the DaVinci, has made it possible to collect detailed kinematic data during surgical procedures. However, the kinematic and dynamic models of the DaVinci are not accessible due to the intellectual property of the robot and the proprietary nature of its software, making it challenging to develop accurate and reliable metrics for surgical skills assessment. Moreover, an exact and accurate kinematic model of the DaVinci would be impossible to acquire due to the high complexity of the robot. In addition, surgical robots do not share a unique kinematic model and thus, acquiring it from each robot is laboring. Therefore, there is a need of a surgical skills assessment system that is independent of the kinematic model of the used surgical system. To address this challenge, we can use machine learning to extract patterns from surgical kinematic data for surgical skills assessment. Machine learning algorithms can be trained to analyze large amounts of kinematic data and identify correlations and patterns that can be used to assess surgical skills. machine learning can provide a powerful tool for extracting patterns from DaVinci kinematic data for surgical skills assessment. By using machine learning to estimate the states that are not directly available by measurements (like surgical skills), we can develop accurate and reliable metrics for surgical skills assessment and train surgeons in various environments. Nonlinear observers can be used for surgical skills assessment without accessing the kinematic and dynamic model of the DaVinci robot. Nonlinear observers are mathematical algorithms that estimate available and unavailable states of a system based on available measurements (see chapter 1), even when the system is nonlinear and its model is not known. Therefore, nonlinear observers can be either built based on the kinematic model or they can use machine learning when the kinematic model is unavailable. To use nonlinear observers for surgical skills assessment with the DaVinci robot, we can collect data from the robot's sensors during a surgical procedure. Based on this data, we can design a nonlinear observer that estimates surgical performance scores by taking as input collected kinematic data. This observer does not require knowledge of the kinematic and dynamic model of the robot, as it uses the available sensor data to estimate the state of the system and the surgical performance score.

To establish the JIGSAWS dataset, several studies have been conducted by capturing surgical activity data using the API of the dVSS. It was used to collect motion data from the MTMs and PSMs, as well as the camera pose, video data, and system events and converted them into an accessible format.

3.5 Dataset

The JIGSAWS includes data on three elementary surgical tasks performed by study subjects (surgeons) on bench-top models. All three tasks (or their variants) are typically part of surgical skills training curricula. The three tasks include:

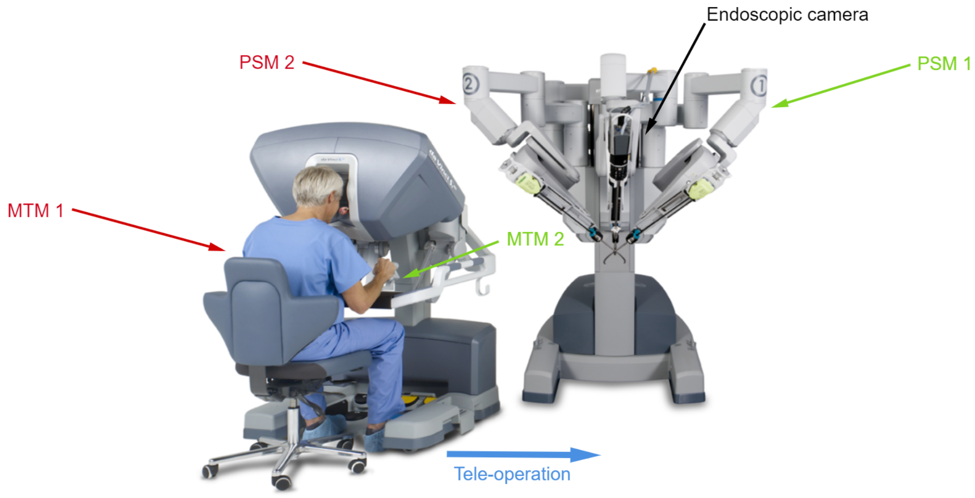

Figure 3.2: An illustration of the Da Vinci surgical system highlighting its different tools.

- **Suturing (SU)**: The subject picks up needle, proceeds to the incision (designated as a vertical line on the bench-top model), and passes the needle through the "tissue", entering at the dot marked on one side of the incision and exiting at the corresponding dot marked on the other side of the incision. After the first needle pass, the subject extracts the needle out of the tissue, passes it to the right hand and repeats the needle pass three more times.

- **Knot-Tying (KT)**: The subject picks up one end of a suture tied to a flexible tube attached at its ends to the surface of the bench-top model, and ties a single loop knot.

- **Needle-Passing (NP)**: The subject picks up the needle (in some cases not captured in the video) and passes it through four small metal hoops from right to left. The hoops are attached at a small height above the surface of the bench-top model.

The subjects were not allowed to move the camera or apply the clutch while performing the tasks. Figure 3.3 shows snapshots of the three surgical tasks.

3.5.1 Subjects

The JIGSAWS includes data from eight subjects, indexed as "B", "C", "D", "E", "F", "G", H", "I" with varying robotic surgical experience. Subjects D and E reported having more than 100 hours, subjects B, G, H, and I reported having fewer than 10 hours, and subjects C and F reported between 10 and 100 hours of robotic surgical experience. All subjects were reportedly right-handed. All subjects repeated each surgical task five times. Each of the

Figure 3.3: Sample frames from the three tasks in the JIGSAWS dataset, from left to right: Suturing frame, Knot-Tying frame, Needle-Passing frame.

five instances of a task performed by a subject is referred as a “trial”. Figure 3.5 depicts the general organization of the JIGSAWS data folder. Figures 3.4, 3.6 and 3.7 depict 3D trajectories of the PSM1 tool while novice, intermediate and expert subjects are performing the Knot-Tying, Needle-Passing and Suturing, respectively.

3.5.2 Data description

The JIGSAWS consists of three components: kinematic data, video data, and manual annotations. Kinematic data have been captured from the dVSS using its API at 30 Hz. The left and right MTMs, and the first and second PSMs (PSM1 and PSM2, also referred as the right and left PSMs in this dataset), are included in the dataset. The motion of each manipulator was described by a local frame attached at the far end of the manipulator using 19 kinematic variables, therefore there are 76 dimensional data to describe the kinematics for all four manipulators listed above. The 19 kinematic variables for each manipulator include Cartesian positions (3 variables, denoted by xyz), a rotation matrix (9 variables, denoted by R), linear velocities (3 variables, denoted by x', y', z') , angular velocities (3 variables, denoted by α', β', γ') where α, β, γ are Euler angles), and a gripper angle (denoted by θ). All kinematic variables are represented within a common coordinate system. Table 3.1 describes the details of the variables included in the kinematic dataset. The kinematic data for the MTMs, PSMs, and the video data were synchronized with the same sampling rate.

In addition, stereo video data were captured from both endoscopic cameras of the dVSS at 30 Hz and at a resolution of 640 x 480. The video and kinematic data were synchronized such that each video frame corresponds to a kinematic data frame captured at the same instant of time. The videos in the dataset being released are saved as AVI files encoded in four

Column indices	Number of variables	Description of variables
1-3	3	Left MTM tool tip position (xyz)
4-12	9	Left MTM tool tip rotation matrix (R)
13-15	3	Left MTM tool tip linear velocity ($x'y'z'$)
16-18	3	Left MTM tool tip rotational velocity ($\alpha'\beta'\gamma'$)
19	1	Left MTM gripper angle velocity (θ)
20-38	19	Right MTM kinematics
39-41	3	PSM1 tool tip position (xyz)
42-50	9	PSM1 tool tip rotation matrix (R)
51-53	3	PSM1 tool tip linear velocity
54-56	3	PSM1 tool tip rotational velocity
57	1	PSM1 gripper angle velocity
58-76	19	PSM2 kinematics

Table 3.1: Kinematic data variables.

character code (FOURCC). format with the DX50 codec. The video files named "capture1" and "capture2" were recorded from the left and right endoscopic cameras, respectively. Figure 3.3 shows a snapshot of corresponding left and right images for a single frame. The dataset does not include calibration parameters for the two endoscopic cameras.

3.5.3 Surgical activity annotation

The manually annotated ground-truth for atomic surgical activity segments known as "gestures" or "surgemes" is a key component of the JIGSAWS. An atomic unit of deliberate surgical activity that produces a measurable and significant outcome is referred to as a surgical gesture. Through consultation with a seasoned cardiac surgeon who has an established robotic surgical practice, a common vocabulary of 15 elements was developed to describe gestures for each of the three tasks in the dataset. This vocabulary is detailed in Table 3.2.

After consulting with the surgeon, one individual watched the videos and annotated the data for surgical gestures. The name of the gesture and the video's beginning and ending frames are included in each annotation. We synchronized the video frames using a common global time, which results in one-to-one mapping (the video frames matching the kinematic data frames). With the exception of a few frames at the conclusion of a trial after the task was completed, a gesture label has been assigned to all frames for each trial. Not all of the gestures in the vocabulary are used to annotate each of the three surgical tasks because the authors used the same vocabulary for all three tasks. The background environ-

Gesture index	Gesture description
G1	Reaching for needle with right hand
G2	Positioning needle
G3	Pushing needle through tissue
G4	Transferring needle from left to right
G5	Moving to center with needle in grip
G6	Pulling suture with left hand
G7	Pulling suture with right hand
G8	Orienting needle
G9	Using right hand to help tighten suture
G10	Loosening more suture
G11	Dropping suture at end and moving to end points
G12	Reaching for needle with left hand
G13	Making C loop around right hand
G14	Reaching for suture with right hand
G15	Pulling suture with both hands.

Table 3.2: Gestures vocabulary.

ment varies between tasks, despite the fact that some gestures can be observed in multiple tasks. The "Pushing needle through tissue (G3)" gesture, for instance, can be seen in both SU and NP tasks. In contrast, the needle is advanced through tissue in SU and a metal hoop in NP.

3.6 Surgical skill annotation using various scoring systems

3.6.1 Using OSATS

The JIGSAWS also includes manual annotation for surgical technical skill. A gynecologic surgeon with extensive robotic and laparoscopic surgical experience watched each video and assigned a global rating score (GRS) using a modified Objective Structured Assessments of Technical Skills (OSATS) approach. The OSATS is a reliable and valid tool that has been increasingly used in orthopaedic skills training. It uses the GRS to structure expert evaluation of technical skills with the experts working from a list of operative competencies that are each rated on a 5-point Likert scale anchored by behavioral descriptors. In this work, we propose a deep learning architecture to provide an automatic and objective surgical skill assessment on a continuous scale ranging from 1 to 5 based on OSATS criteria [54, 16]. This continuous scale has been approximated from the OSATS scale which is originally a Likert scale and thus, composed of ordinal items. Such approximation is possible if intervals between each level in the Likert scale item can be presumed equal [73]. Moreover, a group of researchers maintains that Likert, or ordinal variables with five or more categories can often be used as continuous without any harm to the analysis the researcher plans to use them in [74, 75, 76]. Therefore, according to these researches and since every OSATS Likert item is composed of 5 assessment levels, an approximation with a continuous function is possible and it can be included in a regression process. The original OSATS method [16] was established in 1997 by the Surgical Education Research Group which belongs to the University of Toronto, Canada. The authors claimed that the surgeons were not well assessed regarding to their technical skill. Therefore, they came up with an approach that can deliver an accurate evaluation of residents who performed a variety of structured operative tasks under the supervision of mentor surgeons certified by the Royal College of Physicians and Surgeons of Canada . Both live animals and bench models were used for this. They developed two types of scoring system. A first one called operation-specific checklist which consists of attributing a binary value (1 for "Done Correctly" and 0 for "Not Done or Done Incorrectly") for the performed steps composing a certain surgical task. The second scoring system consists of a detailed global rating scale. This second scoring system aims to give a score on a scale of 1 to 5 on seven criteria: (1) Respect for Tissue, (2) Time and motion, (3) Instrument handling, (4) Knowledge of instruments, (5) Use of assistants, (6) Flow of operation and forward planning, (7) Knowledge of specific procedure. Each score (from 1 to 5) describes the quality of the skill on the concerned criterion. The goal of the whole work is to validate the two scoring systems and compare them. At the end, The authors have strongly proven by the obtained results that the Objective Structured Assessment of Technical Skill can reliably and validly assess surgical skills. They demonstrated and concluded that global ratings are a better method of assessment than task-specific checklists. This is why the JIGSAWS authors have chosen to assess their own trainees by using the global ratings method. Nevertheless, they did not perform the evaluation on live animals but only on workbench models and they also barely modified the assessment criteria by deleting or changing some items according to their own working conditions. For example, there was not any assistant for the subjects and the subjects performed operations with the Da Vinci surgical system using only a suture and a needle. In [19], one can see that "Instrument handling"has been replaced by "Suture/Needle handling". There is no more "Knowledge of specific procedure" and "Knowledge of instruments" but instead, one can find "Overall performance" and "Quality of final product". Even the scores descriptions differ from the original OSATS global rating scale. Interpretations of 1, 3 and 5 OSATS scores can be read in Table 3.3. However, JIGSAWS surgeons did not give interpretation for 2 and 4 scores despite being accorded to several trainees. The meta files contained in the JIGSAWS database show the obtained scores by each subject on each criterion of the modified OSATS criteria regarding to the three surgical tasks: Knot-Tying, Needle-Passing and Suturing

Element	Rating scale
Respect for tissue	1-Frequently used unnecessary force on tissue; 3- Careful tissue handling but occasionally caused inadvertent damage; 5-Consistent appropriate tissue handling;
Suture/needle handling	1- Awkward and unsure with repeated entanglement and poor knot tying; 3-Majority of knots placed correctly with appropriate tension; 5-Excellent suture control
Time and motion	1-Made unnecessary moves; 3-Efficient time/motion but some unnecessary moves; 5-Clear economy of movement and maximum efficiency
Flow of operation	1-Frequently interrupted flow to discuss the next move; 3-Demonstrated some forward planning and reasonable procedure progression; 5-Obviously planned course of operation with efficient transitions between moves;
Overall performance	1-Very poor; 3-Competent; 5-Clearly Superior;
Quality of final product	1-Very poor; 3-Competent; 5-Clearly Superior;

Table 3.3: Elements of modified OSATS scoring system [19].

3.6.2 Using GOALS

Global Operative Assessment for Laparoscopic Skills (GOALS) is a validated assessment tool for grading overall technical proficiency for laparoscopic surgery developed by *Vassiliou et al.* [72]. Table 3.4 presents interpretations of the scores composing this scoring system.

3.6.3 Using GEARS

In collaboration with experienced robotic surgeons, the Global Evaluative Assessment of Robotic Skills [71] is a tool that was developed by dissecting the fundamental components of

Element	Rating scale
Depth Perception	1- Constantly overshoots target, wide swings, slow to correct; 3- Some overshooting or missing target, but quick to correct; 5- Accurately directs instruments in the correct plane to target;
Bimanual dexterity	1- Uses only one hand, ignores non dominant hand, poor coordination between hands; 3- Users both hands, but does not optimize interaction between hands; 5- Expertly uses both hands in a complementary manner to provide optimal exposure;
Efficiency	1- Uncertain, inefficient efforts; many tentative movements; constantly changing focus or persisting without progress; 3- Slow, but planned movements are reasonably organized; 5- Confident, efficient and safe conduct, maintains focus on task until it is better performed by way of an alternative approach;
Tissue handling	1- Rough movements, tears tissue, injures adjacent structures, poor grasper control, grasper frequently slips; 3- Handles tissue reasonably well, minor trauma to adjacent tissue (i.e., occasional unnecessary bleeding or slipping of the grasper); 5- Handles tissues well, applies appropriate traction, negligible injury to adjacent structures;
Autonomy	1- Unable to complete entire task, even with verbal guidance; 3- Able to complete task safely with moderate guidance; 5- Able to complete task independently without prompting;

Table 3.4: Elements of GOALS scoring system [72].

robotic surgical procedures. During a robot-assisted laparoscopic prostatectomy, six domains of surgical performance were evaluated using a 5-point anchored Likert scale. The sum of the ratings in each domain was used to calculate the overall performance score. To determine construct validity, expert surgeons and postgraduate urology residents in their fourth to sixth year were evaluated. The operator, a trained observer, and the attending surgeon all completed the assessments. GEAR is an expansion of GOALS done by encompassing features unique to robotic surgery [71]. Table 3.5 presents interpretations of the scores composing this scoring system.

Element	Rating scale
Depth Perception	1- Constantly overshoots target, wide swings, slow to correct; 3- Some overshooting or missing target, but quick to correct; 5- Accurately directs instruments in the correct plane to target;
Bimanual dexterity	1- Uses only one hand, ignores non dominant hand, poor coordination between hands; 3- Users both hands, but does not optimize interaction between hands; 5- Expertly uses both hands in a complementary manner to provide optimal exposure;
Efficiency	1- inefficient efforts; many uncertain movements; constantly changing focus or persisting without progress; 3- Slow, but planned movements are reasonably organized; 5- Confident, efficient and safe conduct, maintains focus on task until it is better performed by way of an alternative approach;
Tissue handling	1- Rough movements, tears tissue, injures adjacent structures, poor grasper control, grasper frequently slips; 3- Handles tissue reasonably well, minor trauma to adjacent tissue (i.e., occasional unnecessary bleeding or slipping of the grasper); 5- Handles tissues well, applies appropriate traction, negligible injury to adjacent structures;
Force sensitivity	1- Rough moves, tears tissue, injures nearby structures, poor control, frequent suture breakage ; 3- Handles tissues reasonably well, minor trauma to adjacent tissue, rare structure breakage ; 5- Applies appropriate tension, negligible injury to adjacent structures, no suture breakage;
Autonomy	1- Unable to complete entire task, even with verbal guidance; 3- Able to complete task safely with moderate guidance; 5- Able to complete task independently without prompting;
Robotic control	1- Consistently does not optimize view, hand position, or repeated collisions even with guidance; 3- View is sometimes not optimal. Occasionally needs to relocate arms. Occasional collisions and obstruction of assistant 5- Controls camera and hand position optimally and independently. Minimal collisions or obstruction of assistant

Table 3.5: Elements of GEARS scoring system [71].

Author (Year)	Machine learning method	Validation scheme	Dataset used for experiment
Law *et al.* (2017) [35]	Hourgalss Network (CNN) + SVM	LOOCV	Vesicourethral Anastomosis
Funke *et al.* (2019) [38]	3DCNN	LOSO & LOUO	JIGSAWS
Doughty *et al.* (2018) [60]	Two streams CNN	k-fold	JIGSAWS
Fawaz *et al.* (2019) [6]	FCN	LOSO	JIGSAWS
Wang & Fey (2020) [3]	CNN	LOSO & Hold-out	JIGSAWS
Wang & Fey (2020) [46]	CNN-GRU	LOSO	JIGSAWS
Benmansour *et al.* (2018) [55]	DNN	Training on N,I,E trials, validation on N,I trials	JIGSAWS
Kelly *et al.* (2020) [47]	BiLSTM	Stratified CV	BLUS
Ahmidi *et al.* (2015) [61]	SVM	LOSO & LOUO	Septoplasty dataset collected from a cohort study

Table 3.6: Overview of the contributions involving automated and objective surgical skills assessment

Author (Year)	Machine learning method	Validation scheme	Dataset used for experiment
Benmansour *et al.* (2023)	CNN-BiLSTM	LOSO & LOGO	JIGSAWS
Mahtab *et al.* (2016) [50]	Logisitc regression & SVM	LOSO & LOUO	JIGSAWS
Liu *et al.* (2019) [51]	ANN	k-fold	In-Vivo clinical data
Benmansour *et al.* (2018) [56]	RNN	Training on N,I,E trials, validation on N,I trials	JIGSAWS
Benmansour *et al.* (2019) [58]	LSTM	Training on N,I,E trials, validation on N,I trials	JIGSAWS
Zia *et al.* (2018) [62]	ApEn	LOOCV	JIGSAWS
Zia & Essa (2018) [7]	ApEn	LOSO & LOUO	JIGSAWS
Kowalewski *et al.* (2019) [48]	DNN	k-fold	KT & ST with Myo armband
Vaughan & Gabrys (2020) [63]	BiLSTM, ResNet, FCN, CNN	k-fold & LOOCV	Virtual epidural training dataset

Table 3.7: Overview of the contributions involving automated and objective surgical skills assessment (continued)

Author (Year)	Machine learning method	Validation scheme	Dataset used for experiment
Li *et al.* (2019) [64]	RNN-based attention network	k-fold	JIGSAWS and other non-surgical databases
Nguyen *et al.* (2019) [65]	CNN-LSTM	LOSO	JIGSAWS
Castro *et al.* [66]	CNN	LOSO	JIGSAWS
Anh *et al.* (2019) [67]	CNN, CNN-LSTM, Encoder-Decoder	LOSO	JIGSAWS
Khalid *et al.* (2020) [5]	BiLSTM (attention)	LOSO & LOUO	JIGSAWS
Zhang *et al.* (2020) [68]	CNN	LOSO	JIGSAWS
Kim *et al.* (2019) [69]	Temporal CNN	k-fold	Capsulorhexis, videos of cataract surgery
Jin et al. (2018) [37]	Region-based CNN	Data splitting	m2cai16-tool
Mazomenos *et al.* (2018) [70]	VGG & AlexNet	N/A	Virtual transesophageal echocardiography dataset

Table 3.8: Overview of the contributions involving automated and objective surgical skills assessment (continued)

3.7 Data augmentation methods

Data augmentation is a technique used in deep learning to artificially increase the size of a dataset by creating new examples of the same data using various transformations. While data augmentation is commonly used in computer vision applications, it is also used in time series applications [77, 78]. The use of data augmentation can improve the performance of deep learning models for time series by increasing the diversity of the training data, reducing overfitting, and improving generalization. Some of the general benefits of data augmentation for time series in deep learning include:

- Improving model accuracy: By increasing the diversity of the training data, data augmentation can help to improve the accuracy of deep learning models for time series.
- Reducing overfitting: Data augmentation can help to reduce overfitting by making the model more robust to variations in the input data.
- Improving generalization: By increasing the diversity of the training data, data augmentation can help to improve the generalization of deep learning models for time series, allowing them to perform better on unseen data.
- Increasing dataset size: Data augmentation increases the size of the dataset, allowing deep learning models to train on more data and learn more complex patterns.
- Better handling of missing data: Data augmentation can be used to fill in missing data values in time series, improving the robustness of the model.

For basic and simple neural networks architectures that we used in the contributions presented in chapter 4, the JIGSAWS kinematic data were sufficient to obtain satisfying results. But with more complex architectures that are presented in chapter 5, serious model overfitting occurred because of the lack of data and providing a data augmentation pipeline was mandatory. We present here the data augmentation methods we applied in the works presented in chapter 5 as well as some others common data augmentation for time series.

