Neural network modelling of kinematic and dynamic features for signature verification

Moises Diaz [a],[*], Miguel A. Ferrer [a], Jose Juan Quintana [a], Adam Wolniakowski [b], Roman Trochimczuk [b], Kanstantsin Miatliuk [c], Giovanna Castellano [d], Gennaro Vessio [d]

[a] *Instituto Universitario para el Desarrollo Tecnológico y la Innovación en Comunicaciones, Universidad de Las Palmas de Gran Canaria, Las Palmas de Gran Canaria, Spain*
[b] *Department of Electrical Engineering, Białystok University of Technology, Białystok, Poland*
[c] *Department of Mechanical Engineering, Białystok University of Technology, Białystok, Poland*
[d] *Department of Computer Science, University of Bari Aldo Moro, Bari, Italy*

ARTICLE INFO

ABSTRACT

Online signature parameters, which are based on human characteristics, broaden the applicability of an automatic signature verifier. Although kinematic and dynamic features have previously been suggested, accurately measuring features such as arm and forearm torques remains challenging. We present two approaches for estimating angular velocities, angular positions, and force torques. The first approach involves using a physical UR5e robotic arm to reproduce a signature while capturing those parameters over time. The second method, a cost-effective approach, uses a neural network to estimate the same parameters. Our findings demonstrate that a simple neural network model can extract effective parameters for signature verification. Training the neural network with the MCYT300 dataset and cross-validating with other databases, namely, BiosecurID, Visual, Blind, OnOffSigDevanagari-75 and OnOffSigBengali-75 confirm the model's generalization capability. The trained model is available at: https://github.com/gvessio/SignatureKinematics.

1. Introduction

An *online signature* is typically represented by the parametric equations of its trajectory, denoted as $(x(t), y(t))$, which are captured when the signing tool contacts the digital surface. Additionally, some digitizers capture other function-based parameters, such as the vertical pressure exerted by the pen tip, azimuthal and altitude angles of the pen, and even the pen's in-air trajectory. As a physiological biometric trait, a signature is used in various applications, including access control, commercial transactions, document forgery detection, and the provision of evidence in legal scenarios such as the verification of last wills [1]. In biometrics, where impostors may attempt to forge signatures with varying degrees of skill, robust verification methods are crucial.

Since the execution of a signature inherently involves movements of the hand, arm, and forearm, it is hypothesized that these motions may contain kinematic and dynamic unique characteristic of the signer [2]. From a kinematic perspective, this action can be characterized by the arm's angular position, $\theta(t)$, and angular velocity, $\omega(t)$. Dynamically, these movements are facilitated by force torques, $\tau(t)$, applied at the joints.

One method used to obtain this valuable biomechanical information involves a physical robot programmed to mimic the act of signing. While a robot's ability to accurately replicate these movements depends on its configuration, working area, and degrees of freedom, it can effectively capture kinematic and dynamic features during the process. However, accessing these robots is costly and cumbersome.

To improve access to robotic features, our contributions are:

- The angular positions, $\theta_r(t) \in (\theta_1(t), \dots, \theta_6(t))$, angular velocities, $\omega_r(t) \in (\omega_1(t), \dots, \omega_6(t))$, and force torques, $\tau_r(t) \in (\tau_1(t), \dots, \tau_6(t))$ of a UR5e robotic were recorded during the execution of the 16,500 online signatures from the MCYT330 dataset (DS1).
- We trained a multilayer perceptron (MLP)-based neural network to estimate the kinematic and dynamic features $(\hat{\theta}(t), \hat{\omega}(t), \hat{\tau}(t))$ by taking the trajectory of a signature $(x(t), y(t))$ as input.
- We conducted a two-fold experiment: first, we assessed the MLP's estimation capacity, and secondly, we compared the performance of features generated by the UR5e robot against those produced by the MLP in a dynamic time warping (DTW)-based automatic

* Corresponding author.
E-mail addresses: moises.diaz@ulpgc.es (M. Diaz), miguelangel.ferrer@ulpgc.es (M.A. Ferrer), josejuan.quintana@ulpgc.es (J.J. Quintana), a.wolniakowski@pb.edu.pl (A. Wolniakowski), r.trochimczuk@pb.edu.pl (R. Trochimczuk), k.miatliuk@pb.edu.pl (K. Miatliuk), giovanna.castellano@uniba.it (G. Castellano), gennaro.vessio@uniba.it (G. Vessio).
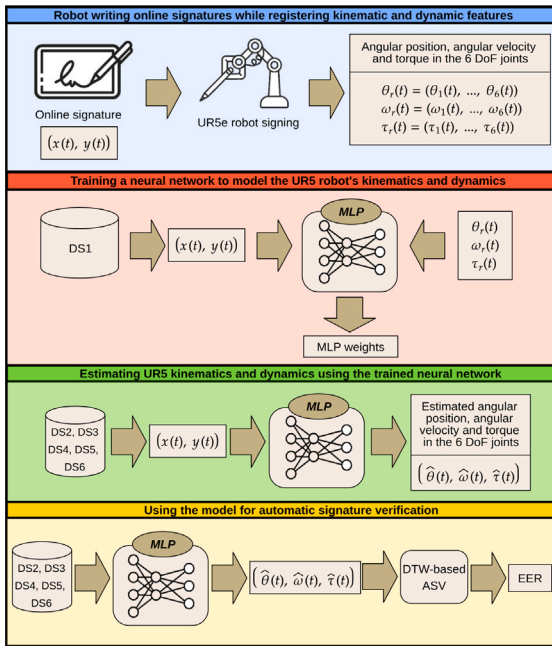
Fig. 1. Overview of the proposed method.