Sliding window process

Sliding window [79] is a technique used in machine learning for time series data to increase the amount of data available for training and to improve the generalization performance of the model. This technique involves dividing the time series into smaller segments, called windows, and using these windows as separate training examples [3]. In sliding window, each window is treated as a separate training example, and the model is trained on these examples instead of the original time series data. The window size and step size can be adjusted to control the number of examples generated from the original time series data. Sliding window data augmentation is beneficial for machine learning models for several reasons:

- Increases the amount of data available for training: By breaking down the time series into smaller windows, more training examples can be generated from the original data, which can improve the model's performance.
- Helps capture local patterns: Sliding window data augmentation allows the model to capture local patterns and trends in the time series that may not be apparent in the original data.
- Helps prevent overfitting: By generating multiple training examples from the same time series data, the model is less likely to overfit to the original data, which can improve the model's generalization performance.
- Helps capture temporal dependencies: Sliding window data augmentation allows the model to capture temporal dependencies between adjacent windows, which can improve the model's ability to make accurate predictions.

In the case of the JIGSAWS time series data, each one has a typical length of between 1 and 5 minutes. With 8 surgeons and 5 trials per surgeon per procedure, the JIGSAWS dataset contains only 40 independent samples for training each surgical task's model. To overcome the training limitation and further boost the training data, we withdraw a large volume of cropped sequences using a sliding window algorithm that has been used by *Wang et al.* [3]. Figure 3.8 depicts a diagram of the sliding window data augmentation pipeline. The small crops on the right side of the figure represent the extracted sub-sequences from the entire trial sequence. It is important to point out that the extracted sub-sequences have the same output label than the entire sequence. The sliding window algorithm can create a large number of new examples of the same data by taking rolling windows of the time series[1]. This can increase the size of the training dataset and help to improve the generalization performance

[1]The extracted sub-sequences using the sliding window algorithm are not considered as new time-series data but rather as fragments of entire sequences that allow us to increase the amount of data samples.

of the neural network. In addition, by creating new examples of the same data with different starting points, the neural network can learn to be more robust to time shifts in the input data. This can help to improve the accuracy and reliability of the neural network.

Tools data separation

Kinematic data recorded for robotic devices contain observations of positions, velocities and other features of robot tools across time. A way to augment time series data of a dataset is to separate data of different tools channels an considering them as different samples. In the case of the JIGSAWS time series, for each row sample in the kinematic data, we separate the MTM and PSM channels of the DaVinci robot into two instances and cast them as distinct samples. Instead of 76 sensors per row sample, for example, there will be 38 sensors for every two row samples. This “splitting and doubling” step is allowable because the MTM and PSM sensors are uncorrelated due to differences in position of the robot control arms. Figure 3.8 depicts a diagram of the sliding window data augmentation pipeline combined with the tools data separation technique. The small crops on the right side of the figure represent the extracted sub-sequences from the entire trial sequence. It is important to point out that the extracted sub-sequences have the same output label than the entire sequence. The sliding window algorithm can create a large number of new examples of the same data by taking rolling windows of the time series. This can increase the size of the training dataset and help to improve the generalization performance of the neural network. In addition, by creating new examples of the same data with different starting points, the neural network can learn to be more robust to time shifts in the input data. This can help to improve the accuracy and reliability of the neural network. In the work presented in chapter 5, we propose a data augmentation method that combines the sliding window algorithm with the tool data separation technique.

Scaling

Scaling [80] involves multiplying the entire time series by a constant factor. This can be useful if we want to simulate different levels of signal strength or amplitude variations. In our work presented in chapter 5, each channel of raw sensor data is z-normalized to minimize the differences in scaling ranges of each sensor. Thus, we did not apply the scaling technique.

Noise injection

Adding random noise to the time series can help simulate variability and randomness in the data. This can be done by adding Gaussian noise, uniform noise, or other types of noise to the time series [77]. However, it is important to choose the right amount of noise that will be applied to JIGSAWS time series in order not to distort the data and thus confuse the machine learning model.

Discrete Fourier Transform The Discrete Fourier Transform (DFT) is a mathematical technique that transforms a time-domain signal into its frequency-domain representation. The DFT is commonly used for data augmentation in time series analysis. Transforming the time series into the frequency domain: The first step in using the DFT for data augmentation is to transform the time series into the frequency domain. This is done by applying the DFT to the time series. The DFT converts the time-domain signal into a set of complex frequency-domain coefficients, where each coefficient represents the amplitude and phase of a sinusoidal waveform at a particular frequency. After moving to the frequency domain, we can apply the aforementioned data augmentation pipeline.

3.8 Validation schemes

Validation schemes are used in deep learning to evaluate the performance of a model on data that was not used during training. Here are some common validation schemes used in deep learning that can be applicable to JIGSAWS data, as well as a custom validation scheme we created for the work presented in chapter 5:

Leave-One-Supertrial-Out (LOSO): As described previously, the JIGSAWS dataset comprises 5 trials for each subject performing one of the three surgical tasks. The LOSO scheme consists in leaving a folder containing one supertrial, i, of each subject for testing the network while the remaining trials are used for the training step. This process is repeated five times in five folders in order to test the robustness of the model with each trial. At the end, the model that delivered the best performance with any of the five trials is saved. The five folders are set before augmenting the data (entire trials). Figure 4.4 depicts the process of the LOSO scheme. The LOSO scheme can reveal the variability and stability of the model's performance across different trials or, which can be useful for assessing the generalization ability of the model in real-world scenarios. However, since the training trials and the testing trials belongs to one subject, some training information may leak in the testing set and thus lead to biased prediction.

Leave-One-User-Out (LOUO): A robust way to test the neural network prediction reliability. It consists in leaving out all trials of a single subject for testing while the data of the remaining subjects are used for training. The process is repeated 8 times across the 8 JIGSAWS subjects (eight different validation folders). At the end, the mean of performance of the network overs the eight folders is computed. Figure 3.10 represents the process of this validation strategy applied to the case of the JIGSAWS dataset. LOUO provides a more stringent evaluation of the model's performance compared to other cross-validation methods, since it tests the model on completely new and independent users or individuals. However, it may require more data and computational resources than other cross-validation methods, since it involves leaving out all trials of a user from the training data and using them for testing. This can be a limitation in cases where the available data is limited or the model is computationally expensive.

Leave-One-Out Cross Validation (LOOCV): For a set of n data points, it consists in leaving out 1 data point for testing while the remaining $n-1$ data points are used for training. The process is repeated until all data points have been used for testing the network. In the case of the JIGSAWS dataset, there are 40 trials for each task. Thus, 1 trial is left for testing and the remaining 39 trials are used for training, regardless of the subjects. At the end, the mean of performance of the network is computed after all data points have been left out for testing. Figure 3.12 depicts the process of this validation strategy applied to the JIGSAWS case. LOOCV provides an unbiased estimate of the model's performance, as it uses all available data for both training and testing, and repeats the process for each data point. Moreover, it can reveal the sensitivity of the model's performance to individual data points, which can be useful for identifying influential observations or outliers that may be affecting the model's predictions. However, LOOCV may suffer from high variability or instability in the estimated performance, especially for models with high variance or noise. This is because the model is trained on a slightly different dataset for each round, which can lead to different performance estimates and make it difficult to compare models or draw conclusions. Additionally, it can be computationally expensive for large datasets, as it requires fitting the model for each data point, which can be time-consuming and memory-intensive. This can be a limitation in cases where the available resources are limited or the model is computationally expensive.

K-fold cross-validation: In this scheme, the data is divided into k equal-sized subsets. The model is trained on $k-1$ subsets and evaluated on the remaining subset. This process is repeated k times, with each subset serving as the validation set once. The performance is then averaged over the k iterations. We point out that LOUO is a particular form of K-fold where the trials of a single subject are left out for testing in each test folder. This technique is useful when the dataset is small and there is a need to use as much data as possible for training. k-fold cross-validation provides a relatively unbiased estimate of the model's performance, as it uses all available data for both training and testing, and averages the results over multiple rounds. However, the choice of the number of folds k can influence the performance estimates and can be a source of variability. A larger value of k may lead to more accurate estimates but may also result in a higher computational cost, while a smaller value of k may lead to higher variability and may not capture the full range of the model's performance.

Leave-One-Grade-Out (LOGO): This validation scheme is a custom scheme we propose for the work presented in chapter 5. It consists in choosing one grade (or score) of each OSATS, GOALS or GEARS criterion for each surgical task and leaving it for the test step. The purpose of this scheme is to test if the network is capable of predicting a grade that is unseen during the training step. Thus, this validation method focuses on the outputs (scores), unlike the aforementioned validation schemes that focus on the inputs (kinematic data). Figure 3.13 depicts the diagram of this validation scheme.

3.9 Discussion

Tables 3.6, 3.7 and 3.8 regroup works and compare the used machine learning method, the validation scheme and the dataset used for experiment. Among the used databases, JIGSAWS [19] is the most common ground to train and the test the machine learning models for the assessment of surgical skills. Its popularity may rely on the fact that it is publicly available and the performed tasks, Knot-Tying, Needle-Passing and Suturing are basic surgical tasks and are easily processed by a machine learning model. In addition, it contains both video and kinematic data recorded from surgeons with different expertise levels that have been categorized as expert, intermediate and novice. Also, it includes scores annotation using the OSATS criteria for performance grading. Thus, expertise levels and OSATS annotation allow surgical skills assessment by adopting both classification and regression methods. While JIGSAWS is a rich database in terms of the variety of data and performed tasks, the variety of expertise levels of the surgeons and the reliable OSATS annotation system, it still suffers from lack of data sample per expertise level and per surgical task. Figures 3.4, 3.6 and 3.7 represent the motion of one of the Patient Side Manipulators of novice, intermediate and expert subjects performing Knot-Tying, Needle-Passing and Suturing, respectively. After examination, we can not notice clear discriminant differences between expertise levels. The only slight characteristic that an expert surgeon shows in these figures is smoothness of the movements. Therefore, using machine learning to identify hidden discriminant patterns is mandatory. Neural networks deal well with time series data because they are able to learn complex temporal patterns and dependencies in the data. Unlike traditional statistical models that assume linear relationships between variables, neural networks are capable of modeling non-linear relationships between variables, making them well-suited for modeling complex time series data. Neural networks are especially effective for time series data because they can learn from historical patterns and use that information to make accurate predictions about future values. This is achieved by the use of recurrent neural networks, which are designed specifically for processing sequential data. RNNs have a feedback loop that allows information to be passed from one time step to the next, which allows the network to maintain an internal state that captures important temporal information. Additionally, neural networks can be trained using backpropagation, a process where the network is trained to minimize the difference between its predicted output and the actual output. This allows the network to learn from its mistakes and continually improve its predictions. Overall, neural networks are able to capture complex patterns in time series data, learn from historical patterns, and make accurate predictions about future values, making them a powerful tool for time series analysis and prediction.

Concerning the validation scheme, LOSO is the most adopted cross-validation method for surgical skills assessment. Results obtained by LOSO are always superior to those obtained with LOUO because, with the LOSO, the model is able to see data from all surgeons and thus, learn features of all surgeons trials and all expertise levels. On the contrary, LOUO requires to leave all trials for one surgeon for testing and they are unseen during training. Since there are only two experts and two intermediates in JIGSAWS, the model will behave poorly when trained with one expert and one intermediate and then tested with the only one remaining expert and intermediate.

3.10 Conclusion

In this chapter, we have described details of a dataset named JIGSAWS on dexterous surgical motion performed using the *DaVinci* Surgical System in a training laboratory. In addition to the data directly captured from the surgical robot, manual annotations for surgical gestures and skill have been included as well. We also presented various surgical scoring systems like OSATS, GEARS and GOALS that are currently used for surgical skills assessment. Several validation strategies to check robustness of the surgical skill assessment model that can be found in the literature have been presented as well. The JIGSAWS dataset will serve the purpose of validating all our networks presented in this thesis.

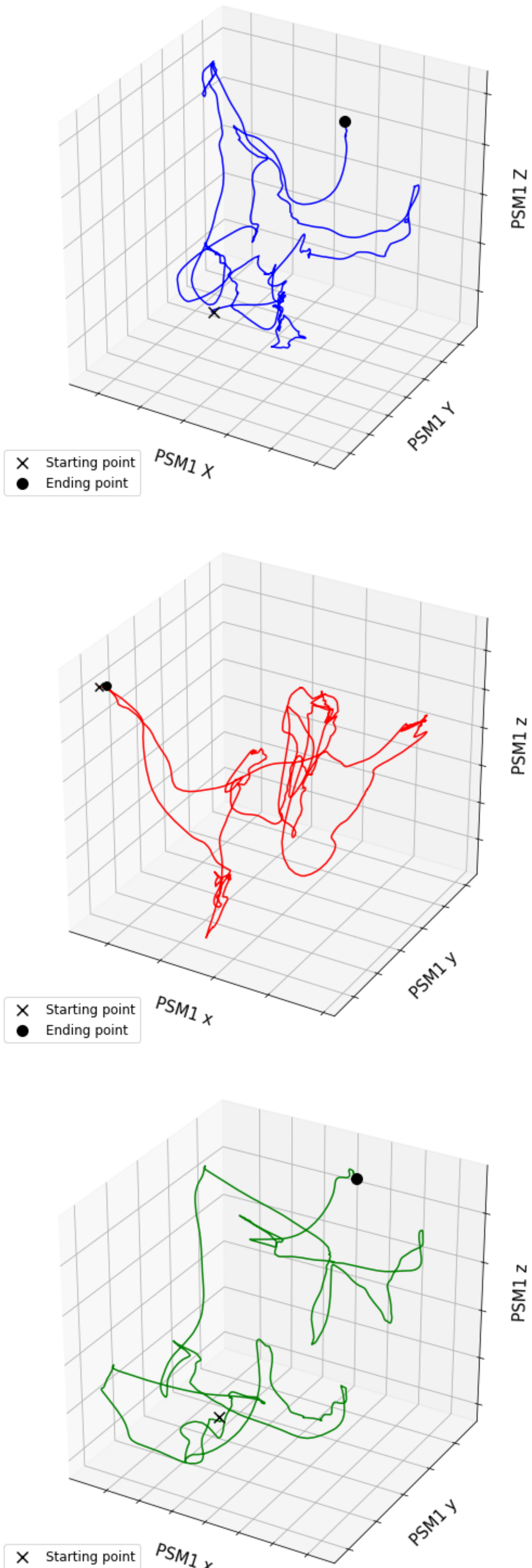

Figure 3.4: PSM1 x, y, z trajectories of subjects with different expertise levels performing the Knot-Tying task. From top to bottom: Novice, intermediate and expert.

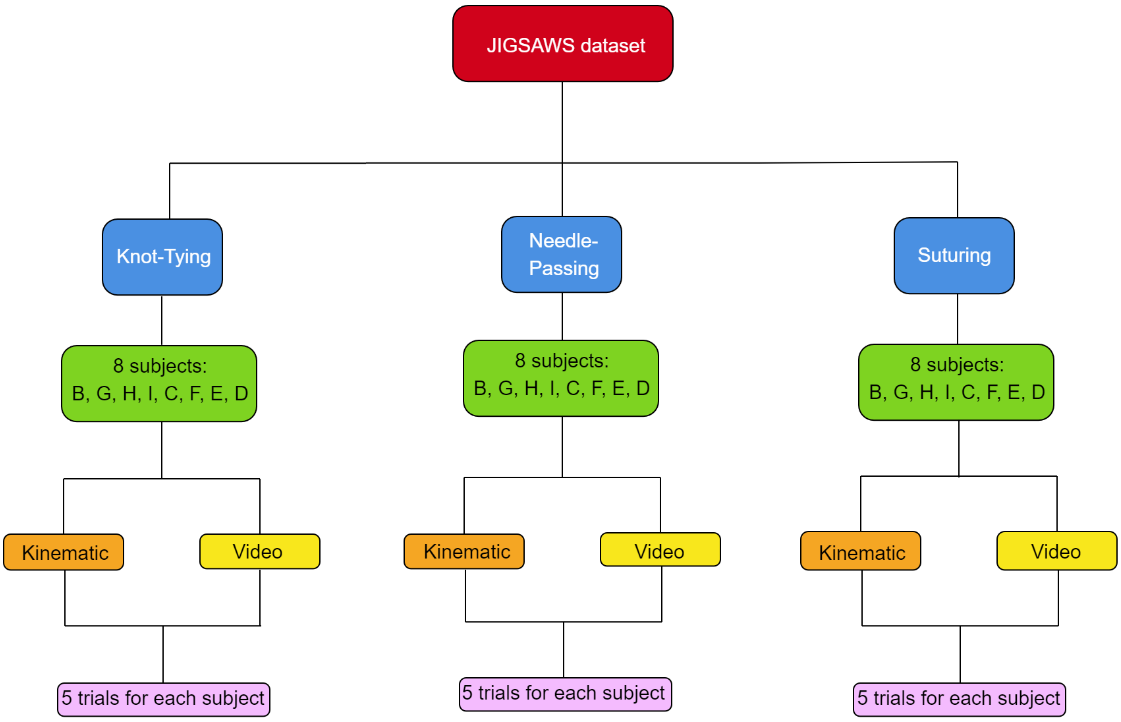

Figure 3.5: Organization of the JIGSAWS dataset.

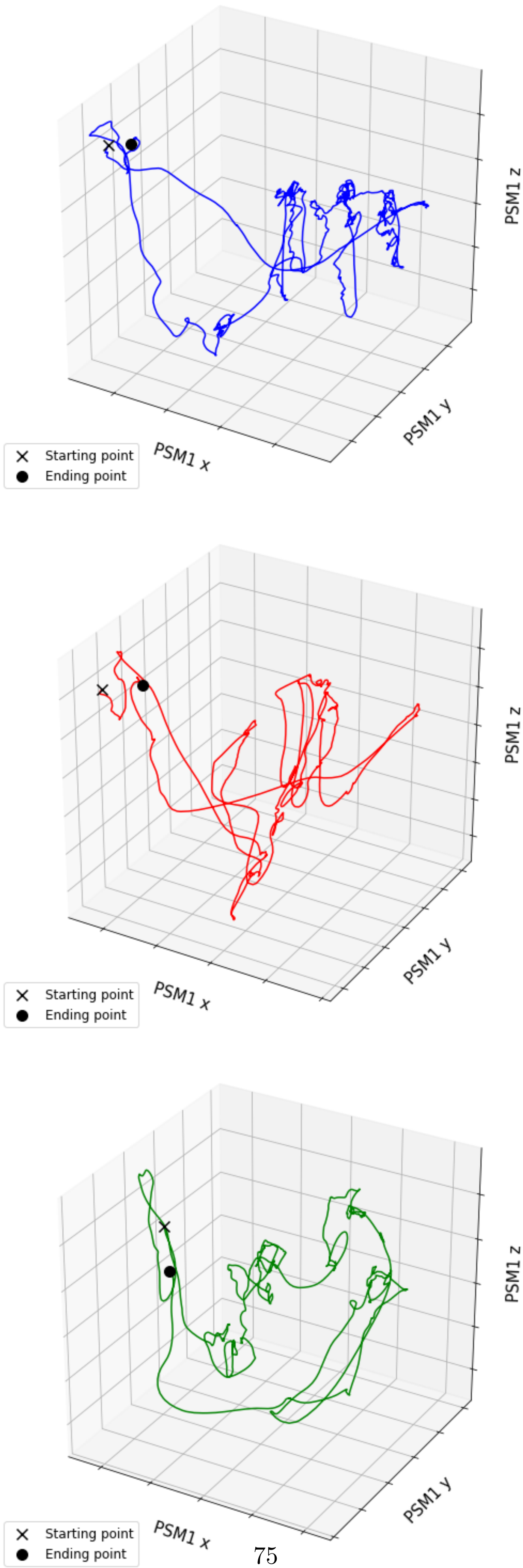

Figure 3.6: PSM1 x, y, z trajectories of subjects with different expertise levels performing the Needle-Passing task. From top to bottom: Novice, intermediate and expert.

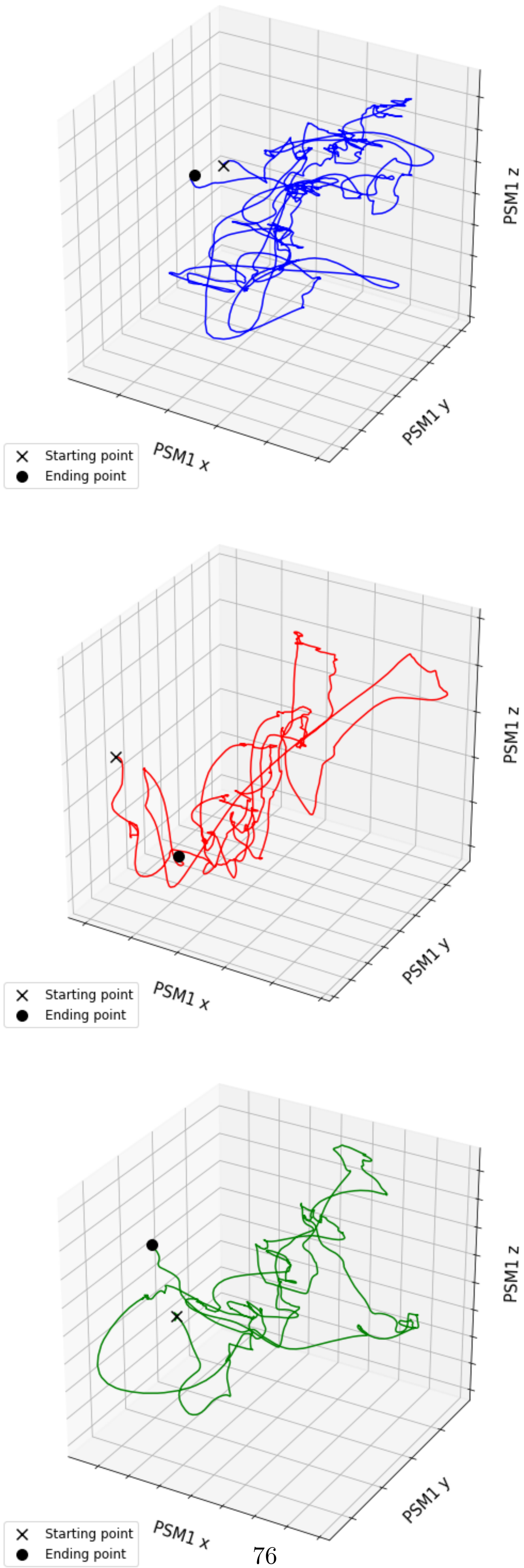

Figure 3.7: PSM1 x, y, z trajectories of subjects with different expertise levels performing the Suturing task. From top to bottom: Novice, intermediate and expert.