signature verifier (ASV). Additionally, we demonstrated the neural network's generalization capability by estimating effective kinematic and dynamic features in five third-party databases, namely, BiosecurID (DS2), Visual (DS3), Blind (DS4), OnOffSig-Devanagari-75 (DS5) and OnOffSigBengali-75 (DS6).

Fig. 1 illustrates the methodology employed in this study. The trained MLP model is publicly available for research and practical applications.

The rest of this paper is organized as follows. Section 2 reviews prior work. Section 3 details the proposed method. Section 4 presents the signature verification experiments conducted. Section 5 concludes the article.

2. Related work

2.1. Robots replicating human writing trajectories

Various robotic arms have been developed to replicate human writing trajectories, drawing significant interest from the art world and at technical exhibitions. For example, the Paul planar robotic arm is designed to create artistic, human-like drawings on traditional media such as pen and paper [3]. Additionally, two-dimensional plotters such as Line-us and iDraw have become increasingly popular for automated writing tasks, demonstrating their ability to produce physically forged trajectories, as noted in [4]. Contrastingly, NAO robots serve a different functional purpose. These robots have been programmed effectively in various educational studies to assist children in enhancing their calligraphy skills, blending robotics with pedagogy [5]. Similarly, the UR5e robot, known for its precision and versatility, has been demonstrated to accurately replicate the trajectory and velocity of human signatures, making it a valuable tool in signature verification research [6]. We have also used UR5e in this work to further explore its application.

2.2. Machine learning to estimate function-based features

Recent advances in machine learning, particularly with neural networks, have significantly improved the analysis of function-based features across various domains, including biometrics and robotics. These

models are especially effective in handling data characterized by complex temporal sequences and variability between executions. For instance, Recurrent Neural Networks have been widely applied to model time-dependent data across diverse fields, ranging from financial forecasting to motion analysis [7–9]. In biometrics, the predictive power of a hybrid CNN-GRU network has been used to effectively distinguish between Parkinsonian and healthy handwriting [10]. A similar approach has been used in robotics to estimate object trajectories for robot manipulation [11]. In this paper, however, we preferred a simple approach based on an MLP and a sliding window over features. Our goal was not specifically to advance state-of-the-art sequence modelling, but rather, to develop a model robust enough to estimate kinematics and dynamics effectively and apply them across different datasets.

2.3. Robotic features in signature verification

One advantage of our work is the estimation of robotic features based on the movement of the UR5e robotic arm when signing, which represents a novel approach in comparison to recent studies [12]. This method leverages real-world robotic data to enhance feature extraction, offering a novel perspective in the field. However, a limitation of our approach is that it relies on a single robot for all signature data. This means our method does not account for individual variations in arm and forearm anatomy among different writers, a factor not captured in the signature datasets.

Robotic features are increasingly being investigated in the context of signature verification to enhance the security capabilities of automatic verification systems. In [13], the authors explored the complexities of inter-joint motion in hand skeleton movements, which are challenging to reproduce. Their research focused on replay attacks involving a robotic hand replicating an in-air signature, highlighting the difficulties in accurately simulating human hand dynamics. Further studies have analysed the kinematics of a virtual skeleton arm during the signature execution process. An IRB120 ABB robotic model aimed at examining the angular position of the joints was used in [2], demonstrating significant improvements in signature verification performance through precise emulation of joint movements.

Investigations carried out thus far have primarily focused on describing the motion of joints, but information about the forces causing these movements has not been extensively explored. In the present work, we introduce the application of torque measurements, a novel approach that considers the force torques required to execute joint rotations in real robots during signature execution. Previous studies [14,15] introduced the concept of torques, which are derived from the geometric properties of written trajectories, to this field. In this article, torques refer specifically to the force torques required to execute rotations in the joints of real robots for signature execution.

Although features extracted from digitizers [16] or engineered features such as curvature and torsion values [17] are well-established, the features proposed in this work introduce a novel feature space in the field of signature verification.

3. Proposed method

Our method aims to estimate the kinematic and dynamic features of a signature, as depicted in Fig. 1. First, we use the UR5e robotic arm to extract these features from signatures. Then, the features are used to train an MLP neural network. Once its weights are adjusted, the MLP is used to estimate these features from the trajectory of a new signature.

3.1. Kinematic and dynamic features with the robotic arm

We utilized the Universal Robots UR5e anthropomorphic robot arm[1] as the platform for replicating signatures and acquiring kinematic and

---

[1] https://www.universal-robots.com/products/ur5-robot/.

dynamic features. The UR5e is commonly employed in industrial and service tasks requiring human collaboration. This robot arm can lift payloads of up to 5 kg, extend to a maximum reach of 850 mm, and operate with 6 degrees of freedom, enabling precise positioning of the end effector in any desired orientation and position.