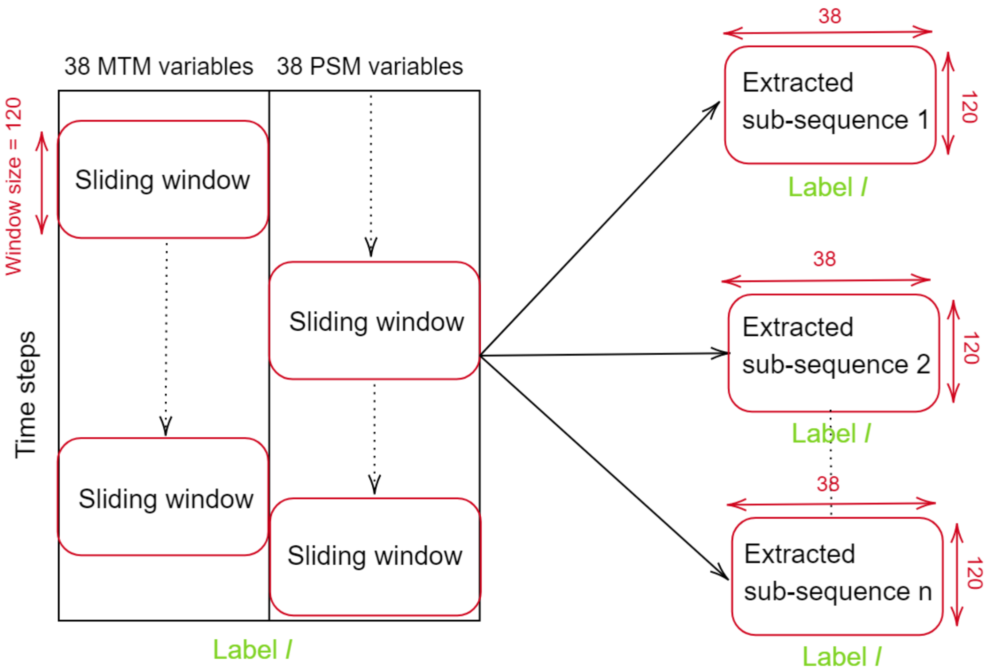

Figure 3.8: Diagram explaining the sliding window and tools data separation that are part of the data augmentation process.

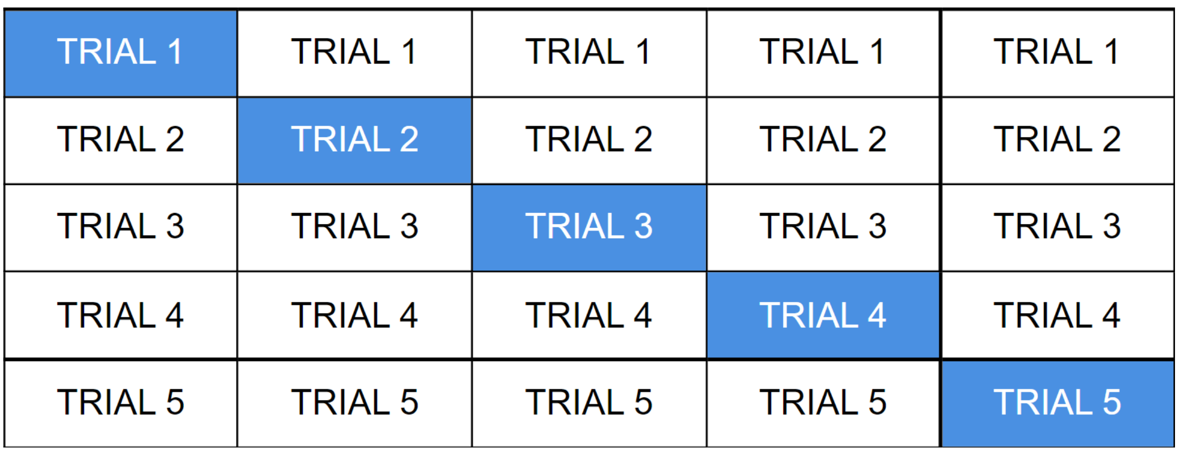

Figure 3.9: A diagram of the LOSO validation strategy: Each column represents a single training/testing set. Blue colored cells represents the trial that is left out for testing while the white cells represents the remaining trials used for training. All 5 trials shown here belongs to a single subject. The same process is applied to the 8 subjects.

Subject B	Subject B	Subject B	Subject B	Subject B	Subject B	Subject B	Subject B
Subject C	Subject C	Subject C	Subject C	Subject C	Subject C	Subject C	Subject C
Subject D	Subject D	Subject D	Subject D	Subject D	Subject D	Subject D	Subject D
Subject E	Subject E	Subject E	Subject E	Subject E	Subject E	Subject E	Subject E
Subject F	Subject F	Subject F	Subject F	Subject F	Subject F	Subject F	Subject F
Subject G	Subject G	Subject G	Subject G	Subject G	Subject G	Subject G	Subject G
Subject H	Subject H	Subject H	Subject H	Subject H	Subject H	Subject H	Subject H
Subject I	Subject I	Subject I	Subject I	Subject I	Subject I	Subject I	Subject I

Figure 3.10: A diagram of the LOUO validation strategy: Each column represents a single training/testing set. Each cell represents all data of a single subject. Green colored cells represents all trials of the subject that is left out for testing while the white cells represents the remaining subjects data used for training.

Trial 1	Trial 1	Trial 1	Trial 1	Trial 1	Trial 1	Trial 1	Trial 1
Trial 2	Trial 2	Trial 2	Trial 2	Trial 2	Trial 2	Trial 2	Trial 2
Trial 3	Trial 3	Trial 3	Trial 3	Trial 3	Trial 3	Trial 3	Trial 3
Trial 4	Trial 4	Trial 4	Trial 4	Trial 4	Trial 4	Trial 4	Trial 4
Trial 5	Trial 5	Trial 5	Trial 5	Trial 5	Trial 5	Trial 5	Trial 5
Trial 6	Trial 6	Trial 6	Trial 6	Trial 6	Trial 6	Trial 6	Trial 6
⋮	⋮	⋮	⋮	⋮	⋮	⋱	⋮
Trial n	Trial n	Trial n	Trial n	Trial n	Trial n	Trial n	Trial n

Figure 3.11: A diagram of the LOOCV validation strategy: Each column represents a single training/testing set. Each cell represents one trial of a given subject performing a given task. Each yellow colored cells represents one trial that is left out for testing while the white cells represents the remaining trials used for training.

Folder 1	Folder 1	Folder 1	Folder 1
Folder 2	Folder 2	Folder 2	Folder 2
. . .	. . .	. . .	. . .
Folder k	Folder k	Folder k	Folder k

Figure 3.12: A diagram of the K-fold cross validation strategy: Each column represents a single training/testing set. Each cell represents one subset of trials that can belong to different subject. Each red colored cells represents one folder (or subset) that is left out for testing while the white cells represents the remaining folders used for training.

Grade '1'	Grade '1'	Grade '1'	Grade '1'	Grade '1'
Grade '2'	Grade '2'	Grade '2'	Grade '2'	Grade '2'
Grade '3'	Grade '3'	Grade '3'	Grade '3'	Grade '3'
Grade '4'	Grade '4'	Grade '4'	Grade '4'	Grade '4'
Grade '5'	Grade '5'	Grade '5'	Grade '5'	Grade '5'

Figure 3.13: A diagram of the LOGO validation strategy: Each column represents a single training/testing outputs set. Each cell represents one grade of a selected OSATS, GEARS or GOALS criterion. Each purple colored cells represents one grade that is left out for testing (unseen output during the training step) while the white cells represents the remaining grades used for training.

Chapter 4

Reconstruction of soft surgical skills with simple baseline neural networks

4.1 Introduction

In this chapter, we present three simple baseline neural network in the automated surgical assessment field where we used small architectures of artificial neural networks. First, we used a simple Deep Feed-forward Neural Network (DNN) for its implementation simplicity and for its low-computational demand. DNNs are a type of artificial neural network that consists of multiple layers of interconnected nodes, or artificial neurons. Unlike recurrent neural networks (RNNs), DNNs process input data in a feedforward manner, meaning that information flows through the network in one direction, from input to output. DNNs can be trained using supervised learning techniques, where the network is fed a large dataset and the weights of the neurons are adjusted to minimize the error between the network's predictions and the true outputs. DNNs have been successfully applied in a variety of tasks, such as image recognition, natural language processing, and prediction problems. They have also shown promising results in surgical skills assessment, where they can be used to analyze video and kinematic data to provide objective evaluations of the surgeon's skills. But since the inputs are time series data, we believed that using a Recurrent Neural Network (RNN) would be more effective because of its feedback connections and thus, is more adapted to time-series. RNNs are a type of artificial neural network that are designed to process sequential data. Unlike feedforward neural networks, which process input data in one direction, RNNs have feedback connections, allowing information to be passed from one step of the sequence to the next. This allows RNNs to maintain a hidden state that can capture information about the context of the sequence and make predictions based on that context. RNNs have been successfully applied to a variety of tasks, including natural language processing, speech recognition, and time series prediction. In surgical skills assessment, RNNs can be used to analyze kinematic data collected during procedures to assess the skill level of the surgeon. The ability of RNNs to capture the context and dependencies in sequential data makes them a powerful tool for evaluating surgical performance. Indeed, the RNN yielded more accurate and plausible results but this type of neural network is subject to two considerable problems the vanishing and exploding gradients. These problems occurs because the RNN is capable of capturing information in short time intervals and is unable to learn long-term dependencies. Finally, to overcome these problems, we employed a Long Short Term Memory Network which is an enhanced RNN and it is capable of capturing information widely spaced in time. The backward feedback allow the LSTM to take into account length-customizable previous entries. This specificity gives the LSTM an advantage over the RNN. In the context of surgery tasks, this enable us to take into account the evaluation of current gesture together with former executed gestures. At the end, results obtained with the LSTM surpassed all the previous results. We describe briefly, the common methodology shared by each paper and compare the obtained results.

4.2 General training approach for task-specific surgical skills using real soft scores

We present here the general adopted approach in our previous works covered in this chapter. The purpose of our work is to design an architecture able to provide real soft scores scales of surgical skills. The suggested approach relies on feeding JIGSAWS kinematic data to three independent deep neural networks models based on various architectures. Each of these networks is responsible for the assessment of all subjects but in only one of the three basic surgical tasks included in JIGSAWS. The way we have chosen to assess subjects on each task relies on transforming JIGSAWS expertise labels into meaningful real soft scores. These real soft scores reflect the global quality of the performed operation. Therefore, we give a complete score of 1/1 if a sample of expert data is given as input, an average score of 0.7/1 if a sample of intermediate data is given as input and a low score of 0.4/1 if a sample of beginner data is given as input. The intuition here is to inform the networks about the scores deserved by each subject according to the corresponding expertise level. We have chosen these ratings according to the number of hours in practicing of each subject (more than 100h for the experts, between

10h and 100h for the intermediate subjects and less than 10h for the novices).

Inputs: The inputs are kinematic data from the JIGSAWS dataset. Each kinematic data sample represents in rows the numbers of frames i.e. the duration of the task completed by the surgeon and in columns, 76 kinematic variables of the DaVinci robot tools.

Outputs: The output of each neural network is a scalar rating between 0 and 1 of the performance of a surgeon who completed one of the three tasks.

The general training approach can be summarized in the following items:

- We design three independents neural network models based on several architectures to evaluate the Knot-Tying task, the Needle-Passing task and the Suturing task independently.
- Each network receives some kinematic data from the JIGSAWS database of surgeons with different expertise levels for the training procedure and some remaining data for the testing procedure. The desired outputs depend on whether the current subject is a beginner, intermediate or expert.
- After the training step, the model will be able to make predictions about unseen kinematic data by giving and adequate score to the data sample of the tested surgeon.

4.3 Task-specific surgical skill assessment using feed forward deep neural networks (DNN)

A feedforward neural network is a type of artificial neural network in which the information flows in only one direction, from the input layer to the output layer. A deep feedforward neural network has multiple hidden layers in addition to the input and output layers. The mathematical equations for a deep feedforward neural network can be represented as:

For each layer $l \in 1, 2, \ldots, L$:

$$z_l = W_l \cdot a_{l-1} + b_l \tag{4.1}$$

$$a_l = g(z_l) \tag{4.2}$$

Where:

- L is the total number of layers in the network (including the input and output layers).
- z_l is the weighted sum of the inputs to layer l.
- W_l is the weight matrix for layer l.
- a_{l-1} is the activation vector of the previous layer, $a_{l-1} = g(z_{l-1})$.
- b_l is the bias vector for layer l.
- g is the activation function, such as ReLU or sigmoid.

The input layer of the network has no activation function, and the output layer may have a different activation function than the hidden layers, depending on the task at hand. The final output of the network is given by:

$$y = a_L \tag{4.3}$$

Where a_L is the activation vector of the output layer.

We used three independent fully connected DNNs and each one is intended to learn kinematic data of one task only. As depicted in figure 4.1, each of the three networks is a fully connected neural network composed of three hidden layers each of which contains 20 nodes. The input layer contains 69920 nodes because it is the size of each kinematic data used for the training and the test. The used activation function in the hidden layer and the output layer is the Sigmoid function.

4.3.1 Experimental setup

In order to train these networks, we used the backpropagation algorithm. To compute the training error, we have chosen the cross-entropy loss function. The desired output depend on the subject who is operating as described in section 4.2. Concerning the choice of the number of hidden layers and the number of nodes in each hidden layer, we took this configuration from [81]. The author of the book used this network for a classification problem: classifying

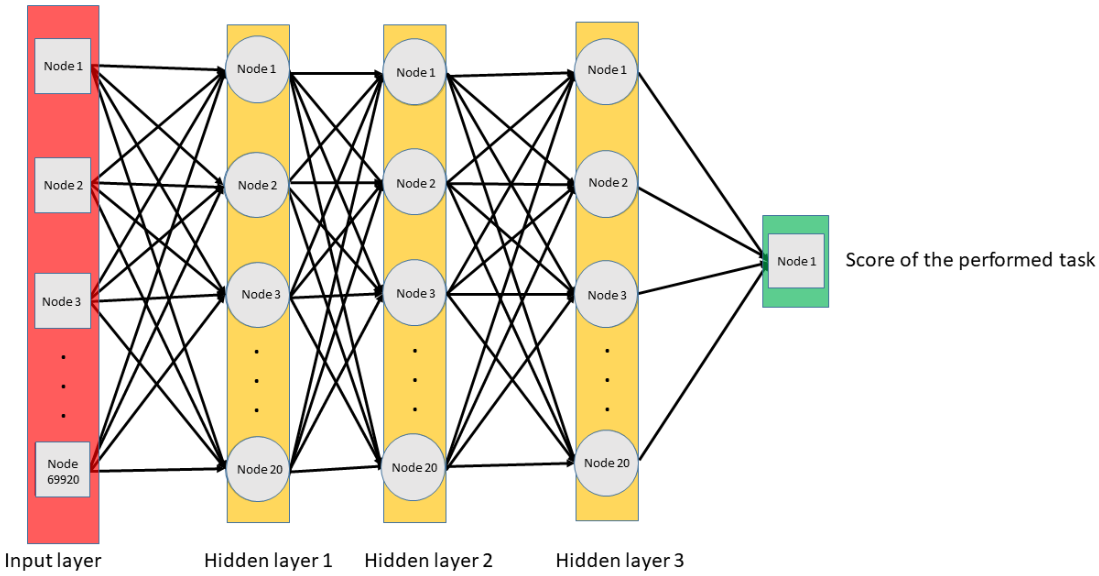

Figure 4.1: Architecture of a single feed-forward neural network for specific task evaluation.

five digits of dimension 5x5 into five categories, which are numbers from 1 to 5. This network yielded great results and thus, we decided to try it to apply our approach. We did not use other configurations because the purpose is to prove that we can assess surgical skills by using specifically one neural network to assess performance of trainees on one task with real soft scores prediction.

4.3.2 Results and discussion

We run the evaluation procedure on the test data. It contains two intermediates and four novices. The intermediate subjects are labeled with the letters C and F and the novice subjects are labeled with the letters B, G, H and I. The test data contains one trial of each intermediate and novice subject performing each task. The following table shows the scores of each novice and intermediate subject after performing the three tasks. For more convenience, we abbreviate the names of the tasks Knot-Tying, Needle-Passing and Suturing respectively as KT, NP and ST. We labeled any task performed by any subject by the abbreviation of the task concatenated with the label representing the subject followed by the number of the trial. For example, KTB4 stands for 'novice subject B performing Knot-Tying task, trial 4'.

As it can be seen in table 4.1, the DNNs yielded globally good results. Novice subject (0.4 outputs) were easier for the network to predict than the intermediate subjects (0.7 outputs). Lack of data for intermediate subjects might be the cause (10 intermediate data samples

Subject and task	Target output	Predicted output
KTB4	0.400	**0.390**
KTC4	0.700	0.877
KTF4	0.700	0.509
KTG4	0.400	**0.365**
KTH4	0.400	**0.377**
KTI5	0.400	0.508
NPB4	0.400	**0.435**
NPC4	0.700	1.000
NPF4	0.700	**0.644**
NPG4	0.400	0.500
NPH4	0.400	0.311
NPI4	0.400	**0.377**
STB4	0.400	0.269
STC4	0.700	**0.782**
STF4	0.700	0.452
STG4	0.400	**0.468**
STH4	0.400	0.273
STI4	0.400	0.345

Table 4.1: Predictions yielded by the three DNNs for the three surgical tasks performed by all novice and intermediate subjects. Best results are show in bold.

versus 20 novice data samples per surgical task). In addition, we can see that particularly, the two intermediate subjects C and F got better ratings than all the novice subjects B, G, H and I; which seems logical since intermediate subjects have more time experience in the domain than the novice subjects. Speaking about the time required for training the networks of both setups; the training loop of each of the three networks is performed on 1000 epochs. Consequently, the training of the three networks is completed after 3000 epochs. The next section is about presenting a similar setup and other results obtained by a recurrent neural network architecture.

4.4 Task-specific surgical skill assessment using recurrent neural networks

An artificial neural network that employs sequential data or time series data is known as a recurrent neural network (RNN). These deep learning algorithms are frequently employed for ordinal or temporal issues, such as language translation, natural language processing (nlp), speech recognition, and image captioning. Recurrent neural networks (RNNs) use training data to learn, just like feedforward and convolutional neural networks (CNNs) do. They stand out due to their "memory," which allows them to affect the current input and output by using data from previous inputs. Recurrent neural networks' outputs are dependent on the previous parts in the sequence, unlike typical deep neural networks, which presume that inputs and outputs are independent of one another. RNNs have been widely used into several sequence modeling of dynamic systems. In the context of surgery, the use of RNNs with kinematic data can bring several advantages:

- Motion prediction: RNNs can be trained on historical surgical kinematic data to make predictions about future motions, which can be useful in tasks such as motion planning or surgical skill assessment.
- Motion analysis: RNNs can be used to analyze surgical kinematic data to extract meaningful features and patterns, which can be used to identify motion styles, detect errors, or evaluate surgical skills.
- Anomaly detection: RNNs can be trained to identify anomalies or deviations in surgical kinematic data, which can be useful in detecting errors or deviations from normal surgical procedures.
- Personalization: RNNs can be trained on individual surgeons' kinematic data to generate personalized models that can be used for surgical skill assessment or motion prediction.

Overall, the use of RNNs with surgical kinematic data can bring significant benefits for surgical analysis and monitoring, leading to improved surgical outcomes and reduced surgical errors.

Structure: A RNN is composed of a set of inputs x_t hidden states, outputs and a nonlinear block. A non-linear block in the RNN takes an input and a hidden state to produce an output and a hidden state which feeds the next nonlinear block with an input and so forth (see figure 4.2). In the case of RNN with feedback, which our case, the output is back connected to the nonlinear block.

The relation involved by the nonlinear block can be written as:

$$h_t = tanh(W_x X_t + W_h h_{t-1} + W_m m_t + b) \tag{4.4}$$

W_x, W_h and W_m are weights to be learnt. b is the bias. To obtain the output score, we use a linear layer:

$$y_t = W_y m_t + b_y \tag{4.5}$$

W_y and b_y being also scalars to be learnt.

Figure 4.2 shows a diagram of single model based on the RNN architecture for surgical that is responsible for assessing subject on one of the three tasks.

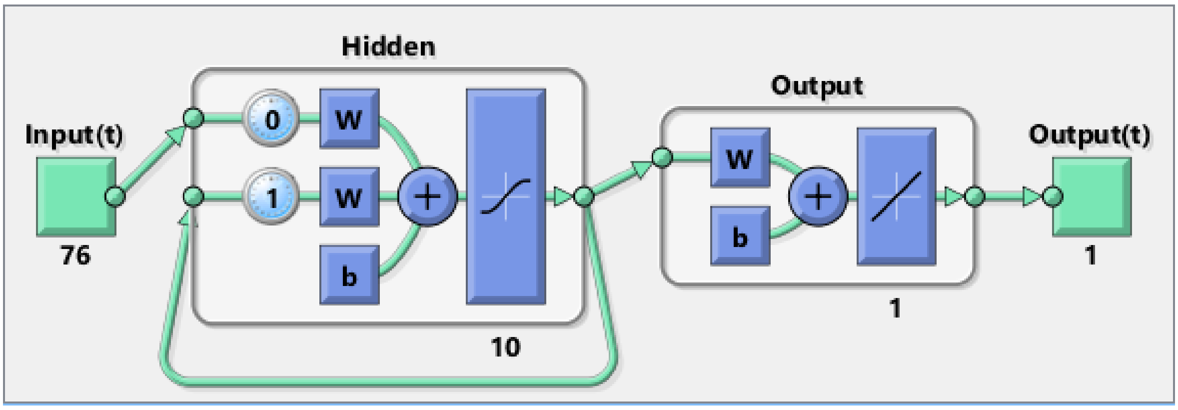

Figure 4.2: Architecture of a single recurrent neural network for specific task evaluation.

4.4.1 Experimental setup

As for the employed setup that uses DNNs, We used three independent recurrent neural networks. Each one is intended to assess one trial of each novice and intermediate subject of the JIGSAWS dataset on one surgical task only. Therefore, each network takes as input, learning kinematic data of one of the three tasks. Each RNN contains 10 hidden layers. The training finishes after 50 epochs. The Levenberg-Marquardt algorithm is employed for the training process and the performance is calculated through the Mean Squared Error. hyperbolic tangent activation function is used in hidden layers and a linear function is used to output the score. Table 4.2 shows predictions results of the models based on the presented RNN architecture in comparison with the previously shown DNNs predictions.