The Forward Kinematics (FK) of the robotic kinetic chain was modelled using Denavit–Hartenberg (D–H) parameters [18]. These parameters define the relative positioning of consecutive elements within the chain. The relative poses between the frames of reference attached to the subsequent $(i-1)$-th and ith links of the robot can be derived using the D–H parameters through four atomic movements:

$$^{i-1}_i T = \mathbf{Trot_z}(\theta_i)\mathbf{Ttransl_z}(d_i) \cdot \mathbf{Ttransl_x}(a_i)\mathbf{Trot_x}(\alpha_i), \tag{1}$$

where $\mathbf{Trot}$ and $\mathbf{Ttransl}$ represent rotation and translation transformations around the z and x axes [19].

The full FK model of the UR5e arm can, therefore, be expressed as follows:

$$^b_e T = \prod_{i=1}^{6}{}^{i-1}_i T(a_i, \alpha_i, d_i, \theta_i), \tag{2}$$

where b denotes the base frame of the arm, while e is the end frame. The FK model of the robot calculates the pose of the end effector given the robot's configuration vector $\theta_r(t) = [\theta_1(t), \dots, \theta_6(t)]$. Conversely, the Inverse Kinematics (IK) [20] involves determining the robot configuration θ when $^b_e T$ is given. We used the analytical model for the FK and the numerical solution for the IK.

Determining the joint torques τ needed to achieve the desired trajectory profile $[\theta(t), \dot{\theta}(t), \ddot{\theta}(t)]$ is the goal of Inverse Dynamics (ID). Accurate implementation often requires an exact knowledge of the robot's dynamic parameters, such as the mass distribution and inertia tensors, which are often inaccessible or inconsistently measured. Given that the UR5e robot lacks built-in force–torque sensors, we opted for estimating torques from motor currents during robot trajectory execution, such as $\tau_i = r_i \cdot K_{I,i} \cdot I_i$, where τ_i is the torque on the ith motor, r_i is the corresponding gear ratio, $K_{I,i}$ is the motor-specific torque-to-current coefficient, and I_i is the current supplied to the motor. The K_I coefficients, pre-calibrated by the robot manufacturer, are retrievable from the robot controller's configuration files. For our experiments, the coefficients were set as $K_I = [0.1094, 0.1100, 0.1097, 0.0820, 0.0822, 0.0824]$, with gear ratios $r = [101, 101, 101, 101, 101, 101]$, as provided by the manufacturer in the calibration files.

The robot arm's control was achieved through a custom ROS package developed at Bialystok University of Technology.[2] We used a list of timed waypoints $[t, x, y, z]$ that were imbued with the desired pen orientation. The pen was assumed to align perpendicular to the writing surface. Next, we solved the IK task using a numerical Jacobian-based algorithm to establish the robot's list of timed configuration vectors $[t, \theta]$. We used the list of configurations to construct a joint-space spline interpolator. This interpolator generates the immediate desired configurations, sent directly to the robot controller at a control frequency of 125 Hz. To achieve the desired accuracy, we had to modify the default robot controller parameters to $k_{\text{gain}} = 2000$ and $t_{\text{lookahead}} = 0.03$ s. The robot controller responded with the current joint state data, including the angular positions $\theta_r(t)$, velocities $\omega_r(t)$, and actual motor currents $I_r(t)$. We computed the torques $\tau_r(t)$ based on the currents.

3.2. Estimation of the kinematic and dynamic features with the neural network model

In order to find a more cost-effective alternative that eliminates the need for a physical robot, the objective is to develop a system capable of estimating the angular velocities $\theta(t)$, angular positions $\omega(t)$,

and force torques $\tau(t)$ of the robotic arm by leveraging the (x, y)-coordinates of points sampled from signatures. This effort, framed as a *multi-target regression*, seeks to map the 2-dimensional space of signature coordinates, $\mathbb{R}^2$, to an 18-dimensional output space, $\mathbb{R}^{18}$, where each dimension corresponds to one of the six values for $\theta(t)$, $\omega(t)$, and $\tau(t)$. The primary challenge of this task lies in accurately estimating these eighteen output values for each point based solely on the two input features despite the inherent variability in the lengths and styles of signatures.

The input features were normalized to the $[0, 1]$ range via min–max scaling, thereby maintaining generalizability across different datasets. Target values were subjected to a similar scaling process to the $[0, 1]$ range. Considering the broad spectrum of target values, this reversible scaling method preserves the model's general applicability and facilitates the learning process. Crucially, the parameters used for scaling were exclusively determined from the training datasets, ensuring that their subsequent application to the test sets would not result in information leakage.

We adopted an MLP for its adaptability in managing tasks that require multiple simultaneous outputs [21]. To enrich the model with sequential information, each signature point was not considered in isolation; instead, for each training instance, we included the (x, y)-coordinates of the point itself, along with the coordinates of the five preceding and succeeding points. This *sliding window* approach, whose horizon was chosen based on its effectiveness in preliminary trials, incorporated contextual information surrounding each point, enhancing the MLP's ability to capture the dynamics of the signature movement. Specifically, given an input vector $\mathbf{x}$, the MLP performs a series of operations to estimate the outputs for $\hat{\theta}(t)$, $\hat{\omega}(t)$, and $\hat{\tau}(t)$. The architecture features a ReLU-activated hidden layer:

$$\mathbf{h} = \text{ReLU}(\mathbf{W}_h\mathbf{x} + \mathbf{b}_h), \tag{3}$$

with twelve units followed by a dropout layer (dropout rate of 0.3) to mitigate overfitting [22]:

$$\mathbf{h}' = \text{Dropout}(0.3, \mathbf{h}). \tag{4}$$

Uniquely, the model was structured with three separate output "heads", each comprising six units, facilitating the concurrent estimation of the three targets. Each output j (for $j \in \{\theta, \omega, \tau\}$) is defined as:

$$\mathbf{y}_j = \sigma(\mathbf{W}_j\mathbf{h}' + \mathbf{b}_j), \tag{5}$$

where σ denotes the sigmoid activation function, ensuring that the estimated values are bounded within the $[0, 1]$ range. Our aim was to strike a balance between effectiveness and simplicity, grounded on common design choices.

The designated loss function was a composite loss, calculated as the sum of three mean squared error (MSE) losses, each corresponding to one of the model's output heads. Mathematically, if L_θ, L_ω, and L_τ represent the MSE losses for the output heads dedicated to θ, ω, and τ respectively, the total loss L_{total} is formulated as:

$$L_{total} = L_\theta + L_\omega + L_\tau, \tag{6}$$

where $L_j = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_{i,j} - y_{i,j})^2$ for $j \in \{\theta, \omega, \tau\}$, n is the number of samples, $\hat{y}_{i,j}$ is the estimated value, and $y_{i,j}$ is the true value for the ith sample of the jth output.

4. Experiments

The experiments aimed to analyse whether the features extracted from the MLP are similar to those extracted from the UR5e, both in terms of estimation and performance in ASV.

4.1. Datasets

Six signature databases were used to extract dynamic and kinematic features, train the MLP network, and validate the generalization of our proposed model. First, we used the MCYT330 sub-corpus [23] (DS1), which consists of signatures from 330 users, each providing 25 genuine and 25 forged signatures across two sessions. The BiosecurID [24] (DS2) multimodal database contains online signature data from 132 users. Each user contains 16 genuine and 12 skilled forgeries. The Visual [25] (DS3) sub-corpus includes data from 94 users, each providing 20 genuine signatures from two sessions and 10 skilled forgeries. The Blind sub-corpus [25] (DS4) includes 88 users, each providing 10 genuine signatures and 10 skilled forgeries. Finally, the OnOffSigDevanagari-75 (DS5) and OnOffSigBengali-75 (DS6) datasets [26] each have 75 users, with 24 genuine signatures per user. These databases provided diverse environments to test the adaptability and effectiveness of our method across different signature capture technologies and forgery scenarios.

4.2. Automatic signature verification

The automatic signature verifier developed in this study comprises two key components: function-based features and the verification mechanism itself.

The function-based features were organized into matrices with N columns and M rows. These columns represent concatenated function-based features that describe various aspects of the signature, while the rows correspond to the number of sampling points, which vary depending on the digitizer's frequency and the signature's length and duration. As such, while the number of columns was fixed across all experiments, the number of rows fluctuated with each signature. Following [2], each feature within the matrix was first normalized within the range $[0, 1]$. To enrich the feature set, each feature's first and second derivatives were calculated using a second-order regression and incorporated into the feature matrix, which was then standardized using z-score normalization.

The UR5e robot provided six initial function-based features for each degree of freedom. In the case of the angular position $\theta_r(t)$, a comprehensive feature matrix was developed. The same approach was applied to angular velocity $\omega_r(t)$ and force torques $\tau_r(t)$, and to their estimation provided by the MLP model. Notably, since the pen attached to the axis of the sixth joint of the UR5e did not rotate around its axis, the angular velocity $\omega_6(t)$ remained constant, and was consequently excluded from the verification process.

For classification, a standard implementation of the Dynamic Time Warping (DTW) [27] algorithm was used to optimize the Euclidean distance between two feature matrices. The computation was accelerated by applying the Sakoe–Chiba band [27], limiting the search width to $M/10$ of the diagonal. The verification process involves comparing a questioned signature q against a set of R reference signatures. The minimum DTW distance quantifies the relationship of q to the reference set: $s_R(q) = \arg\min_{r \in R}[DTW(q, r)]$, where r is a reference signature, and $s_R(q)$ is the non-normalized score.

Two stages of normalization were subsequently applied to these scores. The first stage uses the warping path length $|p|$ to normalize scores for detecting random forgeries: $\hat{s}_{R_1}(q) = s_R(q)/|p|$. In the second stage, a weighted factor μ_R, derived from all reference signatures of the user, is used to normalize scores for detecting skilled forgeries: $\hat{s}_{R_2}(q) = s_R(q)/\mu_R$.