4.4.2 Results and discussion

As it can be seen in table 4.2, RNN models provided more accurate results than the DNNs in the three surgical tasks. While the RNNs predicted novice score subjects better than the DNNs, it clearly outperformed the DNNs on the intermediate subjects, C and F, where the DNN architecture seems to struggle in predicting the right output. Indeed, the lack of intermediate subjects affected RNNs less than the DNNs thanks to feedback connections

Subject and task	Target output	Predicted output (DNN)	Predicted output (RNN)
KTB4	0.400	0.390	**0.401**
KTC4	0.700	0.877	0.992
KTF4	0.700	0.509	**0.697**
KTG4	0.400	0.365	**0.399**
KTH4	0.400	0.377	**0.402**
KTI5	0.400	0.508	**0.397**
NPB4	0.400	0.435	**0.401**
NPC4	0.700	1.000	**0.703**
NPF4	0.700	0.644	**0.665**
NPG4	0.400	0.500	**0.400**
NPH4	0.400	0.311	**0.399**
NPI4	0.400	0.377	**0.405**
STB4	0.400	0.269	**0.399**
STC4	0.700	0.782	**0.693**
STF4	0.700	0.452	0.596
STG4	0.400	0.468	**0.400**
STH4	0.400	0.273	**0.407**
STI4	0.400	0.345	**0.400**

Table 4.2: Predictions yielded by the three RNNs for the three surgical tasks performed by all novice and intermediate subjects. Results are compared with DNNs prediction results. Best results are show in bold.

contained in the RNN structure as well as the following advantages: (1) Their ability to model sequential data. This makes them particularly well-suited for time series data, where the order of data points is important. (2) RNNs have a memory mechanism that allows them to retain information from previous time steps, which can be useful in capturing patterns in time series data. (3) RNNs can handle temporal dependencies in time series data, where the output at a given time step depends on the inputs from previous time steps. This can be useful in modeling time-dependent trends and patterns in the data. However, it is important to note that while RNNs are well-suited for time series analysis, they can be more difficult to train and can suffer from the vanishing gradient and exploding gradient problems, where gradients become respectively very small or huge as they propagate through the network [82].

4.5 Task-specific surgical skill assessment using Long Short Term memory networks

The Long Short-Term Memory (LSTM) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. This is a popular RNN architecture, which was introduced by Sepp Hochreiter and Juergen Schmidhuber as a solution to vanishing gradient problem [83]. Unlike standard feedforward neural networks, LSTM has feedback connections (known as memory) allowing it to learn long term dependecies. These backward feedbacks allow the LSTM to take into account length-customizable previous entries. This specificity gives the LSTM an advantage over the RNNs. In the context of surgery tasks, this enable us to take into account the evaluation of current gesture together with former executed gestures. A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell. LSTM networks are suited to deal with time series data and to overpass two major problems possibly encountered when training traditional RNNs : the vanishing gradient and exploding gradient. These two problems emerge when the RNN is confronted with long term dependencies (important events in a time series). Figure 4.3 depicts the diagram of the proposed LSTM architecture.

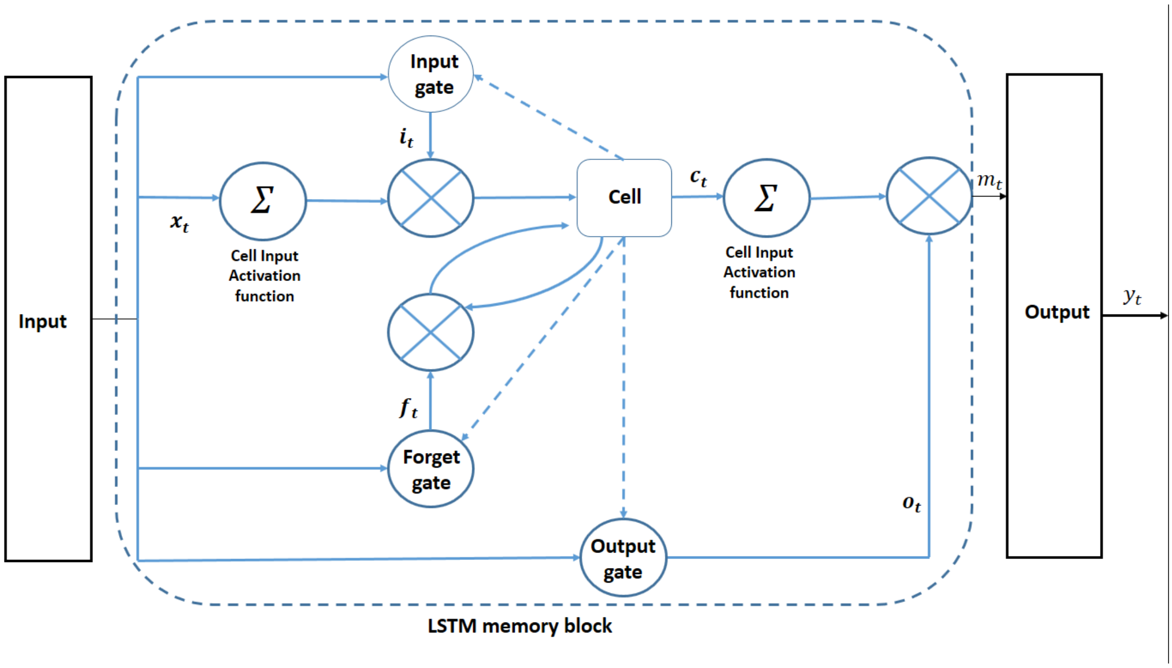

Figure 4.3: Diagram of a Long Short Term Memory with a forget gate.

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \tag{4.6}$$

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \tag{4.7}$$

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \tag{4.8}$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \tag{4.9}$$

$$h_t = o_t \circ \sigma_h(c_t) \tag{4.10}$$

Where :

- $x_t \in R^d$: Input vector to the LSTM unit.
- $f_t \in R^h$: Forget gate's activation vector.
- $i_t \in R^h$: Input gate's activation vector.
- $o_t \in R^h$: Output gate's activation vector.
- $h_t \in R^h$: Hidden state vector also known as output vector of the LSTM unit.
- $c_t \in R^h$: Cell state vector.
- $W \in R^{h \times d}, U \in R^{h \times h}$ and $b \in R^h$: Weight matrices and bias vector parameters which need to be learned during training.
- σ_g: Sigmoid function.
- σ_c: Hyperbolic tangent function.
- σ_h: Another hyperbolic tangent function.

4.5.1 Internal composition of the three networks

Since our three models have identical architectures, we describe only one of them:

- A LSTM layer of 100 units with an input dimension of 920 rows and 76 columns (size of each kinematic data), activation function : hyperbolic tangent.
- First hidden dense layer with 40 units, activation function : sigmoid.
- Second hidden dense layer with 80 units, activation function : sigmoid.
- Second hidden dense layer with 80 units, activation function : sigmoid.

- Output dense layer with a single unit carrying the performance score between 0 and 1, activation function : sigmoid.

4.5.2 training and testing procedure

We took 3 trials of each task performed by each subject as training data. The remaining trials have been left for testing the network. The loss function chosen is the binary crossentropy since the output is a score between 0 and 1. We trained our models on a certain batch size over 500 epochs and the Adam optimizer has been used to update weights.

Subject and task	Target output	Predictions (DNN)	Predictions (RNN)	Predictions (LSTM)
KTB4	0.400	0.390	**0.401**	**0.417**
KTC4	0.700	0.877	0.992	**0.756**
KTF4	0.700	0.509	**0.697**	0.521
KTG4	0.400	0.365	**0.399**	**0.403**
KTH4	0.400	0.377	**0.402**	**0.444**
KTI5	0.400	0.508	**0.397**	**0.400**
NPB4	0.400	0.435	**0.401**	**0.457**
NPC4	0.700	1.000	**0.703**	0.899
NPF4	0.700	0.644	**0.665**	**0.661**
NPG4	0.400	0.500	**0.400**	**0.400**
NPH4	0.400	0.311	**0.399**	**0.429**
NPI4	0.400	0.377	**0.405**	**0.431**
STB4	0.400	0.269	**0.399**	**0.423**
STC4	0.700	0.782	**0.693**	0.894
STF4	0.700	0.452	0.596	0.460
STG4	0.400	0.468	**0.400**	0.353
STH4	0.400	0.273	**0.407**	**0.429**
STI4	0.400	0.345	**0.400**	**0.437**

Table 4.3: Predictions yielded by the three LSTMs for the three surgical tasks performed by all novice and intermediate subjects. Results are compared with both DNNs and RNNs prediction results. Best results are show in bold.

Task/Neural network	DNN	RNN	LSTM
Knot-Tying	0.82	0.82	0.82
Needle-Passing	0.82	0.82	0.82
Suturing	0.62	0.84	0.82

Table 4.4: Spearman rank correlation results between true and predicted outputs (No validation scheme).

4.5.3 Surgical skill evaluation and discussion

After running the evaluation criteria on the test data. We have chosen to assess intermediate and novice subjects only and we took one trial to assess on each task. Table 4.3 show the scores of each novice and intermediate subject after performing the three tasks. The results prove that the RNNs and the LSTMs yielded the best performance evaluation. Indeed, we can see that particularly, the two intermediate subjects got better global score than all the novice subjects with the RNNs and the LSTMs rather than with the DNNs. These results are plausible since intermediate subjects have more time experience in the domain and are more skilled than the novice subjects. RNNs and LSTMs surpass the DNNs because, in this case, we are dealing with times series data since kinematic data represent positions and velocities of the DaVinci robot at each time step. LSTMs and RNNs are the most suitable type of neural network models to deal with such type of data because of their ability to learn from both current and previous state of the network during the training step. Moreover, LSTMs have the ability to counter the vanishing and exploding gradient problems that may occur while using a RNN. This is why we consider, according to obtained results in the tables, that the RNNs and the LSTMs behave better than the DNNs for the surgical assessment task. In addition, we computed the Spearman rank correlation coefficient ρ between true and predicted outputs. The high values of ρ (close to 1) indicate that distributions of true and predicted outputs follow the same trend and that the trained models well fitted the test data.

4.6 Surgical skill assessment based on the Leave-One-Supertrial-Out validation scheme

While the previously presented results have proven the effectiveness of the proposed neural networks simple architectures in predicting exact surgical skills scores on only one selected trial of each JIGSAWS subject, generalizability of the networks is still not verified: the generalizability of neural networks refers to the ability of a neural network to correctly apply learned patterns to new, unseen data and make accurate predictions. In other words, it measures the degree to which a neural network can transfer the knowledge it has learned from the training data to unseen data. A network that is highly generalizable is able to make accurate predictions on new data, while a network with poor generalizability is likely to overfit to the training data and make inaccurate predictions on new data [84]. Consequently, a validation strategy is mandatory to check whether a neural network model is able to generalize on different sets of unseen data. In this section, we present predictions results with the previously presented DNN, RNN and LSTM architectures obtained through the Leave-One-Supertrial-Out (LOSO) validation scheme that is proposed in JIGSAWS [19] and

TRIAL 1	TRIAL 1	TRIAL 1	TRIAL 1	TRIAL 1
TRIAL 2	TRIAL 2	TRIAL 2	TRIAL 2	TRIAL 2
TRIAL 3	TRIAL 3	TRIAL 3	TRIAL 3	TRIAL 3
TRIAL 4	TRIAL 4	TRIAL 4	TRIAL 4	TRIAL 4
TRIAL 5	TRIAL 5	TRIAL 5	TRIAL 5	TRIAL 5

Figure 4.4: A diagram of the LOSO validation strategy: Each column represents a single training/testing set. Blue colored cells represents the trial that is left out for testing while the white cells represents the remaining trials used for training. All 5 trials shown here belongs to a single subject.

is defined in chapter 3. For more convenience, the LOSO diagram is depicted again in figure 4.4 to recall the process.

4.6.1 Training and testing procedures

While the experimental setup remains the same as presented previously for each neural network architecture, the training/testing procedure differs: This time, in regard to the LOSO strategy, a neural network model is designed, trained and tested using kinematic data from each of the 5 sets depicted in figure 4.4. Therefore, 5 models are designed based on each architecture for each surgical task. The training is stopped when the validation loss stops improving and the best model is saved for predictions. The goal of the LOSO is to verify robustness of the network by testing it on each trial of a given subject. At the end, model's average performance of prediction over the 5 training/testing set is computed. The LOSO different sets are defined as follows:

- **Set 1:** Training with trials 2,3,4 and 5, testing on trial 1.
- **Set 2:** Training with trials 1,3,4 and 5, testing on trial 2.
- **Set 3:** Training with trials 1,2,4 and 5, testing on trial 3.
- **Set 4:** Training with trials 1,2,3 and 5, testing on trial 4.
- **Set 5:** Training with trials 1,2,3 and 4, testing on trial 5.

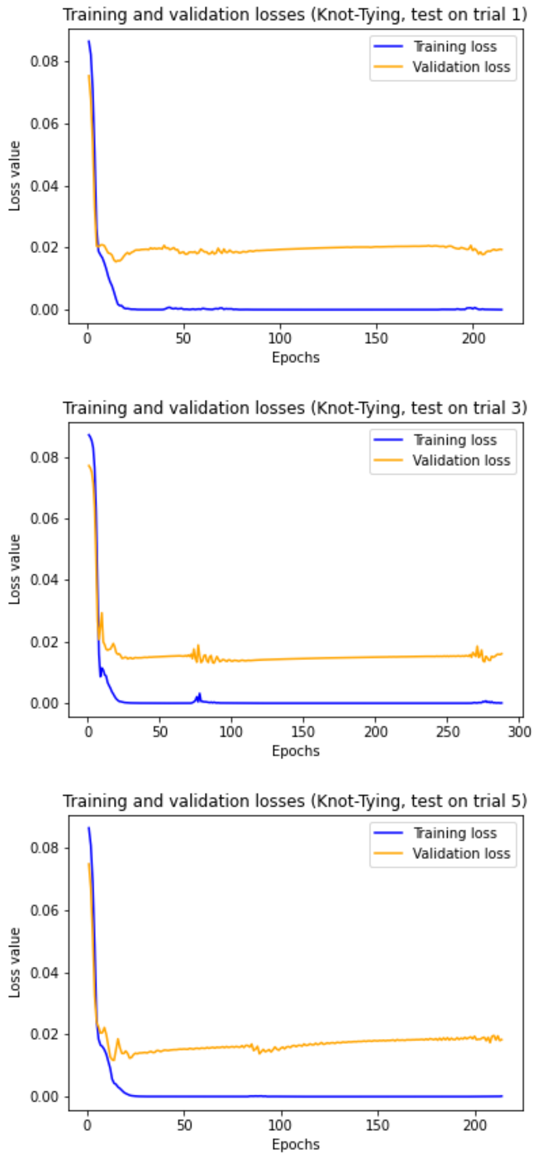

Figure 4.5: Training and validation losses curves while training and testing the DNN on trials 1,3 and 5 of all subjects performing the Knot-Tying task.

4.6.2 Results and discussion

The computed averages for each task and each neural network architecture are reported in table 4.5. Figures from 4.5 to 4.13 depict the evolution of both training and validation loss functions while training each network architecture on LOSO sets 1, 3 and 5 for each surgical task. Training and validation loss curves of LOSO sets 2 and 4 are not shown so as not to overwhelm the manuscript. According to figures 4.5, 4.6 and 4.7, the curves clearly indicate that the DNN is overfitting on the training data no matter which LOSO set is employed and which surgical task is treated. Figures 4.8, 4.12 and 4.10 shows that the RNN-based models

Subject and task	Target output	Predictions (DNN)	Predictions (RNN)	Predictions (LSTM)
KTB	0.40	0.41	0.35	0.37
KTC	0.70	0.86	**0.78**	**0.75**
KTD	1.00	**0.93**	**0.93**	**0.90**
KTE	1.00	**0.92**	**0.95**	**0.99**
KTF	0.70	0.52	0.34	0.54
KTG	0.40	**0.43**	**0.46**	**0.43**
KTH	0.40	0.50	**0.42**	**0.41**
KTI	0.40	0.47	**0.37**	**0.39**
NPB	0.40	**0.48**	0.20	0.18
NPC	0.70	0.93	0.96	0.87
NPD	1.00	**0.91**	**0.68**	0.81
NPE	1.00	**0.90**	**0.88**	**0.89**
NPF	0.70	0.54	0.55	0.59
NPH	0.40	**0.49**	**0.49**	0.52
NPI	0.40	0.56	**0.43**	**0.48**
STB	0.40	0.50	0.55	**0.49**
STC	0.70	0.81	0.81	0.80
STD	1.00	**0.94**	**0.92**	**0.95**
STE	1.00	**0.93**	**0.96**	0.87
STF	0.70	**0.60**	**0.52**	**0.61**
STG	0.40	**0.44**	**0.42**	**0.44**
STH	0.40	**0.44**	**0.41**	**0.38**
STI	0.40	**0.44**	**0.42**	**0.46**

Table 4.5: Average score performance prediction over the 5 trials of all subjects performing the three tasks while testing the three neural network architectures under the LOSO validation scheme. Best results are shown in bold.

behaves largely better than the DNNs and achieve good validation loss after considerably less epochs than the DNNs on each LOSO set and each surgical task. This further supports more the idea of RNN and time-series data adequacy. While the LSTM have been proven to be able to overcome RNNs limitation, in this case, LSTM yielded slightly less accurate predictions than the RNNs but are still better than the DNNs. This might be because LSTMs have a complex architecture and need more data to process or need a data augmentation pipeline to increase the starting amount of kinematic data. Results reported in table 4.5 show that predictions obtained with the LOSO scheme are fine but less accurate than predictions obtained with testing on a single trial for each subject (table 4.3). On the one hand, this means that the models fails to accurately predict scores of some trials but on the other hand, this LOSO setup represents a more 'real-world' scenario than the previous setup that involves testing a single trial per subject. Globally, the obtained results with the LOSO scheme clearly indicate that subject kinematic data have distinguishing patterns for expertise level and the

Task/Neural network	DNN	RNN	LSTM
Knot-Tying	0.93	0.62	0.93
Needle-Passing	0.70	0.79	0.87
Suturing	0.95	0.85	0.93

Table 4.6: Spearman rank correlation results between true and predicted outputs (LOSO).

proposed models are capable of regressing surgical performance scores based on the expertise levels. The remaining training/validation curves of trials 2 and 4 for each subject and each task are in the appendix (5).

4.7 Conclusion

In this chapter, we presented our previous contributions which consisted in establishing, training and testing three neural network architectures, DNN, RNN and LSTM with kinematic data of the JIGSAWS subjects and comparing their outcomes using identical sets of training and testing kinematic data for the Knot-Tying, Needle-Passing and Suturing tasks. Two validation setups have been compared: the first one consists in leaving out a single trial of each subject for testing while the second setup, LOSO, consists in performing a rotation of tests on the 5 trials of each subject. On the one hand, the experimental results have shown that the RNNs and LSTMs yielded the most logical and plausible results by giving the right score to the right skill level. On the other hand, DNNs showed clear signs of overfitting that might be caused by the lack of kinematic data in the JIGSAWS dataset. Thus, a data augmentation pipeline would greatly benefits the dataset. While the study of this chapter is based on attributing adequate real soft scores according to self-proclaimed expertise levels of the JIGSAWS surgeons [19], it stills suffers from subjectivity. A objective scoring system based on various surgical criteria would greatly increase objectivity of the surgical skill assessment. Consequently, we present in the next chapter, a new method that involves evaluating surgical skills on the basis of OSATS criteria that have been introduced in chapter 3 using a more complex neural network architecture.

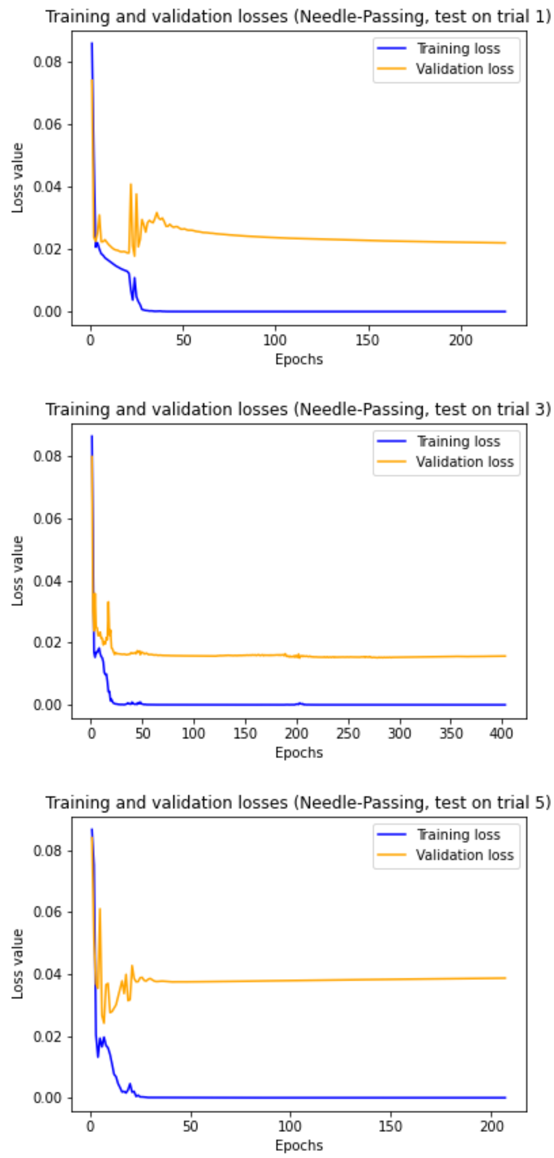

Figure 4.6: Training and validation losses curves while testing the DNN on trials 1,3 and 5 of all subjects performing the Needle-Passing task.

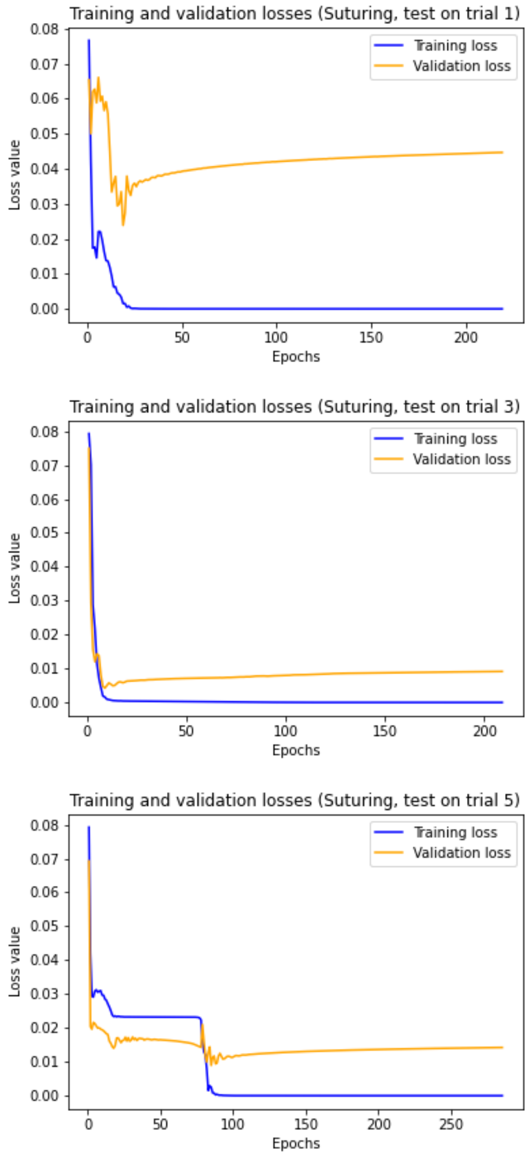

Figure 4.7: Training and validation losses curves while testing the DNN on trials 1,3 and 5 of all subjects performing the Suturing task.

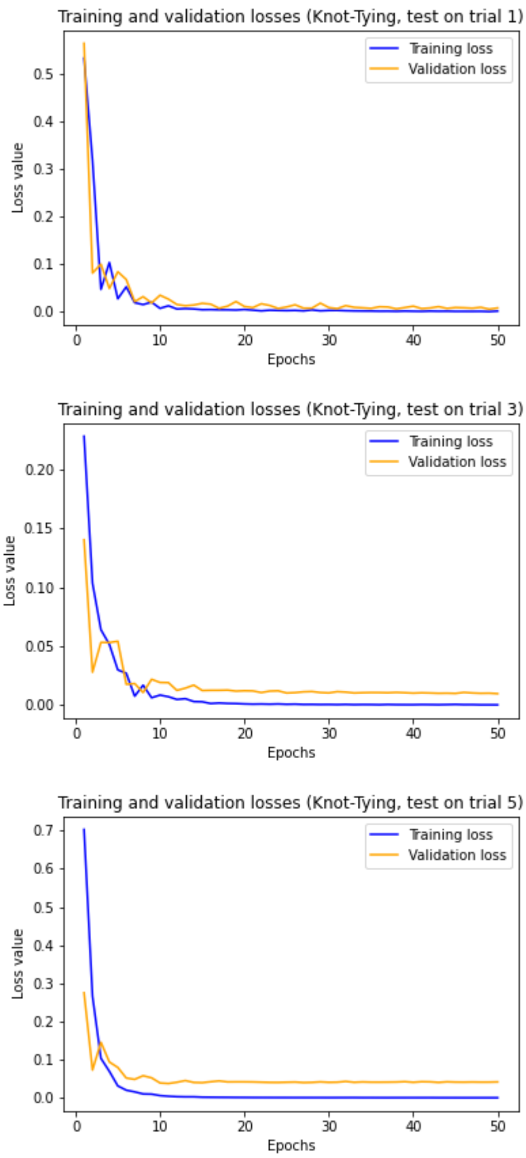

Figure 4.8: Training and validation losses curves while testing the RNN on trials 1,3 and 5 of all subjects performing the Knot-Tying task.

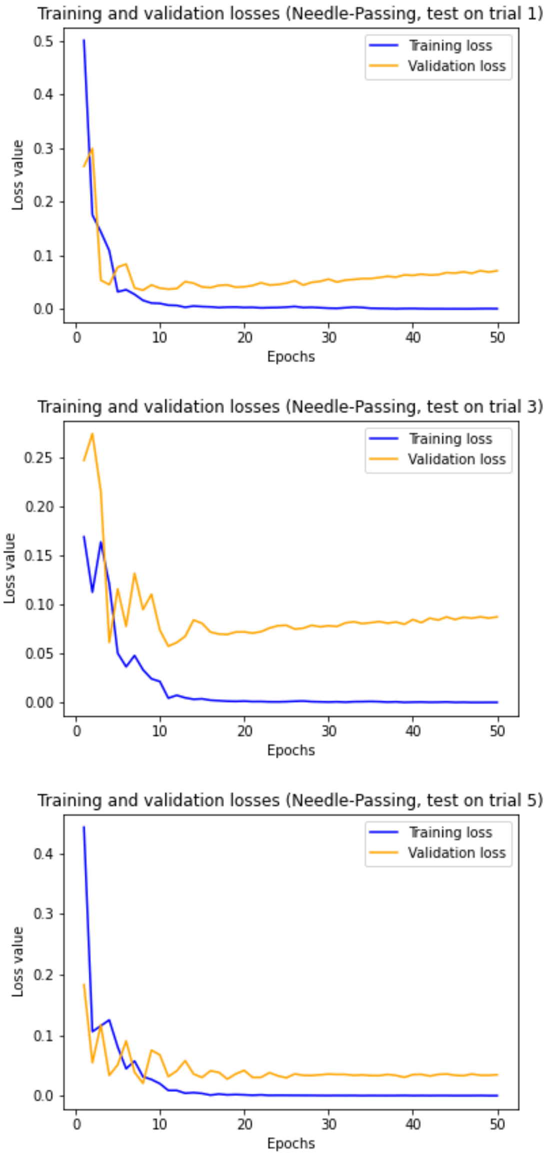

Figure 4.9: Training and validation losses curves while testing the RNN on trials 1,3 and 5 of all subjects performing the Needle-Passing task.

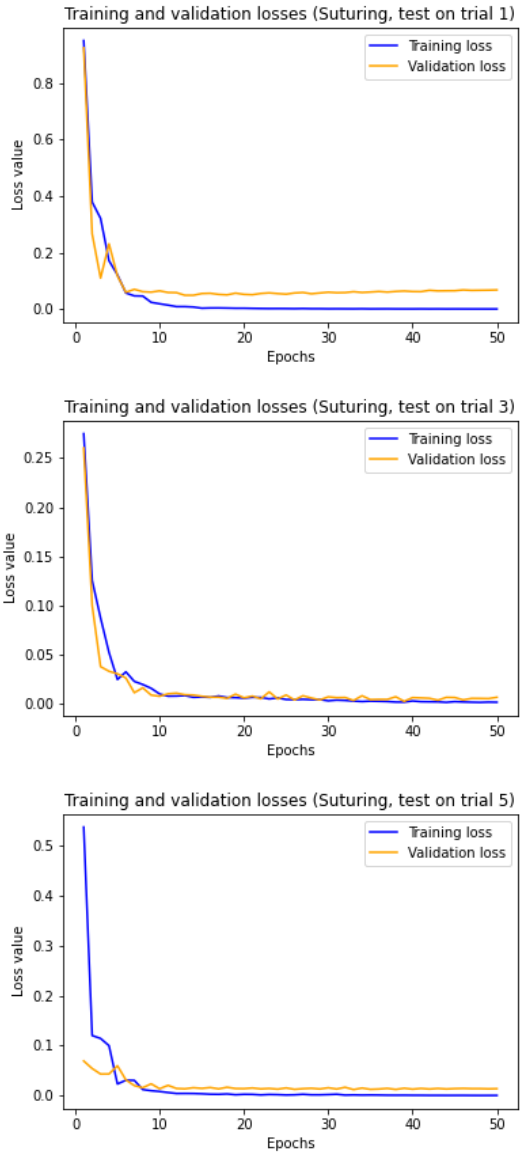

Figure 4.10: Training and validation losses curves while testing the RNN on trials 1,3 and 5 of all subjects performing the Suturing task.

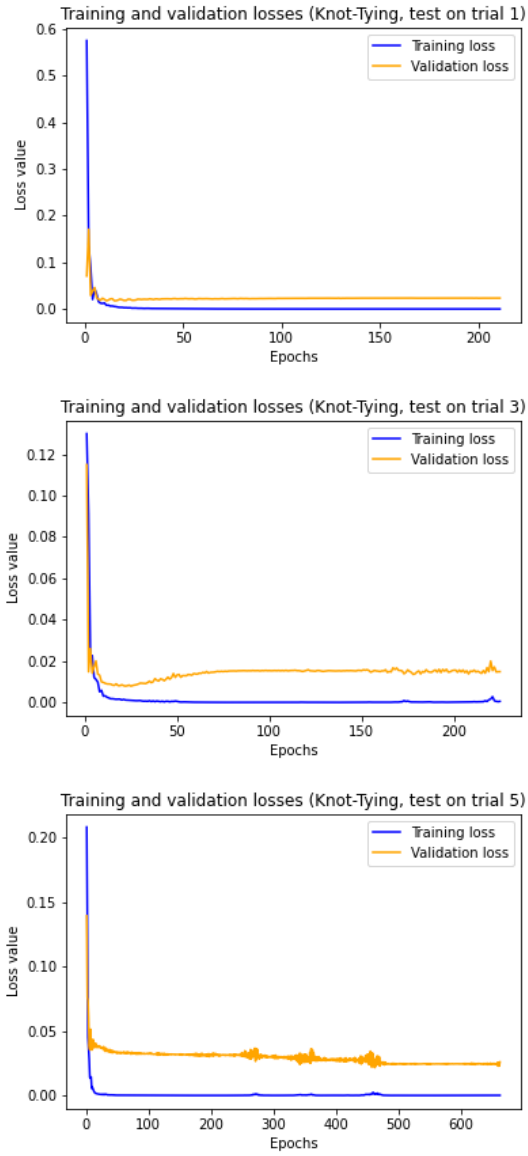

Figure 4.11: Training and validation losses curves while testing the LSTM on trials 1,3 and 5 of all subjects performing the Knot-Tying task.

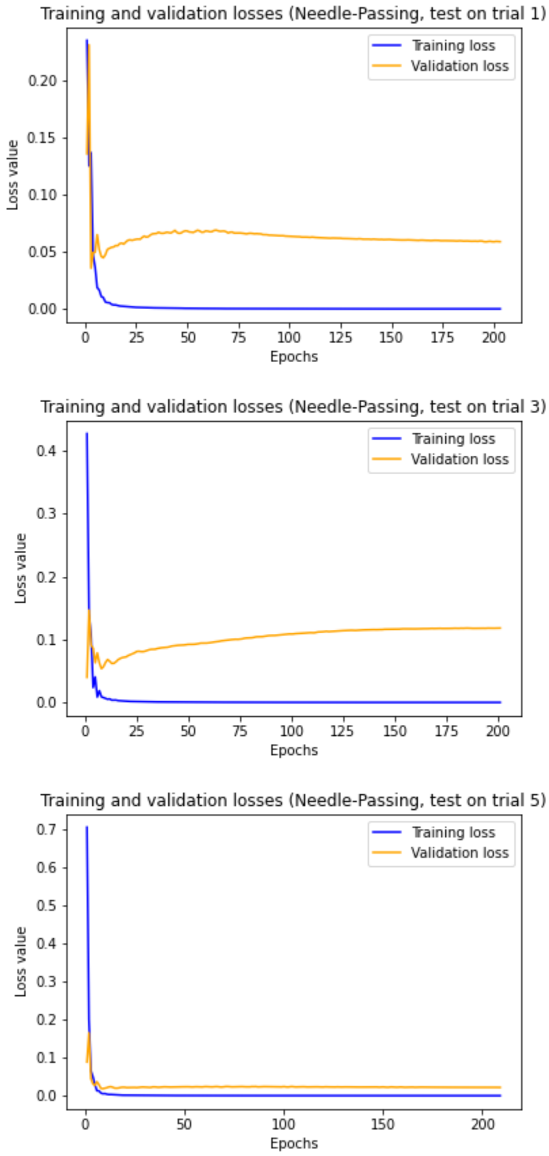

Figure 4.12: Training and validation losses curves while testing the LSTM on trials 1,3 and 5 of all subjects performing the Needle-Passing task.

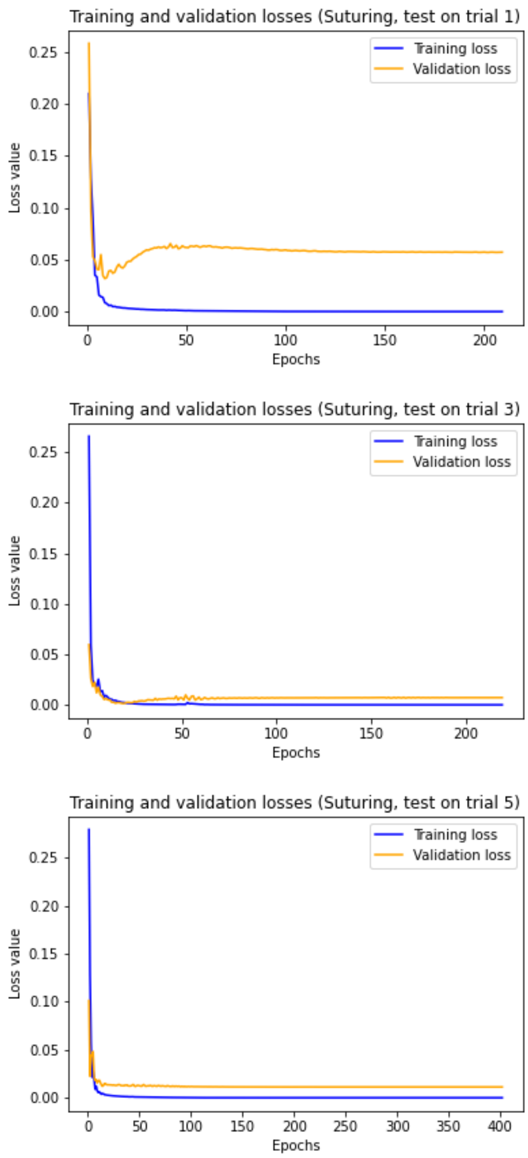

Figure 4.13: Training and validation losses curves while testing the LSTM on trials 1,3 and 5 of all subjects performing the Suturing task.

Chapter 5

Regression using an encoder-decoder network

5.1 Introduction

In this chapter, we present a deep learning encoder-decoder architecture based on a CNN and a BiLSTM to provide an automatic and objective surgical skill assessment on a continuous scale ranging from 1 to 5 based on OSATS criteria from kinematic data only. Kinematic data are first encoded to a latent space and then are transformed by the decoder to a meaningful grading system. We established three independent neural networks to evaluate surgeons on the three tasks independently. This work is an extension to our previous works presented in chapters 4 where we presented approaches that allows a global evaluation based on a single global performance score using neural networks. The reason behind the choice of this combination between the two network types is to take advantage of the characteristics of both network types by capturing both spatial and temporal information from kinematic data. Figure 5.1 depicts the proposed global approach. In addition, we propose an offline/offline and a offline/online dynamic observation of surgical skills using the CNN+BiLSTM architecture that acts as a neural observer.

5.2 Model architecture

The proposed model is composed of a CNN block followed by a BiLSTM block (see figure 5.2).

The CNN block (Encoder): The CNN component architecture is highly inspired from the CNN in [46]. It contains three convolution stages with 38, 76 and 152 filters, respectively. All filters are size (2,2) and move across the input with a stride of 1. A 2D Max pooling operation is performed after each convolution operation with a pool size of (2,2) and a stride of 2. To enhance the model generalization over the input training data and to prevent overfitting, we added Gaussian noise at the input layer and after each convolution operation. A 20% Dropout regularization is also included after each max pooling stage. In addition, we employed a batch normalization process that standardizes the inputs to a layer for each mini-batch. This normalization helps stabilizing the learning process and reducing the time required to train the network efficiently. The ReLu activation function has been used in each convolution layer. At the end of the last convolution stage, the extracted features are reshaped in a way they can fit in the next temporal component, which is the BiLSTM block.

The BiLSTM block (Decoder): The BiLSTM component contains two LSTM layers with 16 units each: the first one reads the input in a forward way regarding the time steps of the kinematic data sample, while the second one reads the inputs in a backward way. This structure allows the networks to capture both past and future information about the sequence at every time step. Next, the outputs of both LSTM layers are concatenated and transmitted to a single output node that carries the predicted OSATS score. The activation function of the bidirectional LSTM layer is a hyperbolic tangent and the activation function of the dense output node is a linear function. Batch normalization process is included in this block as well.

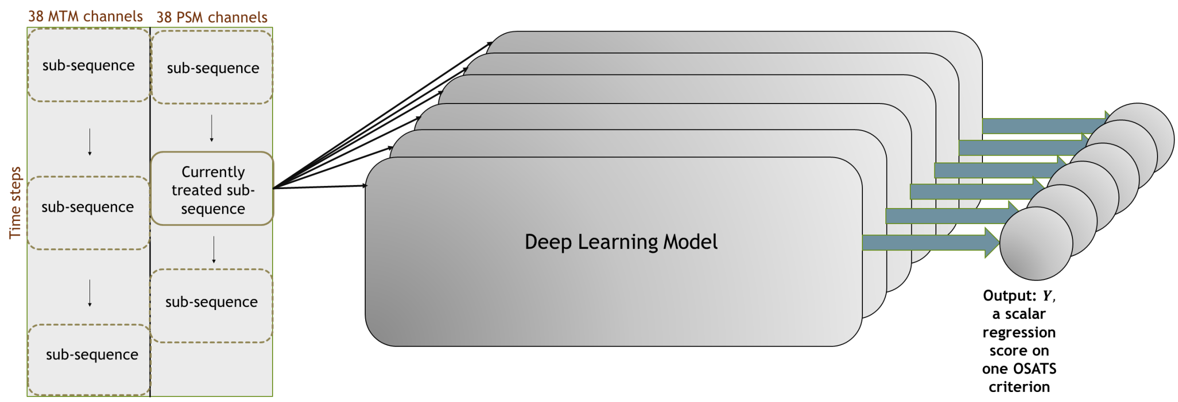

Figure 5.1: A drawing of the general adopted approach for automatic surgical skill evaluation on OSATS criteria for one surgical task: each of the 6 deep learning models is responsible for assessing the surgeon trial in one criterion

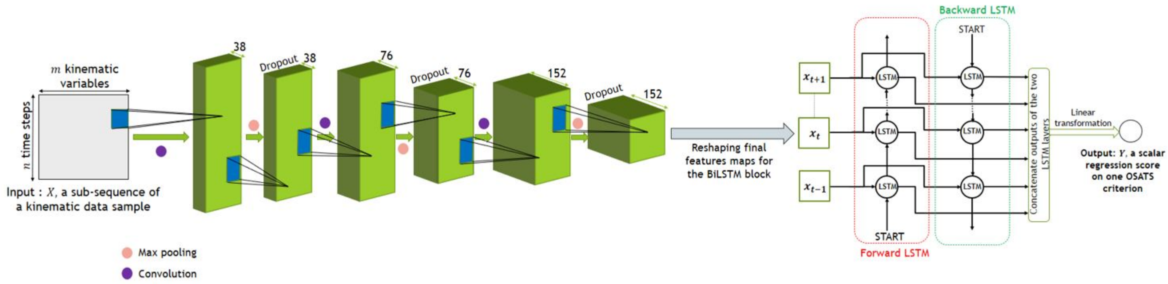

Figure 5.2: Architecture of the proposed CNN+BiLSTM network: The network takes as input a crop of a kinematic data and yield a scalar output carrying the score on one OSATS criterion (Best view with colors).

5.3 Experimental setup

5.3.1 Training and testing steps

Our model has been trained and tested with different sets of the sub-sequences extracted after executing the data augmentation process described in the chapter 3, which includes the sliding window and tools data splitting methods. To get the most robust model for each architecture, we adopted three distinct validation schemes (described in chapter 3 as well):A random training/validation/test data splitting that we will call Random Data Splitting, the Leave-One-Supertrial-Out (LOSO) cross validation scheme and a custom scheme that we named Leave-One-Grade-Out (LOGO).

We have a trained and tested individual models built according to the CNN+BiLSTM architecture, on the basis of each validation scheme, for each surgical task and for each OSATS criterion.

5.3.2 Implementation

We used Keras, a deep learning Python library to implement our network algorithm and train it from scratch. The parameters of the layers of the networks are initialized with the Glorot Uniform initialization method. We employed the optimizer Adam for all the networks with a learning rate of 0.001 while adopting the Random Data Split validation scheme. The learning rate has been decreased to 0.0001 while employing the two remaining validation schemes. The exponential decay rates of the 1st and 2nd moment estimates are set to 0.9 and 0.999, respectively. The gradient descent updates are performed with the mini batch learning method using a mini batch size of 32. To avoid overfitting, we added a L2 regularization with a value of 0.01 as well as input and hidden noise layers. The goal is to minimize the loss function which is a Mean Squared Error function between the real OSATS values and the predicted OSATS values. The size of sliding window is set to 60 with a step size of 30 when running the Random Split Data validation scheme. We increased the sliding window size to 120 while keeping the same step size of 30 when running the LOSO and the LOGO validation schemes. Theses values have been chosen after several experimentation based on trial-and-error. To obtain the best model with the lowest error possible while training each network, we included an Early Stopping callback in the training function. This callback allows to stop the training if an arbitrary number of epochs is exceeded without any learning improvement (also called as patience epochs). At the end, the weights that brought the lowest validation loss are saved and returned to build the final and the best model.

5.4 Model evaluation metrics

We compute Spearman's rank correlation coefficients to highlight correlations between the ground truth OSATS scores and the predicted OSATS scores. The Spearman rank correlation coefficient (often symbolized by ρ) is a non parametric measure of rank correlation. It assesses how well the relationship between two variables can be described using a monotonic function. Intuitively, the Spearman correlation between two distributions will be high (close to 1). when observations have a similar rank and low when having a dissimilar rank (close to -1). In addition we compute 3 statistical parameters: median, standard deviation and mean while evaluating the model on the basis of the LOGO scheme.

In addition, we computed the mean squared error (MSE) between true and predicted OSATS scores for each criterion and reported the results in Table 5.3. This metric measures the average of the squares of the errors. What this means, is that it returns the average of the sums of the square of each difference between the estimated value and the true value. The mean squared error is always 0 or positive. When a MSE is small, this is an indication that the model accurately predicts the outputs. An important piece to note is that the MSE is sensitive to outliers. This is because it calculates the average of every data point's error. Because of this, a larger error on outliers will amplify the MSE.

5.5 Results and discussion

Results are obtained through three validation strategies.

5.5.1 Random Split validation scheme

For this first validation scheme, we tested and compared several neural network architectures. All the models of each architecture have the same purpose: predicting each of the six OSATS score for a surgeon trial, independently. Hence, we created 6 independent models based on each architecture and for each surgical task. Table 5.1 shows the means of the computed Spearman coefficients (ρ) over the six OSATS scores for the three surgical tasks while adopting the Random Data Splitting validation scheme. The high correlations between the ground-truth labels and the predicted labels prove the similarity between the two distributions leading to the conclusion that all the networks are capable of giving an adequate grade deserved by tested surgeon in a 2-seconds window (We have chosen a sliding window of size 60 and a step size of 30 for this validation scheme). Specifically, CNN, LSTM, BiLSTM and the combined architectures CNN+LSTM and CNN+BiLSTM delivered the highest rank correlations. The DNN delivered lower ρ correlations because of its simple architecture and its weakness in capturing spatial and temporal features of the kinematic data. On the other hand, CNN has the ability to capture spatial features while the LSTM can capture temporal information widely spaced in time without any information loss while processing the kinematic data sample. BiLSTM benefits from its ability to read time series data in a forward direction, and in a backwards direction as well. BiLSTMs effectively increase the amount of information available to the network, improving the context available to the algorithm. Finally, the combined architectures take benefits from the LSTM, BiLSTM and CNN networks by capturing both spatial and temporal information.