4.3. Experimental setup and metrics

For feature estimation using the MLP, we employed a four-model approach to estimate each signature in the DS1 dataset. Each model used identical hyperparameters, but was tested on distinct quarters of the dataset. The outputs from these models were averaged to produce

Table 1
UR5e kinematic and dynamic estimation performances.

Model	Parameter	MAE	MSE
MLP	Angular velocities, $\theta(t)$	0.0041	0.0001
	Angular positions, $\omega(t)$	0.0571	0.0063
	Force torques, $\tau(t)$	0.0126	0.0004
RNN	Angular velocities, $\theta(t)$	0.1237	0.0663
	Angular positions, $\omega(t)$	0.0276	0.0015
	Force torques, $\tau(t)$	3.1420	23.2520
LSTM	Angular velocities, $\theta(t)$	0.1283	0.0694
	Angular positions, $\omega(t)$	0.0268	0.0015
	Force torques, $\tau(t)$	3.0116	20.1721
GRU	Angular velocities, $\theta(t)$	0.1370	0.0800
	Angular positions, $\omega(t)$	0.0285	0.0016
	Force torques, $\tau(t)$	3.8371	34.4615

the final results. To assess generalizability, a final model was trained on the entire DS1 dataset. Training utilized the Adam optimizer [28] with a commonly used learning rate of 0.01. An early stopping mechanism based on the validation loss observed on 20% of the training set was employed to prevent overfitting. The patience parameter was set to 1 epoch, leading to convergence within a few epochs. Performance metrics included the mean absolute error (MAE) and mean squared error (MSE).

For signature verification, we approached the *random forgeries* and *skilled forgeries* experiments, following the common standard of the ICDAR 2021 signature competition [29]. We used five random genuine signatures from each writer as reference signatures, while the remaining genuine signatures served as test samples. For random forgery, one genuine signature from each user was randomly selected from all other users. As recommended in [29], all the skilled forgeries were used in the test, but not employed as reference negative signatures. Finally, to illustrate the system's efficacy at various False Acceptance Rate (FAR) and False Rejection Rate (FRR) levels, the Detection Error Trade-off (DET) curves were shown on a logarithmic scale. As is common in signature verification [1], we used the Equal Error Rate (EER) to evaluate the performance. Note that the EER represents the point on the DET plot where the FAR and FRR are equal. These experiments were conducted ten times, the DET curves were averaged, and the EER is reported as the average ± standard deviation for each case.

4.4. Feature estimation results

The UR5e kinematic and dynamic estimation results are summarized in Table 1. To explore alternatives, we show the results obtained not only with the proposed MLP model but also with state-of-the-art techniques, including Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU) models, which are well-known for their effectiveness in sequence prediction tasks.

Specifically, the RNN, LSTM, and GRU architectures employed in this study consist of a bidirectional layer with 32 units followed by a time-distributed layer with 18 output units, corresponding to the six values for $\theta(t)$, $\omega(t)$, and $\tau(t)$. The experimental setup for these models was the same as for the MLP to ensure a fair comparison. The only difference was the additional preprocessing step needed to pad the sequences to ensure uniform length, as these models accept sequences as input instead of sliding windows of points.

The findings reveal distinct differences in model performance across the various parameters. The MLP significantly outperforms the RNN, LSTM, and GRU models in estimating angular velocities and force torques, with much lower MAE and MSE values. This suggests that the sliding window approach used in the MLP effectively captures the local context of signature points, which is crucial for accurately estimating these parameters. Conversely, the sequential models may struggle to capture this fine-grained context, leading to higher errors. For angular
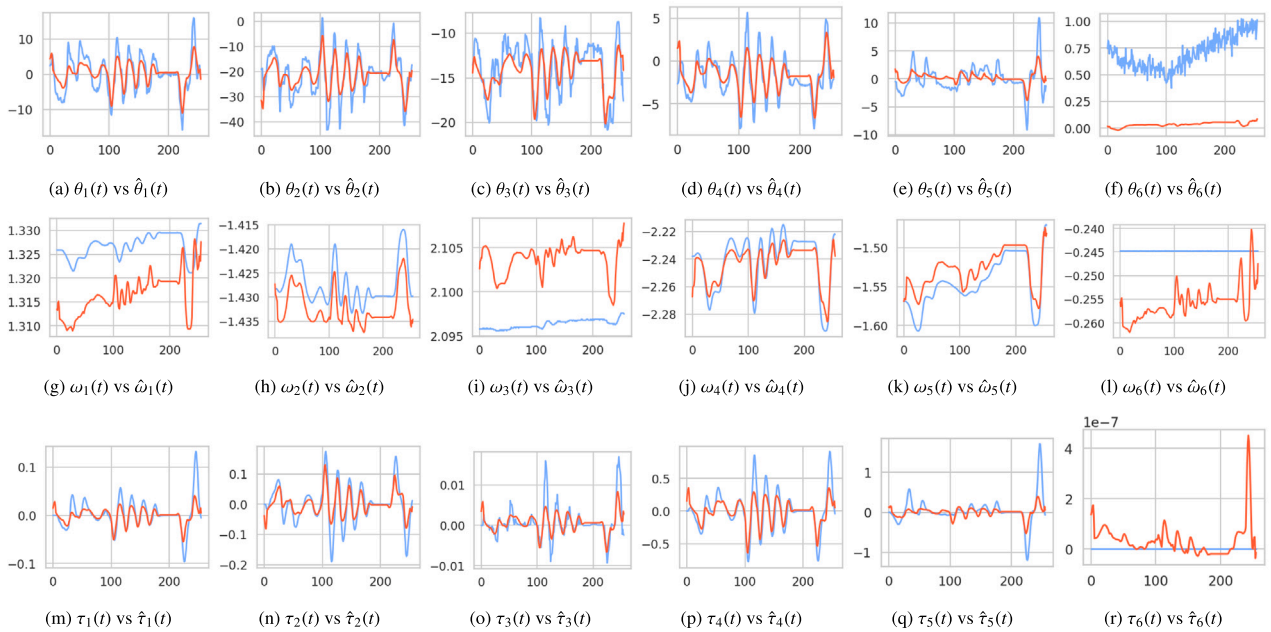
Fig. 2. Visual representation of the sequences of angular positions $\theta(t)$, angular velocities $\omega(t)$, and force torques $\tau(t)$ for signature 005f01 of DS1. The sequences obtained by the UR5e are shown with a blue line, while those estimated by the MLP are shown with a red line.

positions, the RNN, LSTM, and GRU models exhibit slightly better performance than the MLP. This indicates that angular positions benefit from the sequential nature of these models, which can better capture temporal dependencies in the data.

Overall, the results showcase high accuracy in the model's estimations, especially with regard to angular velocities and force torques. However, numerical results alone do not fully allow to evaluate the model's proficiency in accurately capturing the underlying trends over time. To address this, Fig. 2 visually compares the model's estimated trends against actual data. The model demonstrates high accuracy in estimating angular velocities and force torques. However, there are some minor deviations in amplitude and fluctuations along the *y*-axis. The sixth value for the angular velocities and force torques is challenging to estimate as the actual values remain relatively constant. Although the model shows an overall accurate replication of the function behaviour for angular positions, there is a slight offset. In summary, the model is somewhat accurate, but does not always perfectly mimic the function behaviour.

4.5. Online signature verification results

The effectiveness of the method is demonstrated in two ways. First, Fig. 3 shows that features generated by the neural network perform similarly to, or even better than, those obtained from the robot at the EER level. In the figure, grey lines represent baseline results using function-based features directly extracted from the UR5e robot: angular positions, θ, are depicted with solid lines, angular velocities, ω, with dashed lines, and torques, τ, with dotted lines. The blue lines represent performance using estimated features with the same database (DS1). Comparing the results for each type of features, we observe almost similar results on θ. However, in the case of ω and τ, the EER was always better when these sequences were extracted from the neural network. Also, corresponding DET curves from MLP are always below the UR5e ones. These results satisfy our requirements with the neural network, which are consistent in both random and skilled forgeries. It could be said that although the UR5e robotic arm provides precise measurements, it can result in overfitting while performing signature verification. Moreover, the MLP model is designed to generalize better from training data to unseen data, even though it uses estimated features. This improves its performance and helps prevent overfitting.

Second, Fig. 4 examines the generalization capacity of the MLP model, trained solely with data from DS1, and further tested with signatures from DS2 to DS6, which were never processed by the UR5e robot. This created a challenging situation for evaluating the utility of the estimated features. The findings revealed that in DS2, while there was a significant decline in EER for θ in random forgeries, ω and τ delivered results comparable to DS1. Similar outcomes were observed in skilled forgeries for DS2, with angular velocities and force torques showing improved EERs. For DS3, the angular position again displayed outlying results in random forgeries; however, angular velocities and torques performed slightly better. In skilled forgeries, stable EERs comparable to DS2 were achieved with angular velocities and torques. For DS4, we observed better performance for θ in both random and skilled forgeries, with slightly higher results for skilled forgeries in angular velocities and torques. DS5 and DS6 reported similar performance in random forgeries, consistent with the other results. Note that these two datasets did not include skilled forgeries. These results underscore the robustness of the angular velocity and force torques obtained by the MLP across different datasets and forgery scenarios.

5. Conclusions and future works

We introduced a neural network model to estimate the kinematic and dynamic features of a robotic arm. We trained the model using real robotic features and demonstrated its practical application in signature verification. The MLP demonstrated the ability to estimate both kinematic and dynamic features from the trajectory of unseen signatures. Furthermore, the results also showed that the neural network used can reduce the need for experimentation with a real robot to extract features due to its generalization in ASV. The robotic features and the model are publicly available for further research.