	Knot-Tying	Needle-Passing	Suturing
DNN	0.82	0.76	0.76
CNN	0.98	0.97	0.97
LSTM	0.95	0.93	0.94
BiLSTM	0.95	0.93	0.95
CNN+LSTM	0.95	0.93	0.94
CNN+BiLSTM	0.95	0.93	0.94

Table 5.1: Means of Spearman's correlation coefficients calculated over the six OSATS scores between predicted and real values obtained with each network architecture (Random split validation scheme).

5.5.2 LOSO validation scheme

Due to GPU resource limitation, we tested only the CNN+BiLSTM architecture intended for this contribution. Table 5.2 shows the Spearman correlation coefficient obtained for each of the 6 models on each OSATS criterion and for each surgical task. Table 5.4 shows comparison between the ρ coefficients obtained by our model and the models of two others contributions. By adopting the same validation methodology ($LOSO$) proposed by [7] and [6], we are able to compare our proposed CNN+BiLSTM regression model to their best performing regression methods. Our regression model outperforms the FCN regression model [6] and the ApEn [7] regression method in the three surgical tasks. In other words, the prediction and the ground truth OSATS scores are more correlated when using CNN+BiLSTM than the ApEn-based and FCN-based solutions on the three surgical tasks. We point out the differences of our proposed method with [6]: The authors of this contribution did not apply any data augmentation strategy and they have trained their Fully Convolutional Network with raw kinematic data, while we have trained our models with crops of kinematic data obtained after applying the data augmentation method described in Sub-section 3.4. Also, they used one-dimensional convolution layers while we used two-dimensional convolution layers in our CNN component. the latter difference seems to explain why we have got better results: applying 2D filters allows the CNN to capture temporal information about the input sequence along its temporal axis but also to capture spatial information along the kinematic variables axis, which enables the CNN to choose the most significant features and capture correlation features between physical quantities (positions, velocities and accelerations). In addition, the BiLSTM block of our model allows the model to have a temporal understanding of the extracted features and to process them in two opposite directions. Finally, the authors designed a single FCN model to predict a vector of the six OSATS scores while we desgined six independent CNN+BiLSTM models for each OSATS score. Concerning the work [7], the authors adopted the approximate entropy (ApEn) algorithm to extract features from each surgical trial which are later fed to a nearest neighbor classifier. According to the authors, the features they use try to differentiate between different skill levels using data repeatability. To justify the low performance on the Needle Passing task, they have claimed that the Needle Passing task is a less repetitive task as compared to the Knot-Tying and the Suturing tasks. According to us, their method that consists in evaluating holistic features for predicting skill level is outdated since the arrival of powerful deep learning libraries like Keras [85], that allows data scientists to easily build, train and test neural networks, knowing that neural networks are the most performing methods in the machine learning and deep learning fields.

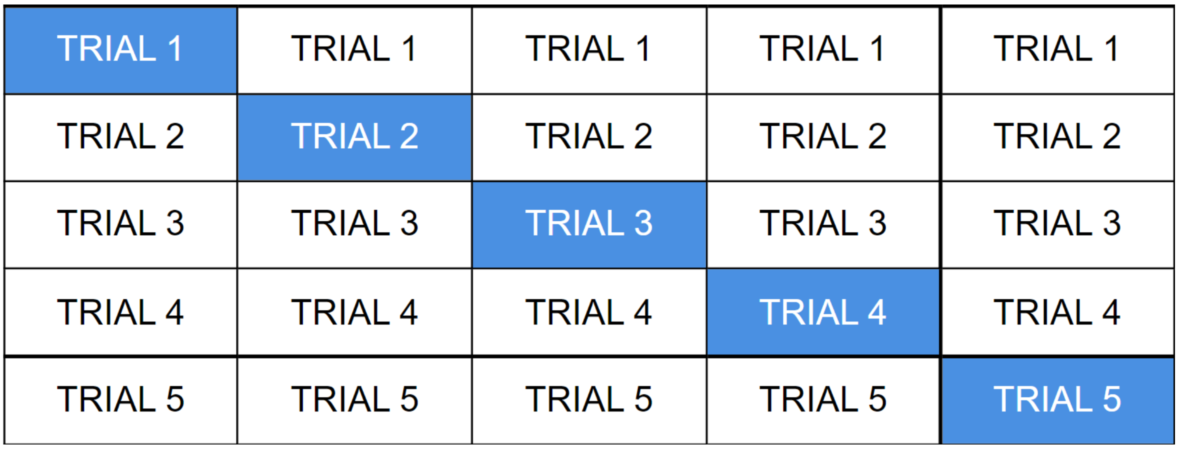

Figure 5.3: A Diagram representing the LOSO validation scheme process.

	Knot-Tying	Needle-Passing	Suturing
Respect for tissue (RFT)	0.83	0.49	0.46
Suture/Needle handling (SNH)	0.82	0.79	0.75
Time and motion (TM)	0.87	0.85	0.68
Flow of operation (FO)	0.76	0.58	0.62
Overall performance (OP)	0.89	0.58	0.71
Quality of final product (QFP)	0.75	0.31	0.67

Table 5.2: Individual Spearman's correlation coefficients computed for each of the six OSATS scores between predicted and real values obtained with the CNN+BiLSTM architecture (LOSO validation scheme).

5.5.3 LOGO validation scheme

The challenge intended by this validation methodology is to check the performance of our CNN+BiLSTM model on predicting the regression outputs of unseen input kinematic data whose true outputs have been unseen,too, during the training step. This can help to significantly increase the level of confidence we will attribute to our network. Tables 5.5, 5.6 and 5.7 refer to the results of 3 statistical parameters computed for the predicted outputs by our proposed CNN+BiLSTM model for each OSATS criterion, for the Knot-Tying task, the Needle Passing task and the Suturing task, respectively. We can not compute the Spearman correlation coefficient in this case because the real outputs vector has constant values (knowing that we leave a unique grade for testing). Hence, there is no variation in the real outputs vector so its standard deviation is equal to 0 which will result in zero division in the Spearman function, thereby being undefined. Instead, we computed the median, the standard deviation

	Knot-Tying	Needle-Passing	Suturing
Respect for tissue (RFT)	0.22	0.72	0.42
Suture/Needle handling (SNH)	0.23	0.23	0.52
Time and motion (TM)	0.11	0.21	0.52
Flow of operation (FO)	0.31	0.23	0.42
Overall performance (OP)	0.23	0.47	0.43
Quality of final product (QFP)	0.43	0.93	0.35
Mean over the six criteria	0.25	0.47	0.44

Table 5.3: Mean Squared Error between predicted and actual OSATS scores obtained using the CNN+BiLSTM architecture (LOSO validation scheme)

Compared methods	Knot-Tying	Needle-Passing	Suturing
ApEn [7] (*LOSO*)	0.66	0.45	0.59
FCN [6] (*LOSO*)	0.65	0.57	0.60
CNN+BiLSTM (proposed) (*LOSO*)	**0.82**	**0.60**	**0.65**

Table 5.4: Comparative results obtained by our model and the models of two other contributions (Means of the Spearman coefficient over the six OSATS scores)

and the mean of the predicted outputs for each OSATS criterion. We compared the mean of predicted outputs to the real outputs for each OSATS criterion. We consider that the result is quite good when the difference between ground truth scores and predicted scores is not greater than 1. We can notice that all the trained models for each criterion have difficulties in predicting a score that was unseen during the training phase. One can notice that the results are good on the "Time and Motion" (TM) criterion in the Knot-Tying and in the Suturing tasks (table 5.6, table 5.7). The reason behind this might be the fact that this criterion is more correlated with the kinematic data since the latter is a description of the surgical robot tools motions through time steps. However, a criterion like "Respect for Tissue" (RFT) seems to be more related to forces and dynamics, which are not described in the kinematic data.

Statistical parameters	RFT	SNH	TM	FO	OP	QFP
Median	3.07	2.89	2.08	3.07	2.91	1.88
Standard deviation	0.10	0.36	0.50	0.12	0.15	0.20
Mean of the predicted outputs	3.05	2.73	**1.98**	**3.06**	2.91	**1.90**
Real outputs	4.00	4.00	**2.00**	**4.00**	4.00	**1.00**

Table 5.5: Regression results with the CNN+BiLSTM architecture for the Knot-Tying task while adopting the LOGO validation scheme.

Statistical parameters	RFT	SNH	TM	FO	OP	QFP
Median	2.47	2.99	2.30	2.81	2.10	2.70
Standard deviation	0.32	0.29	0.76	0.16	0.19	0.61
Mean of the predicted outputs	2.45	2.84	2.66	2.80	2.16	2.78
Real outputs	4.00	4.00	3.00	2.00	3.00	1.00

Table 5.6: Regression results with the CNN+BiLSTM architecture for the Needle Passing task while adopting the LOGO validation scheme.

Statistical parameters	RFT	SNH	TM	FO	OP	QFP
Median	2.15	2.04	4.127	4.02	4.15	4.08
Standard deviation	0.25	0.06	0.08	0.13	0.07	0.06
Mean of the predicted outputs	2.24	2.05	**4.12**	3.99	**4.14**	**4.07**
Real outputs	1.00	1.00	**5.00**	5.00	**5.00**	**5.00**

Table 5.7: Regression results with the CNN+BiLSTM architecture for the Suturing task while adopting the LOGO validation scheme.

5.6 Regression versus classification

In this section, we present results obtained through a classification method using OSATS scores. It consists in considering OSATS outputs as categorical instead of considering it as a continuous scale. The goal of this experiment is to determine which approach leads to the best results and thus, gives the best and most meaningful feedback to the trainee. To compare this approach to the proposed regression approach appropriately, we used the same CNN+BiLSTM architecture with the same hyperparameters described previsouly. The difference lies on the following: For this classification approach, we used the categorical Crossentropy loss function instead of the mean squared error loss function. In addition, desired outputs have been one-hot encoded and we used a Softmax activation function in the last layer of the network architecture to compute the predicted output on a one-hot encoded 6 elements array:

- Score 0 becomes $[1, 0, 0, 0, 0, 0]$
- Score 1 becomes $[0, 1, 0, 0, 0, 0]$
- Score 2 becomes $[0, 0, 1, 0, 0, 0]$
- Score 3 becomes $[0, 0, 0, 1, 0, 0]$
- Score 4 becomes $[0, 0, 0, 0, 1, 0]$
- Score 5 becomes $[0, 0, 0, 0, 0, 1]$

To evaluate performance of a network at predicting categorical outputs, metrics as accuracy, precision or F1-score are usually used. In our case, we are interested in the comparison with the regression approach. Thus, we computed the Spearman rank correlation coefficient between predicted and target outputs. To do so, we performed the reverse of one-hot encoding for each predicted outputs and went back to original score by considering the index of the maximum value in the one-hot encoded array (For example, if a predicted array is $[0.012, 0.014, 0.861, 0.001, 0.002, 0.110]$, the maximum value has the index 2 and the output would be then classified as 2. After that, we are able to compute Spearman (ρ) correlation coefficients between predicted and target outputs for the classification method. To provide a comparison on the same basis than the regression approach, we used the LOSO validation scheme.

Results are reported in table 5.8: For each surgical task, the left column represents the computed Spearman coefficients for each OSATS criterion of the regression approach, while the right one represents the computed ρ coefficients of the classification approach. Clearly, ρ coefficients obtained through the proposed regression approach are superior to ρ coefficients obtained through the classification approach on the three surgical tasks. This means that predicted and target outputs are more correlated by adopting the regression approach than by adopting the classification approach. We can conclude that the proposed regression method provides a more meaningful and meticulous feedback to surgical trainees.

5.7 Training parameters study

In this section, we provide an experimental study involving a comparison between regression results obtained using two neural network architectures and by affecting different values to

	Knot-Tying		Needle-Passing		Suturing	
Respect for tissue (RFT)	0.83	0.66	0.49	0.46	0.46	0.38
Suture/Needle handling (SNH)	0.82	0.62	0.79	0.67	0.75	0.54
Time and motion (TM)	0.87	0.79	0.85	0.71	0.68	0.54
Flow of operation (FO)	0.76	0.63	0.58	0.59	0.62	0.54
Overall performance (OP)	0.89	0.71	0.58	0.46	0.71	0.57
Quality of final product (QFP)	0.75	0.50	0.31	0.28	0.67	0.56

Table 5.8: Comparative table showing correlation results obtained for both classification and regression methods. For each task, the left column shows the Spearman correlation results obtained with the proposed regression method while the right one shows the Spearman correlation results obtained with the classification method. (CNN+BiLSTM and LOSO validation scheme)

training batch size and dropout in the convolutional layers. The first architecture is the proposed CNN+BiLSTM and the second one is a variation of the latter where we put a GRU recurrent network instead of a LSTM (CNN+BiGRU) and we kept same number of units (16) and same hyperparameters described previously. The structural difference between these two recurrent networks lies on the number of gates: LSTM has three gates named *input, output* and *forget* gates, whereas GRU has only two gates named *reset* and *update* gates. Moreover, GRUs and LSTMs deal differently with the vanishing gradient problem which is encountered with the classic RNN. Here are the main points comparing the two:

- The GRU unit controls the flow of information like the LSTM unit, but without having to use a memory unit. It just exposes the full hidden content without any control.
- GRUs are computationally more efficient since they have a less complex structure. Thus, they train faster but are as effective as LSTMS in making predictions on sequence data.

For a detailed description, this paper [86] explains this efficiently.

We trained and tested models on the basis of the two architectures following the same data augmentation method described in chapter 3 for all surgical tasks and all OSATS criteria. Learning rate, weights initialization, L2 regularization values and other implementation details described in subsection 5.2 stayed the same as well. LOSO validation scheme has been adopted while testing every model. Spearman rank correlation coefficient (ρ) has been used to compute correlation between predicted and target outputs. Regression results are reported in tables 5.9 and 5.10 for the CNN+BiLSTM and the CNN+BiGRU, respectively. For the batch size, we have chosen to double it twice starting from 32. Dropout values have been set to 25%, 50% and 75% as these values are common in the deep learning community. Each batch size is tested alongside a single dropout value.

As it can be seen, both architectures have shown good regression correlation results for the three tasks and the six OSATS criteria while affecting 32 and 25% values for the batch size and the dropout regularization, respectively. Increasing dropout to 50% and and the batch size to 64 led to a significant decrease in the quality of scores prediction for the CNN+BiLSTM architecture while the CNN+BiGRU suffered only from a slight decrease in performance after increasing the values of the two training parameters. Finally, increasing batch size and dropout to 128 and 75% respectively led to a near decorrelation between predicted and true outputs concerning the CNN+BiLSTM while the CNN+BiGRU behaves better but nevertheless showed a huge drop in prediction quality.

We conclude that training both architectures with the smallest batch size and the lowest percentage of dropout led to the best OSATS scores prediction quality. To support this study, we cite this work [87] where *Ibrahem Kandel et al.* conducted a study about the effect of batch size on the performance of CNNs in medical image classification using a VGG16 network and training it with different batch sizes. Their results have shown that the network that has been trained with the lowest batch size provided the best accuracy and thus the best generalization. On the other hand, dropout regularization has been introduced in the deep learning field to prevent overfitting and improve robustness and generalization of the neural network on validation and test data. But, after a certain threshold percentage, the network may underfit. This may or may not happen depending on the network architecture complexity and the training dataset size and nature.

	Knot-Tying			Needle-Passing			Suturing		
Batch size	32	64	128	32	64	128	32	64	128
Dropout	25%	50%	75%	25%	50%	75%	25%	50%	75%
Respect for tissue	0.76	0.49	0.01	0.50	0.08	0.12	0.50	0.33	0.17
Suture/Needle handling	0.77	0.47	0.04	0.81	0.58	0.24	0.74	0.74	0.01
Time and motion	0.85	0.54	0.21	0.83	0.67	0.68	0.69	0.67	0.11
Flow of operation	0.76	0.42	0.13	0.55	0.22	0.01	0.64	0.58	0.40
Overall performance	0.89	0.72	0.10	0.56	0.40	0.23	0.79	0.58	0.36
Quality of final product	0.79	0.06	0.07	0.31	0.37	0.16	0.80	0.67	0.27
Mean on all criteria	0.80	0.45	0.02	0.59	0.38	0.24	0.69	0.59	0.22

Table 5.9: Correlation results obtained using the CNN+BiLSTM architecture by varying training batch sizes and dropout values.

	Knot-Tying			Needle-Passing			Suturing		
Batch size	32	64	128	32	64	128	32	64	128
Dropout	25%	50%	75%	25%	50%	75%	25%	50%	75%
Respect for tissue	0.82	0.66	0.15	0.38	0.43	0.19	0.54	0.52	0.36
Suture/needle handling	0.84	0.67	0.23	0.80	0.78	0.44	0.77	0.77	0.58
Time and motion	0.85	0.76	0.33	0.86	0.85	0.44	0.70	0.49	0.42
Flow of operation	0.78	0.67	0.65	0.63	0.64	0.62	0.65	0.66	0.67
Overall performance	0.88	0.78	0.44	0.60	0.60	0.22	0.74	0.66	0.29
Quality of final product	0.62	0.64	0.12	0.23	0.25	0.05	0.71	0.68	0.55
Mean on all criteria	0.80	0.70	0.32	0.58	0.60	0.33	0.69	0.63	0.48

Table 5.10: Correlation results obtained using the CNN+BiGRU architecture by varying training batch sizes and dropout values.

5.8 Dynamic observation of surgical skills using OSATS

Dynamic observation of surgical skills is an approach that aims to provide instant evaluation of surgical skills in small temporal windows. By analyzing a surgeon's movements in real-time, this strategy can provide objective measures of surgical proficiency, which can help surgeons identify areas of improvement and optimize their techniques. In this section, we provide results obtained through a dynamic observation of surgical skills with both offline and online training approaches of a CNN+BiLSTM model.

5.8.1 Offline/Offline surgical skills reconstruction

This first approach consists in training a CNN+BiLSTM model with the sub sequences of kinematic data obtained with the data augmentation pipeline described previously. The output $y(k)$ of the model contains only the value of the OSATS score of a given criterion. The best model is saved and then is applied to new, unseen sub sequences extracted from a test sample in order to make soft score predictions on 4-seconds windows every 1 time step. This approach does not involve any fine-tuning of the neural observer and predictions are made directly on the performance score.

Figure 5.5 shows a real-time assessment of surgical skills using the CNN+BiLSTM described in this chapter. The assessment is based on the respect for tissue OSATS criterion on the three JIGSAWS tasks. At every frame, the performance is graded on a soft score scale between 1 and 5 as described previously by analyzing a temporal window of 120 frames (4 seconds) extracted from the whole sequence. MTM and PSM movements are evaluated separately according to the proposed data augmentation pipeline. Performance is compared to the ground truth output which is constant all over the trial, since it has been attributed to the

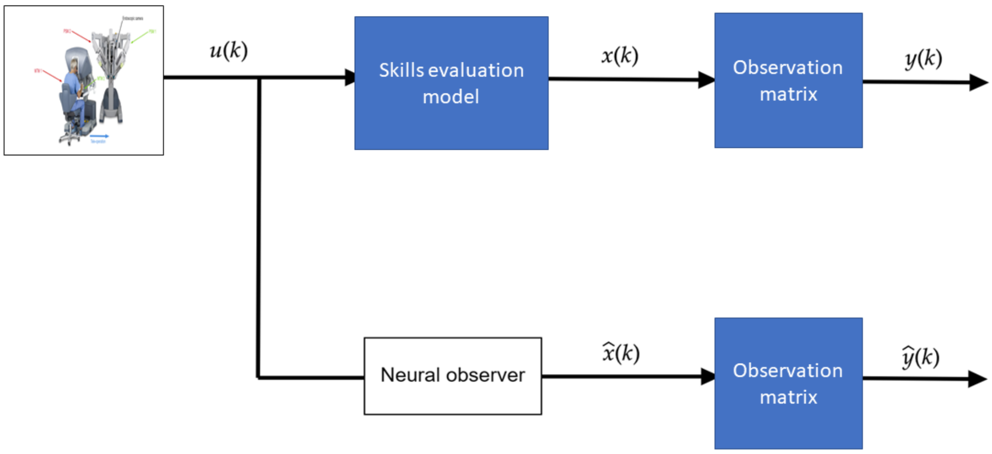

Figure 5.4: Neural observer scheme for dynamic observation of surgical skills following the Offline/Offline approach.

trainee only after finishing the task. We notice some correlation between scores attributed by the CNN+BiLSTM model to the MTM and PSM movements, especially in the suturing task. This means that the performance using the MTMs is sometimes highly reflected in the movements of the PSMs and vice-versa. Tables 5.11 and 5.12 refer to the mean squared error for the offline/offline approach and the offline/online approach, respectively.

5.8.2 Offline/Online surgical skills reconstruction

This second approach consists in adding a closed loop with a comparator (see figure 5.6) in order to compute the estimation error $e(k)$ and update weights of the neural observer through the optimizer. In this case, the output $y(k)$ contains only the kinematic variables at the last time step of $u(k)$ and the performance score is removed. This approach allows a fine-tuning of the neural observer based only on the states $x(k)$ of the system. However, the final predictions of the neural observer are made on the kinematic variables states and on the performance score.

Figure 5.7 shows online observation of surgical skills based on the respect for tissue criterion of subject H performing the three tasks. Because of the additional fine-tuning step of the neural observer, dynamic score predictions of both Master tool Manipulator and Patient Side Manipulator are more correlated with the desired outputs in the offline/online approach than

	Knot-Tying	Needle-Passing	Suturing
Master Tool Manipulator	1.23	0.55	3.16
Patient Side Manipulator	0.66	0.49	4.44

Table 5.11: Per-window mean squared error between predicted and true outputs over the entire trial 5 of subject H performing the three tasks (Offline/Offline approach).

	Knot-Tying	Needle-Passing	Suturing
Master Tool Manipulator	7.66	0.83	2.10
Patient Side Manipulator	6.46	0.57	2.25

Table 5.12: Per-window mean squared error between predicted and true outputs over the entire trial 5 of subject H performing the three tasks (Offline/Online approach).

in the Offline/Offline approach.

These approaches allow a real-time reconstruction of surgical skills: In the context of surgery training, this means that a trainee is able to get instant feedback of her/his performance and thus, is able to correct trajectories of the robot tools in real-time, independently of the supervision of an expert surgeon. Additionally, one or more segments of the trajectory where the trainee needs to improve her/his movements are targeted and highlighted and thus, she/he is able to provide a better performance in the next operation. In the context of real surgery, this technique has the potential to improve patient outcomes. Surgical errors and complications are a leading cause of morbidity and mortality in healthcare, and many of these errors can be attributed to human factors, such as lack of experience or fatigue. By providing real-time feedback on surgical proficiency, dynamic evaluation using a neural network can help surgeons identify and correct errors before they lead to complications, potentially reducing the risk of adverse events and improving patient outcomes.

Finally, dynamic evaluation of surgical skills has the potential to improve the efficiency of surgical training. Traditionally, surgical training involves a long period of apprenticeship, during which trainees observe and assist experienced surgeons. While this approach can be effective, it is time-consuming and can be limited by the availability of experienced mentors. By providing instant feedback on surgical skills, dynamic evaluation using a neural network or another machine learning method can help trainees optimize their techniques and progress more quickly through their training, potentially reducing the overall time required for surgical training.

5.9 Conclusion

In this chapter, we presented the main contribution of this thesis: A novel deep learning method for automatic and objective surgical skill assessment from kinematic data only. Our proposed method has addressed a fundamental problem in the field by considering surgical skills assessment as a regression problem on a soft continuous scale instead of a classification problem using predefined categories. In addition, we proposed a novel validation scheme for deep learning models, LOGO, which is a challenging one but feasible as our results with the CNN+BiLSTM have shown. On the OSATS criteria, our proposed model, which is based on a CNN and a BiLSTM combination, produced brand-new, cutting-edge predictions. Finally, we proposed a new way to assess surgeons in real-time: the dynamic observation of surgical skills. This approach has not been yet adopted in the literature.

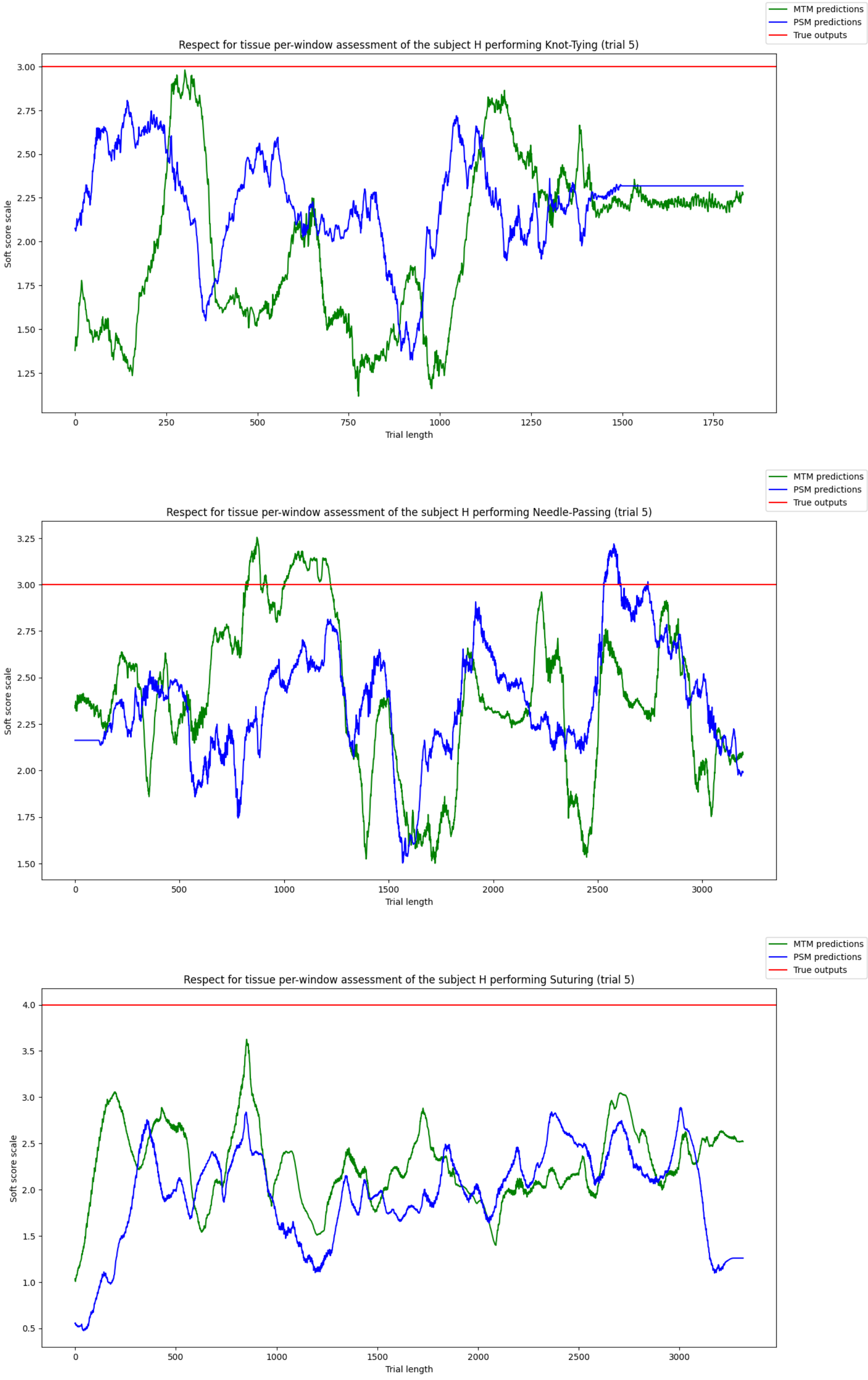

Figure 5.5: Per-window evaluation of surgical skills on Knot-Tying, Needle-Passing and Suturing, showing a dynamic evaluation and useful feedback to the trainee. Subject H is assessed on the respect for tissue criterion from OSATS (Offline/Offline approach).

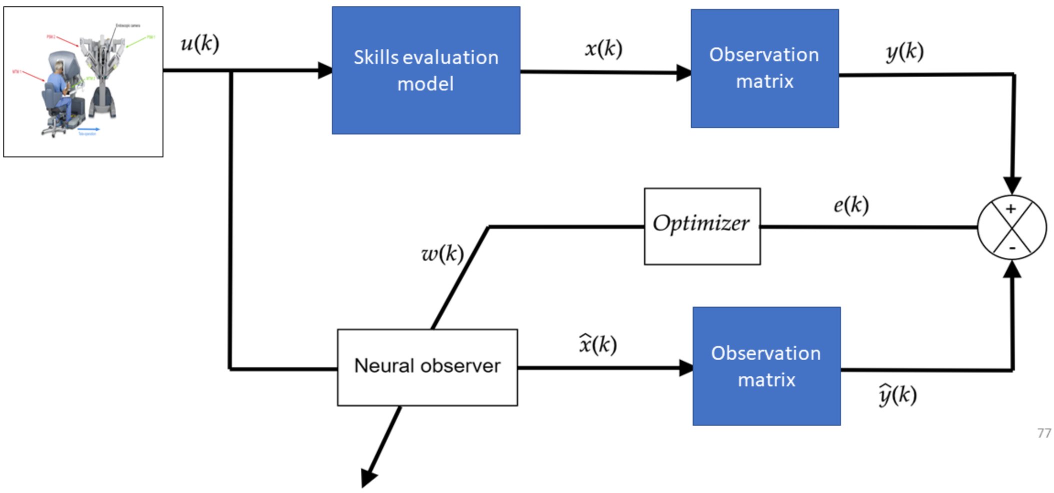

Figure 5.6: Neural observer scheme for dynamic observation of surgical skills following the Offline/Online approach.

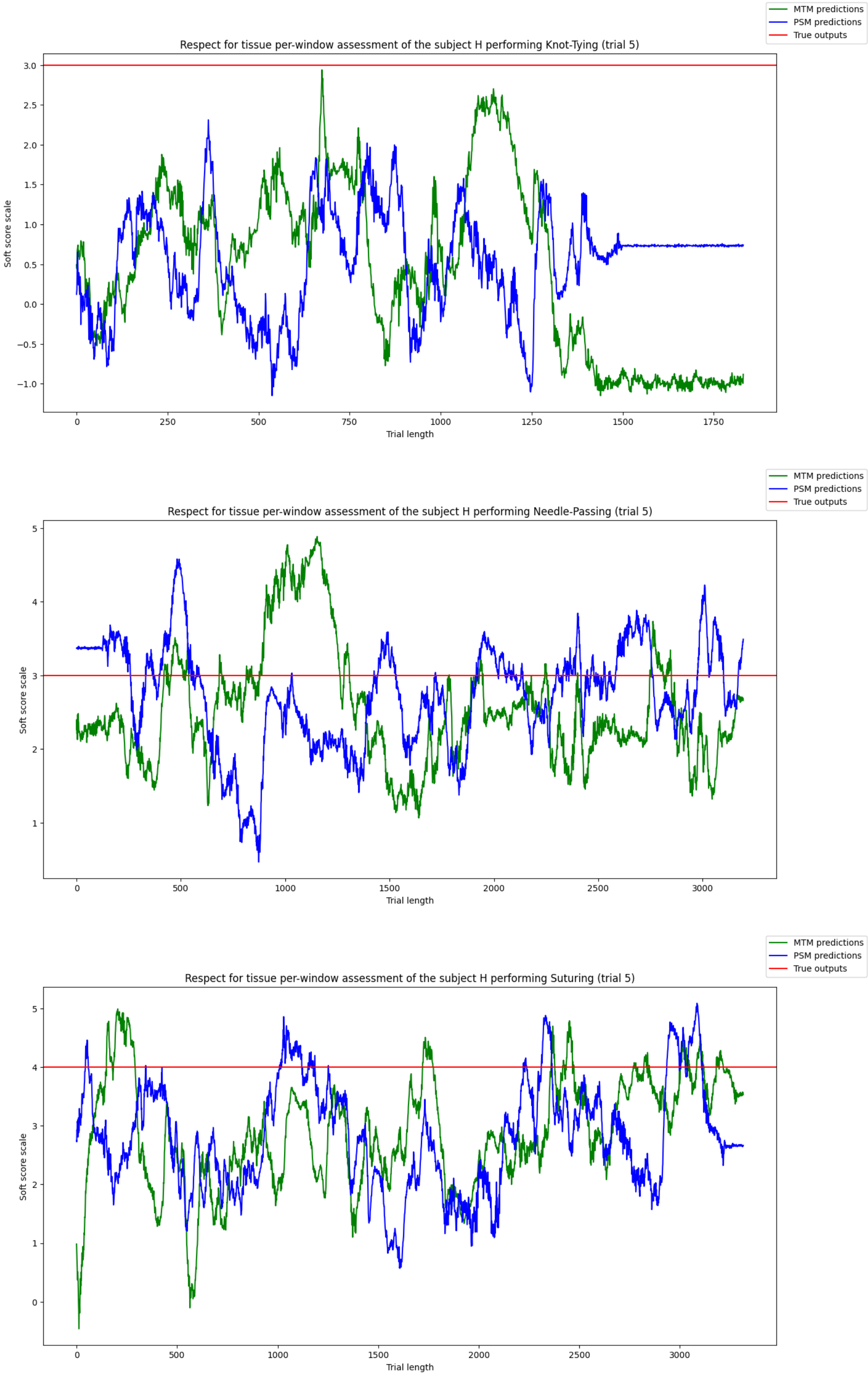

Figure 5.7: Per-window evaluation of surgical skills on Suturing showing a dynamic evaluation and useful feedback to the trainee. Subject H is assessed on the respect for tissue criterion from OSATS (Offline/Online approach).

General conclusion

In this thesis, we have investigated the potential of deep learning for surgical skills assessment. The field of surgical skills assessment has seen significant advances in recent years, with the development of various tools and techniques for evaluating the performance of surgeons. However, many of these methods are subjective, time-consuming, and can be prone to error.

We have shown that deep learning can be used to develop an objective, reliable, and automated system for surgical skills assessment. By using deep feedforward neural netwroks, convolutional neural networks and recurrent neural networks, we have demonstrated the ability of deep learning algorithms to process and analyze large amounts of surgical kinematic data, enabling the accurate evaluation of surgical skills. Since most of the recent contributions in the field of surgical skills assessment proposed classification-based methods, we addressed a fundamental problem by proposing methods that rely only on regression on float scores based on various surgical criteria.

One of the key advantages of deep learning for surgical skills assessment is its ability to capture complex patterns and relationships in the data, making it well-suited for the analysis of surgical performance. Additionally, deep learning algorithms can learn from vast amounts of data, allowing for the creation of highly accurate models for surgical skills assessment.

We have also discussed the challenges that must be overcome in the development of deep learning systems for surgical skills assessment. These include issues such as data availability and quality and the need for robust validation and testing.

Despite these challenges, the results of our research demonstrate the tremendous potential of deep learning for surgical skills assessment. The development of accurate and reliable surgical skills assessment systems is essential for the improvement of patient outcomes and the advancement of the field of surgery.

In conclusion, this thesis highlights the significance of deep learning for surgical skills assessment and provides a foundation for further research in this area. By exploring the potential of deep learning for surgical skills assessment, we have demonstrated the ability of these algorithms to provide objective and reliable evaluations of surgical performance, which is essential for the improvement of patient outcomes and the advancement of the field of surgery.

For future work, we intend to find other ways to employ the LOGO validation scheme in other surgical databases. We believe that it is a solid validation strategy that allows more reliability to the trained models than other commonly used validation strategies. Also, it would be interesting to explore other surgical or non-surgical databases involving kinematic sequences in order to check the general applicability of the proposed CNN+BiLSTM architecture. Finally, we aim to expand our model to a real-time surgical skill assessor because it is able to provide feedback by taking small crops of kinematic data using the proposed approach of dynamic evaluation of surgical skills in chapter 5.

Appendices

Appendix

Remaining training/validation curves

We provide here the remaining training/validation plots from chapter 4 for the remaining trials 2 and 4 for each subject and for each surgical task using DNN, RNN and LSTM. The adopted validation strategy is LOSO.

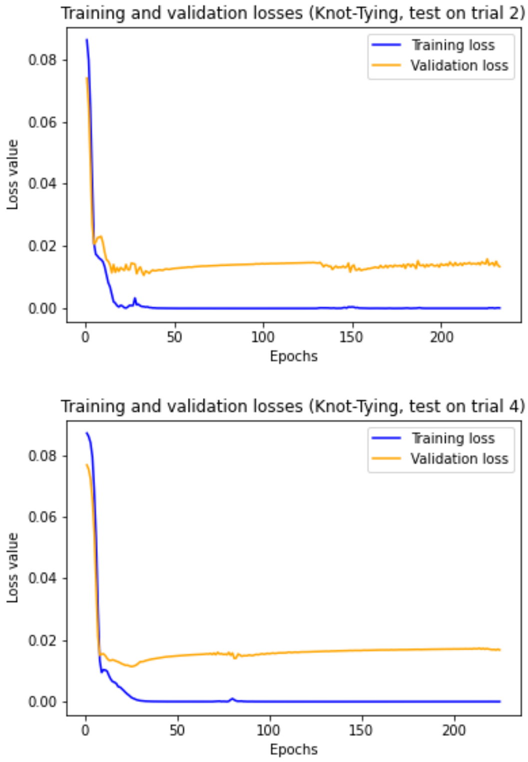

Figure 8: Training and validation losses curves while training and testing the DNN on trials 2 and 4 of all subjects performing the Knot-Tying task.

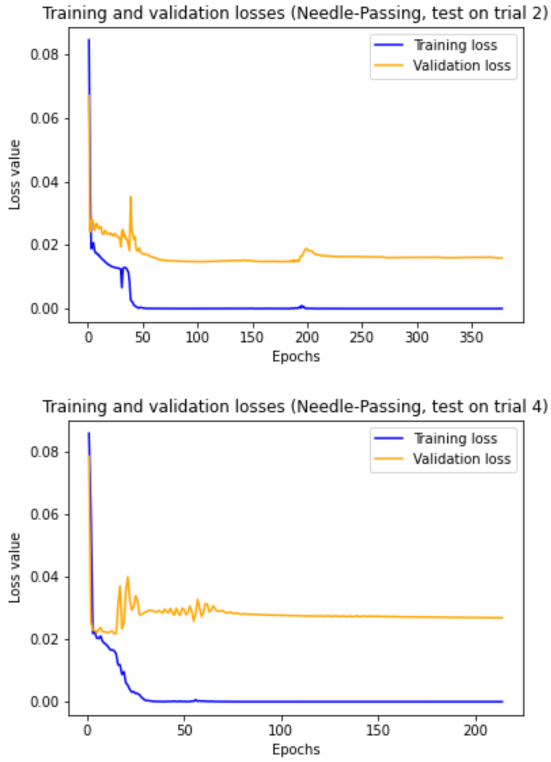

Figure 9: Training and validation losses curves while training and testing the DNN on trials 2 and 4 of all subjects performing the Needle-Passing task.

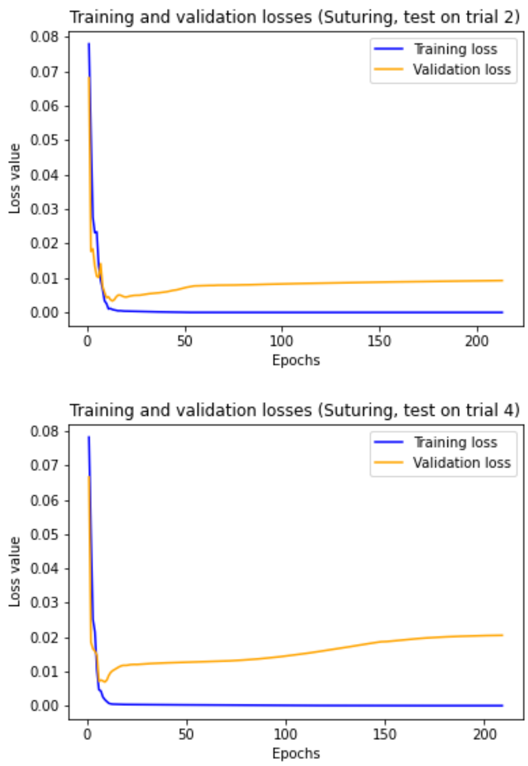

Figure 10: Training and validation losses curves while training and testing the DNN on trials 2 and 4 of all subjects performing the Suturing task.

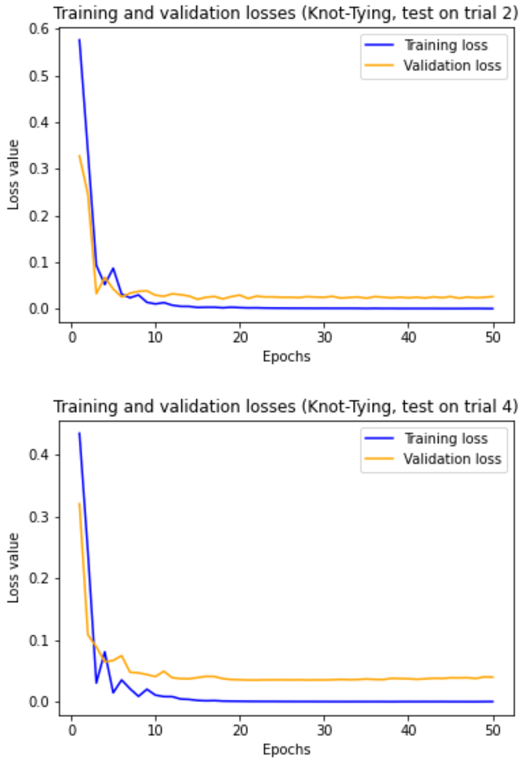

Figure 11: Training and validation losses curves while training and testing the RNN on trials 2 and 4 of all subjects performing the Knot-Tying task.

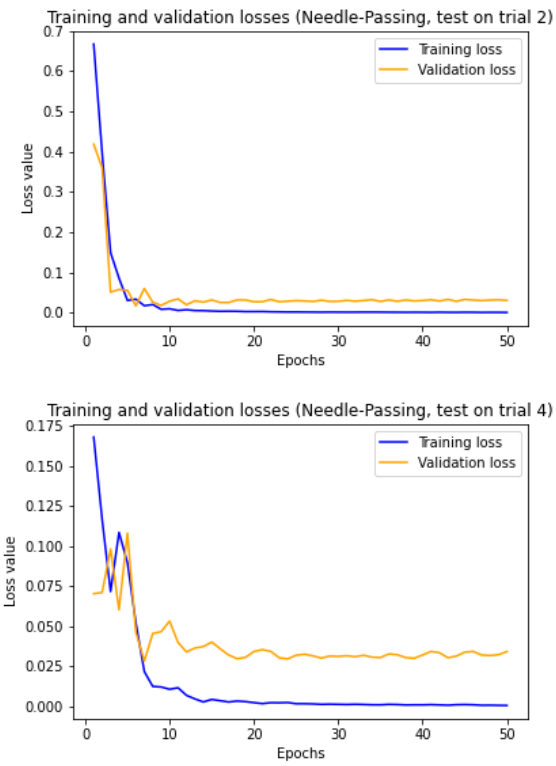

Figure 12: Training and validation losses curves while training and testing the RNN on trials 2 and 4 of all subjects performing the Needle-Passing task.

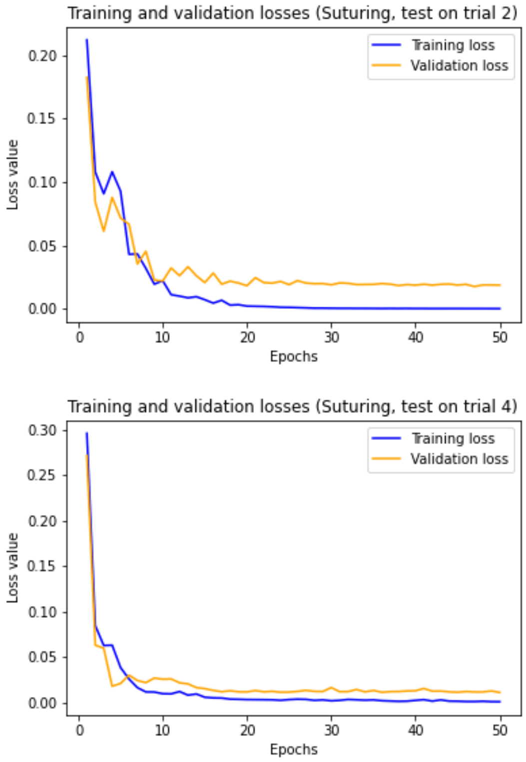

Figure 13: Training and validation losses curves while training and testing the RNN on trials 2 and 4 of all subjects performing the Suturing task.

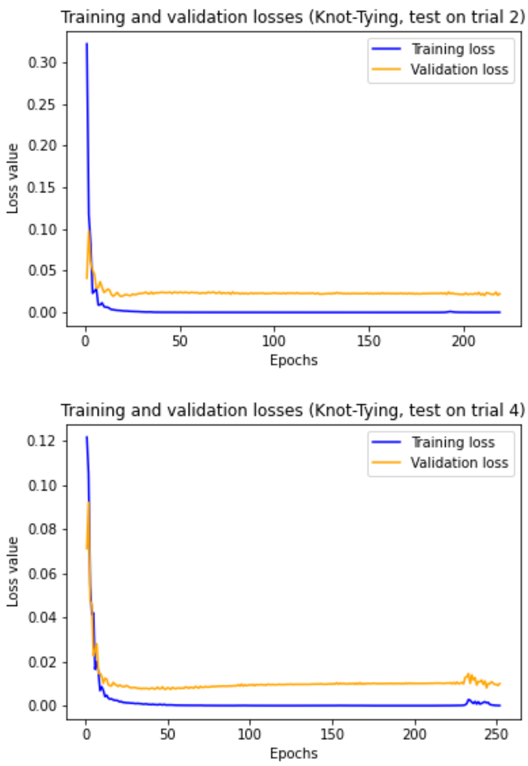

Figure 14: Training and validation losses curves while training and testing the LSTM on trials 2 and 4 of all subjects performing the Knot-Tying task.