For future works, one promising direction is to develop a custom robotic arm tailored specifically for signature verification, potentially replicating the kinematics and dimensions of the human arm based on anthropometric data. It could enhance the accuracy of signature analysis by capturing the nuances of signature dynamics more precisely. A potential limitation of our work is the assumption that the same robot configuration is suitable for all signers. While personalizing the robot model for each signer might better capture individual human
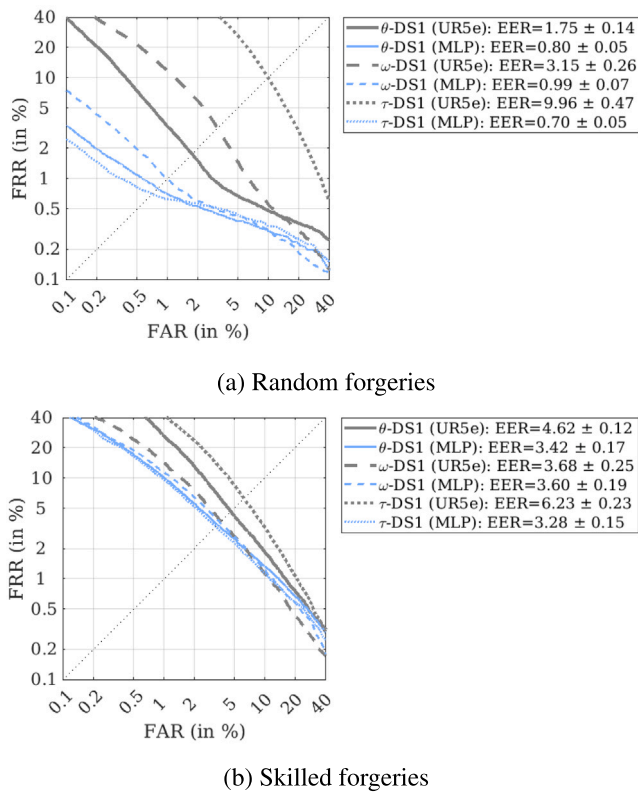
(a) Random forgeries

(b) Skilled forgeries

Fig. 3. Comparison of performance between the baseline using UR5e features and estimated features, illustrated with DET plots and EER for signature verification results.
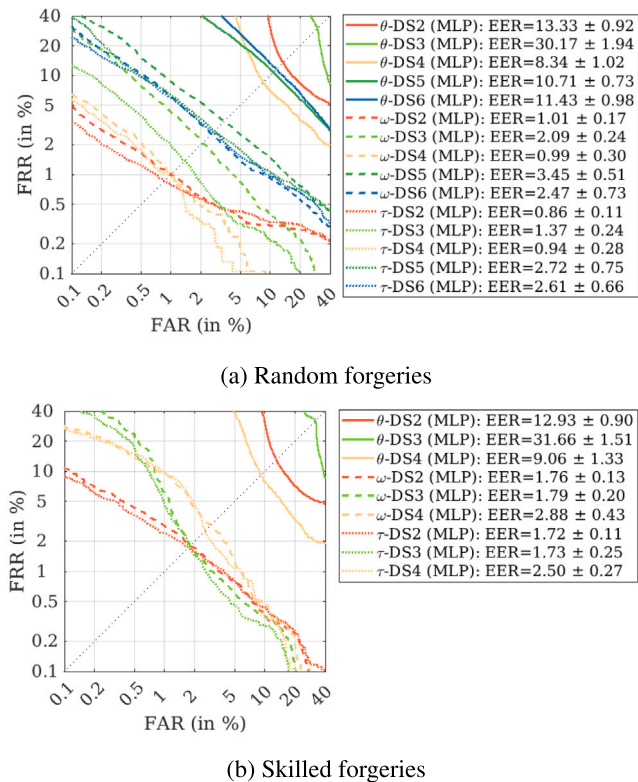

(a) Random forgeries

(b) Skilled forgeries

Fig. 4. Performance results across different databases, trained with DS1 data, illustrated through DET plots and EER values for signature verification.

variability, it is uncertain whether this would actually improve the performance of the automatic signature recognizer. Additionally, investigating interpretable neural network models could offer valuable insights into the decision-making process. Further improvements include attention mechanisms to refine model performance. These advancements could significantly improve the effectiveness and adaptability of the signature verification system.

CRediT authorship contribution statement

Moises Diaz: Writing – review & editing, Writing – original draft, Funding acquisition, Formal analysis, Conceptualization. **Miguel A. Ferrer:** Supervision, Investigation, Funding acquisition, Conceptualization. **Jose Juan Quintana:** Visualization, Methodology, Investigation. **Adam Wolniakowski:** Software, Resources, Investigation, Conceptualization. **Roman Trochimczuk:** Writing – original draft, Validation, Software. **Kanstantsin Miatliuk:** Supervision, Methodology, Formal analysis. **Giovanna Castellano:** Methodology, Investigation, Conceptualization. **Gennaro Vessio:** Writing – review & editing, Writing – original draft, Data curation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

Data availability

Data will be made available on request.