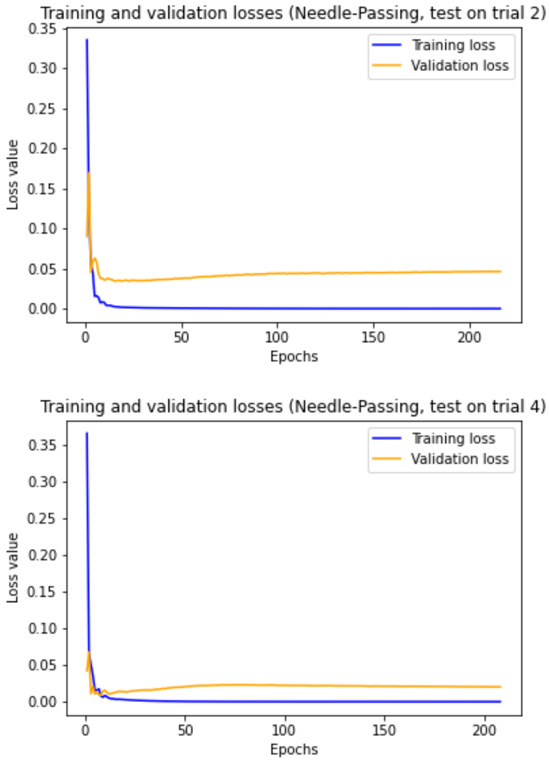

Figure 15: Training and validation losses curves while training and testing the LSTM on trials 2 and 4 of all subjects performing the Needle-Passing task.

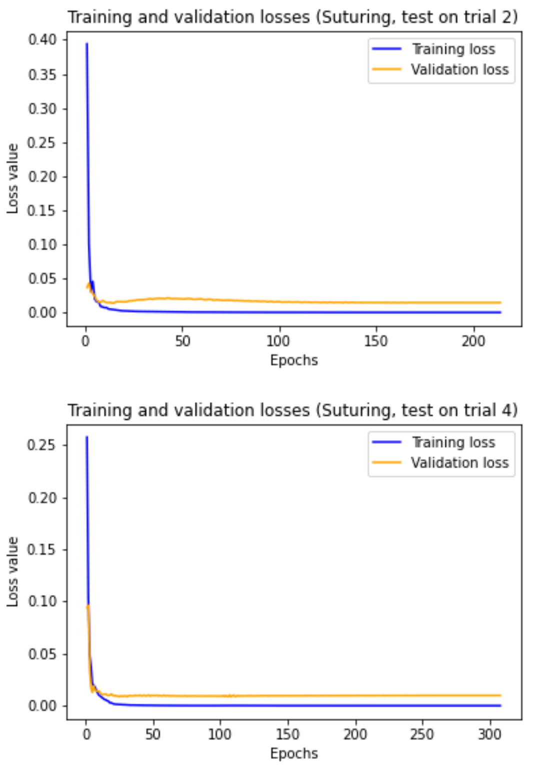

Figure 16: Training and validation losses curves while training and testing the LSTM on trials 2 and 4 of all subjects performing the Suturing task.

Bibliography

[1] Alma Y Alanis and Edgar N Sanchez. *Discrete-Time Neural Observers: Analysis and Applications*. Academic Press, 2017.

[2] Erim Yanik, Xavier Intes, Uwe Kruger, Pingkun Yan, David Diller, Brian Van Voorst, Basiel Makled, Jack Norfleet, and Suvranu De. Deep neural networks for the assessment of surgical skills: A systematic review. *The Journal of Defense Modeling and Simulation*, 19(2):159–171, 2022.

[3] Ziheng Wang and Ann Majewicz Fey. Deep learning with convolutional neural network for objective skill evaluation in robot-assisted surgery. *International journal of computer assisted radiology and surgery*, 13(12):1959–1970, 2018.

[4] Colin Lea, Austin Reiter, René Vidal, and Gregory D. Hager. Segmental spatiotemporal cnns for fine-grained action segmentation. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 36–52, Cham, 2016. Springer International Publishing.

[5] Shuja Khalid, Mitchell Goldenberg, Teodor Grantcharov, Babak Taati, and Frank Rudzicz. Evaluation of deep learning models for identifying surgical actions and measuring performance. *JAMA network open*, 3(3):e201664–e201664, 2020.

[6] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Accurate and interpretable evaluation of surgical skills from kinematic data using fully convolutional neural networks. *International journal of computer assisted radiology and surgery*, 14(9):1611–1617, 2019.

[7] Aneeq Zia and Irfan Essa. Automated surgical skill assessment in rmis training. *International Journal of Computer Assisted Radiology and Surgery*, 13, 12 2017.

[8] Richard K Reznick. Teaching and testing technical skills. *The American journal of surgery*, 165(3):358–361, 1993.

[9] Ben Bridgewater, Anthony D Grayson, Mark Jackson, Nicholas Brooks, Geir J Grotte, Daniel JM Keenan, Russell Millner, Brian M Fabri, and Jones Mark. Surgeon specific mortality in adult cardiac surgery: comparison between crude and risk stratified data. *Bmj*, 327(7405):13–17, 2003.

[10] Sarah E Tevis and Gregory D Kennedy. Postoperative complications and implications on patient-centered outcomes. *journal of surgical research*, 181(1):106–113, 2013.

[11] Marcus E Semel, Stuart R Lipsitz, Luke M Funk, Angela M Bader, Thomas G Weiser, and Atul A Gawande. Rates and patterns of death after surgery in the united states, 1996 and 2006. *Surgery*, 151(2):171–182, 2012.

[12] John F Sweeney. Postoperative complications and hospital readmissions in surgical patients: an important association. *Annals of surgery*, 258(1):19, 2013.

[13] Elise H Lawson, Bruce Lee Hall, Rachel Louie, Susan L Ettner, David S Zingmond, Lein Han, Michael Rapp, and Clifford Y Ko. Association between occurrence of a postoperative complication and readmission: implications for quality improvement and cost savings. *Annals of surgery*, 258(1):10–18, 2013.

[14] Harsha V Polavarapu, Afif N Kulaylat, Susie Sun, and Osama H Hamed. 100 years of surgical education: the past, present, and future. *Bull Am Coll Surg*, 98(7):22–27, 2013.

[15] Rajesh Aggarwal, Oliver T Mytton, Milliard Derbrew, David Hananel, Mark Heydenburg, Barry Issenberg, Catherine MacAulay, Mary Elizabeth Mancini, Takeshi Morimoto, Nathaniel Soper, et al. Training and simulation for patient safety. *BMJ Quality & Safety*, 19(Suppl 2):i34–i43, 2010.

[16] JA Martin, Glenn Regehr, Richard Reznick, Helen Macrae, John Murnaghan, Carol Hutchison, and M Brown. Objective structured assessment of technical skill (osats) for surgical residents. *Journal of British Surgery*, 84(2):273–278, 1997.

[17] Richard M Satava. Virtual reality surgical simulator. *Surgical endoscopy*, 7(3):203–205, 1993.

[18] Simon DiMaio, Mike Hanuschik, and Usha Kreaden. The da vinci surgical system. In *Surgical Robotics*, pages 199–217. Springer, 2011.

[19] Yixin Gao, S Swaroop Vedula, Carol E Reiley, Narges Ahmidi, Balakrishnan Varadarajan, Henry C Lin, Lingling Tao, Luca Zappella, Benjamın Béjar, David D Yuh, et al. Jhu-isi gesture and skill assessment working set (jigsaws): A surgical activity dataset for human motion modeling. In *MICCAI workshop: M2cai*, volume 3, 2014.

[20] Simon Haykin. *Kalman filtering and neural networks*. John Wiley & Sons, 2004.

[21] George A Rovithakis and Manolis A Christodoulou. *Adaptive control with recurrent high-order neural networks: theory and industrial applications*. Springer Science & Business Media, 2012.

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[23] AN Lakhal, AS Tlili, N Benhadj Braiek, et al. Neural network observer for nonlinear systems application to induction motors. *International Journal of Control and Automation*, 3(1):1–16, 2010.

[24] Yaqiang Li, Tianfu Sun, Wentao Zhang, Shechuan Li, Jianing Liang, and Zheng Wang. A torque observer for ipmsm drives based on deep neural network. In *2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pages 1530–1535. IEEE, 2019.

[25] Agapius Bou Ghosn, Philip Polack, and Arnaud de La Fortelle. Learning-based observer evaluated on the kinematic bicycle model. In *2022 10th International Conference on Systems and Control (ICSC)*, pages 423–428. IEEE, 2022.

[26] Jonathan Douissard, Monika E. Hagen, and P. Morel. *The da Vinci Surgical System*, pages 13–27. Springer International Publishing, Cham, 2019.

[27] K Semm. Pelvi-trainer, a training device in operative pelviscopy for teaching endoscopic ligation and suture technics. *Geburtshilfe und Frauenheilkunde*, 46(1):60–62, 1986.

[28] Simon DiMaio, Mike Hanuschik, and Usha Kreaden. *The da Vinci Surgical System*, pages 199–217. Springer, 01 2011.

[29] Fernando Pérez-Escamirosa, Antonio Alarcón-Paredes, Gustavo Adolfo Alonso-Silverio, Ignacio Oropesa, Oscar Camacho-Nieto, Daniel Lorias-Espinoza, and Arturo Minor-Martínez. Objective classification of psychomotor laparoscopic skills of surgeons based on three different approaches. *International Journal of Computer Assisted Radiology and Surgery*, 15(1):27–40, 2020.

[30] Andru Twinanda, Sherif Shehata, Didier Mutter, Jacques Marescaux, Michel De Mathelin, and Nicolas Padoy. Endonet: A deep architecture for recognition tasks on laparoscopic videos. *IEEE Transactions on Medical Imaging*, 36, 02 2016.

[31] Manish Sahu, Anirban Mukhopadhyay, Angelika Szengel, and Stefan Zachow. Tool and phase recognition using contextual CNN features. *CoRR*, abs/1610.08854, 2016.

[32] Dandan Zhang, Ruoxi Wang, and Benny Lo. Surgical gesture recognition based on bidirectional multi-layer independently rnn with explainable spatial feature extraction. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1350–1356. IEEE, 2021.

[33] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[34] Lingling Tao, Luca Zappella, Gregory D Hager, and René Vidal. Surgical gesture segmentation and recognition. In *International conference on medical image computing and computer-assisted intervention*, pages 339–346. Springer, 2013.

[35] Hei Law, Khurshid Ghani, and Jia Deng. Surgeon technical skill assessment using computer vision based analysis. In *Machine learning for healthcare conference*, pages 88–99. PMLR, 2017.

[36] Alvin Goh, David Goldfarb, James Sander, Brian Miles, and Brian Dunkin. Global evaluative assessment of robotic skills: Validation of a clinical assessment tool to measure robotic surgical skills. *The Journal of urology*, 187:247–52, 11 2011.

[37] Amy Jin, Serena Yeung, J. Jopling, J. Krause, D. Azagury, A. Milstein, and Li Fei-Fei. Tool detection and operative skill assessment in surgical videos using region-based convolutional neural networks. *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 691–699, 2018.

[38] Isabel Funke, Sören Torge Mees, Jürgen Weitz, and Stefanie Speidel. Video-based surgical skill assessment using 3d convolutional neural networks. *International journal of computer assisted radiology and surgery*, 14(7):1217–1225, 2019.

[39] Robert DiPietro and Gregory D Hager. Unsupervised learning for surgical motion by learning to predict the future. In *International conference on medical image computing and computer-assisted intervention*, pages 281–288. Springer, 2018.

[40] Robert DiPietro, Narges Ahmidi, Anand Malpani, Madeleine Waldram, Gyusung I Lee, Mija R Lee, S Swaroop Vedula, and Gregory D Hager. Segmenting and classifying activities in robot-assisted surgery with recurrent neural networks. *International journal of computer assisted radiology and surgery*, 14(11):2005–2020, 2019.

[41] Ilker Gurcan and Hien Van Nguyen. Surgical activities recognition using multi-scale recurrent networks. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2887–2891, 2019.

[42] Laura Sbernini, Lucia Rita Quitadamo, Francesco Riillo, Nicola Di Lorenzo, Achille Lucio Gaspari, and Giovanni Saggio. Sensory-glove-based open surgery skill evaluation. *IEEE Transactions on Human-Machine Systems*, 48(2):213–218, 2018.

[43] Giovanni Menegozzo, Diego Dall'Alba, Chiara Zandonà, and Paolo Fiorini. Surgical gesture recognition with time delay neural network based on kinematic data. In *2019 International Symposium on Medical Robotics (ISMR)*, pages 1–7, 2019.

[44] Shahin Sefati, Noah J Cowan, and René Vidal. Learning shared, discriminative dictionaries for surgical gesture segmentation and classification. In *MICCAI Workshop: M2CAI*, volume 4, 2015.

[45] Filippo Cavallo, Stefano Sinigaglia, Giuseppe Megali, Andrea Pietrabissa, Paolo Dario, Franco Mosca, and Alfred Cuschieri. Biomechanics–machine learning system for surgical gesture analysis and development of technologies for minimal access surgery. *Surgical Innovation*, 21(5):504–512, 2014.

[46] Ziheng Wang and Ann Majewicz Fey. Satr-dl: Improving surgical skill assessment and task recognition in robot-assisted surgery with deep neural networks. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 1793–1796. IEEE, 2018.

[47] Jason Kelly, Ashley Petersen, Thomas Lendvay, and Timothy Kowalewski. Bidirectional long short-term memory for surgical skill classification of temporally segmented tasks. *International Journal of Computer Assisted Radiology and Surgery*, 15:2079–2088, 12 2020.

[48] Timothy M Kowalewski, Bryan Comstock, Robert Sweet, Cory Schaffhausen, Ashleigh Menhadji, Timothy Averch, Geoffrey Box, Timothy Brand, Michael Ferrandino, Jihad Kaouk, et al. Crowd-sourced assessment of technical skills for validation of basic laparoscopic urologic skills tasks. *The Journal of urology*, 195(6):1859–1865, 2016.

[49] Piyush Poddar, Narges Ahmidi, S. Swaroop Vedula, Lisa Ishii, Gregory D. Hager, and Masaru Ishii. Automated objective surgical skill assessment in the operating room using unstructured tool motion. *CoRR*, abs/1412.6163, 2014.

[50] Mahtab Jahanbani Fard, Sattar Ameri, Ratna Babu Chinnam, Abhilash K. Pandya, Michael D. Klein, and R. Darin Ellis. Machine learning approach for skill evaluation in robotic-assisted surgery. *CoRR*, abs/1611.05136, 2016.

[51] Daochang Liu, Tingting Jiang, Yizhou Wang, Rulin Miao, Fei Shan, and Ziyu Li. Surgical skill assessment on in-vivo clinical data via the clearness of operating field. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 476–484. Springer, 2019.

[52] Karl-Friedrich Kowalewski, Carly Garrow, Mona Schmidt, Laura Benner, Beat Müller, and Felix Nickel. Sensor-based machine learning for workflow detection and as key to detect expert level in laparoscopic suturing and knot-tying. *Surgical Endoscopy*, 33, 11 2019.

[53] Andrew Hung, Jian Chen, Zhengping Che, Tanachat Nilanon, Anthony Jarc, Micha Titus, Paul Oh, Inderbir Gill, and Yan Liu. Utilizing machine learning and automated performance metrics to evaluate robot-assisted radical prostatectomy performance and predict outcomes. *Journal of Endourology*, 32, 02 2018.

[54] Hamza Asif, Carter McInnis, Frances Dang, Henry Ajzenberg, Peter L. Wang, Adam Mosa, Gary Ko, Boris Zevin, Stephen Mann, and Andrea Winthrop. Objective structured assessment of technical skill (osats) in the surgical skills and technology elective program (sstep): Comparison of peer and expert raters. *The American Journal of Surgery*, 2021.

[55] Malik Benmansour, Wahida Handouzi, and Abed Malti. Task-specific surgical skill assessment with neural networks. In *International Conference on Advanced Intelligent Systems for Sustainable Development*, pages 159–167. Springer, 2018.

[56] Malik Benmansour and Abed Malti. Simple and efficient recurrent neural network to evaluate classified surgery tasks. In *5th International Conference on Automation, Control Engineering and Computer Science-ACECS*, 2018.

[57] Razvan Pascanu, Tomás Mikolov, and Yoshua Bengio. Understanding the exploding gradient problem. *CoRR*, abs/1211.5063, 2012.

[58] Malik Benmansour and Abed Malti. Skills evaluation of specific surgical tasks using long short term memory networks. In *International Conference on Advanced Intelligent Systems for Sustainable Development*, pages 331–339. Springer, 2019.

[59] Malik Benmansour, Abed Malti, and Pierre Jannin. Deep neural network architecture for automated soft surgical skills evaluation using objective structured assessment of technical skills criteria. *International Journal of Computer Assisted Radiology and Surgery*, pages 1–9, 2023.

[60] Hazel Doughty, Dima Damen, and Walterio Mayol-Cuevas. Who's better? who's best? pairwise deep ranking for skill determination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6057–6066, 2018.

[61] Narges Ahmidi, Piyush Poddar, Jonathan D Jones, S Swaroop Vedula, Lisa Ishii, Gregory D Hager, and Masaru Ishii. Automated objective surgical skill assessment in the operating room from unstructured tool motion in septoplasty. *International journal of computer assisted radiology and surgery*, 10(6):981–991, 2015.

[62] Aneeq Zia, Yachna Sharma, Vinay Bettadapura, Eric L Sarin, and Irfan Essa. Video and accelerometer-based motion analysis for automated surgical skills assessment. *International journal of computer assisted radiology and surgery*, 13(3):443–455, 2018.

[63] Neil Vaughan and Bogdan Gabrys. Scoring and assessment in medical vr training simulators with dynamic time series classification. *Engineering Applications of Artificial Intelligence*, 94:103760, 2020.

[64] Zhenqiang Li, Yifei Huang, Minjie Cai, and Yoichi Sato. Manipulation-skill assessment from videos with spatial attention network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[65] Xuan Anh Nguyen, Damir Ljuhar, Maurizio Pacilli, Ramesh Mark Nataraja, and Sunita Chauhan. Surgical skill levels: Classification and analysis using deep neural network model and motion signals. *Computer methods and programs in biomedicine*, 177:1–8, 2019.

[66] Dayvid Castro, Danilo Pereira, Cleber Zanchettin, David Macêdo, and Byron LD Bezerra. Towards optimizing convolutional neural networks for robotic surgery skill evaluation. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.

[67] Nguyen Xuan Anh, Ramesh Mark Nataraja, and Sunita Chauhan. Towards near real-time assessment of surgical skills: A comparison of feature extraction techniques. *Computer methods and programs in biomedicine*, 187:105234, 2020.

[68] Dandan Zhang, Zicong Wu, Junhong Chen, Anzhu Gao, Xu Chen, Peichao Li, Zhaoyang Wang, Guitao Yang, Benny Lo, and Guang-Zhong Yang. Automatic microsurgical skill assessment based on cross-domain transfer learning. *IEEE Robotics and Automation Letters*, 5(3):4148–4155, 2020.

[69] Tae Soo Kim, Molly O'Brien, Sidra Zafar, Gregory D Hager, Shameema Sikder, and S Swaroop Vedula. Objective assessment of intraoperative technical skill in capsulorhexis using videos of cataract surgery. *International journal of computer assisted radiology and surgery*, 14(6):1097–1105, 2019.

[70] Evangelos B Mazomenos, Kamakshi Bansal, Bruce Martin, Andrew Smith, Susan Wright, and Danail Stoyanov. Automated performance assessment in transoesophageal echocardiography with convolutional neural networks. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 256–264. Springer, 2018.

[71] Alvin C Goh, David W Goldfarb, James C Sander, Brian J Miles, and Brian J Dunkin. Global evaluative assessment of robotic skills: validation of a clinical assessment tool to measure robotic surgical skills. *The Journal of urology*, 187(1):247–252, 2012.

[72] Melina C Vassiliou, Liane S Feldman, Christopher G Andrew, Simon Bergman, Karen Leffondré, Donna Stanbridge, and Gerald M Fried. A global assessment tool for evaluation of intraoperative laparoscopic skills. *The American journal of surgery*, 190(1):107–113, 2005.

[73] Gitta H Lubke and Bengt O Muthén. Applying multigroup confirmatory factor models for continuous outcomes to likert scale data complicates meaningful group comparisons. *Structural equation modeling*, 11(4):514–534, 2004.

[74] Gail M Sullivan and Anthony R Artino Jr. Analyzing and interpreting data from likert-type scales. *Journal of graduate medical education*, 5(4):541–542, 2013.

[75] Susan Jamieson. Likert scales: How to (ab) use them? *Medical education*, 38(12):1217–1218, 2004.

[76] James Carifio and Rocco Perla. Resolving the 50-year debate around using and misusing likert scales, 2008.

[77] Qingsong Wen, Liang Sun, Fan Yang, Xiaomin Song, Jingkun Gao, Xue Wang, and Huan Xu. Time series data augmentation for deep learning: A survey. *arXiv preprint arXiv:2002.12478*, 2020.

[78] Arthur Le Guennec, Simon Malinowski, and Romain Tavenard. Data augmentation for time series classification using convolutional neural networks. In *ECML/PKDD workshop on advanced analytics and learning on temporal data*, 2016.

[79] Eamonn Keogh, Selina Chu, David Hart, and Michael Pazzani. An online algorithm for segmenting time series. In *Proceedings 2001 IEEE international conference on data mining*, pages 289–296. IEEE, 2001.

[80] Gautier Pialla, Maxime Devanne, Jonathan Weber, Lhassane Idoumghar, and Germain Forestier. Data augmentation for time series classification with deep learning models. *Advanced Analytics and Learning on Temporal Data (AALTD)*, 2022.

[81] Kim Phil. Matlab deep learning with machine learning, neural networks and artificial intelligence. *Apress, New York*, 2017.

[82] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318. Pmlr, 2013.

[83] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[84] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

[85] Francois Chollet. Keras. https://keras.io, 2015.

[86] Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014.

[87] Ibrahem Kandel and Mauro Castelli. The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset. *ICT express*, 6(4):312–315, 2020.