References

[1] M. Faundez-Zanuy, J. Fierrez, M.A. Ferrer, M. Diaz, R. Tolosana, R. Plamondon, Handwriting biometrics: Applications and future trends in e-security and e-health, Cogn. Comput. 12 (2020) 940–953.
[2] M. Diaz, M.A. Ferrer, J.J. Quintana, Anthropomorphic features for on-line signatures, IEEE Trans. Pattern Anal. Mach. Intell. 41 (12) (2019) 2807–2819.
[3] P. Tresset, F.F. Leymarie, Portrait drawing by Paul the Robot, Comput. Graph. 37 (5) (2013) 348–363.
[4] J.J. Bird, A. Naser, A. Lotfi, Writer-independent signature verification; Evaluation of robotic and generative adversarial attacks, Inform. Sci. 633 (2023) 170–181.
[5] S. Chandra, P. Dillenbourg, A. Paiva, Children teach handwriting to a social robot with different learning competencies, Int. J. Soc. Robotics 12 (3) (2020) 721–748.
[6] K. Miatliuk, et al., Universal robot employment to mimic human writing, in: 20th Int. Carpathian Control Conf., ICCC, IEEE, 2019, pp. 1–5.
[7] G. Giorgi, A. Saracino, F. Martinelli, Using recurrent neural networks for continuous authentication through gait analysis, Pattern Recognit. Lett. 147 (2021) 157–163.
[8] Y.-F. Lin, et al., Forecasting fluctuations in the financial index using a recurrent neural network based on price features, IEEE Trans. Emerg. Top. Comput. Intell. 5 (5) (2020) 780–791.
[9] M. Woźniak, M. Wieczorek, J. Siłka, D. Połap, Body pose prediction based on motion sensor data and recurrent neural network, IEEE Trans. Ind. Inform. 17 (3) (2020) 2101–2111.
[10] M. Diaz, et al., Sequence-based dynamic handwriting analysis for Parkinson's disease detection with one-dimensional convolutions and BiGRUs, Expert Syst. Appl. 168 (2021) 114405.
[11] M. Seker, A.E. Tekden, E. Ugur, Deep effect trajectory prediction in robot manipulation, Robot. Auton. Syst. 119 (2019) 173–184.
[12] S. Bhavani, R. Bharathi, A multi-dimensional review on handwritten signature verification: strengths and gaps, Multimedia Tools Appl. 83 (1) (2024) 2853–2894.

[13] Z. Deng, L. Huang, C. Wang, Enhanced in-air signature verification via hand skeleton tracking to defeat robot-level replays, in: 39th Annual Computer Security Applications Conf., 2023, pp. 451–462.

[14] H. Lei, V. Govindaraju, A comparative study on the consistency of features in on-line signature verification, Pattern Recognit. Lett. 26 (15) (2005) 2483–2489.

[15] V.S. Nalwa, Automatic on-line signature verification, Proc. IEEE 85 (2) (1997) 215–239.

[16] M. Singhal, K. Shinghal, Secure deep multimodal biometric authentication using online signature and face features fusion, Multimedia Tools Appl. 83 (10) (2024) 30981–31000.

[17] H. Tan, L. He, Z.-C. Huang, H. Zhan, Online signature verification based on dynamic features from gene expression programming, Multimedia Tools Appl. 83 (5) (2024) 15195–15221.

[18] Craig, Introduction to Robotics : Mechanics & Control / John J. Craig, Addison-Wesley Pub. Co., Reading, Mass., 1986, Includes bibliographies and index..

[19] P. Corke, Representing position and orientation, in: Robotics, Vision and Control: Fundamental Algorithms in Python, 2023, pp. 23–86.

[20] P.M. Kebria, S. Al-wais, H. Abdi, S. Nahavandi, Kinematic and dynamic modelling of UR5 manipulator, in: IEEE Int. Conf. on Systems, Man, and Cybernetics, SMC, 2016, pp. 004229–004234.

[21] D. Xu, et al., Survey on multi-output learning, IEEE Trans. Neural Netw. Learn. Syst. 31 (7) (2019) 2409–2429.

[22] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.

[23] J. Ortega-Garcia, et al., MCYT baseline corpus: A bimodal biometric database, IEE Proc., Vis. Image Signal Process. 150 (6) (2003) 395–401.

[24] J. Galbally, et al., On-line signature recognition through the combination of real dynamic data and synthetically generated static data, Pattern Recognit. 48 (9) (2015) 2921–2934.

[25] A. Kholmatov, B. Yanikoglu, SUSIG: An on-line signature database, associated protocols and benchmark results, Pattern Anal. Appl. 12 (2009) 227–236.

[26] M.A. Ferrer, et al., Static and dynamic synthesis of Bengali and Devanagari signatures, IEEE Trans. Cybern. 48 (10) (2017) 2896–2907.

[27] A. Fischer, M. Diaz, R. Plamondon, M.A. Ferrer, Robust score normalization for DTW-based on-line signature verification, in: 13th Int. Conf. on Document Analysis and Recognition, ICDAR, 2015, pp. 241–245.

[28] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.

[29] R. Tolosana, et al., ICDAR 2021 competition on on-line signature verification, in: Document Analysis and Recognition–ICDAR 2021: 16th Int. Conf., Proceedings, Part IV, Springer, 2021, pp. 723–737.