



Research article

Spatiotemporal and kinematic characteristics augmentation using Dual-GAN for ankle instability detection

Xin Liu^{1,2,3}, Chen Zhao^{1,3}, Bin Zheng^{2,*}, Qinwei Guo^{4,*}, Yuanyuan Yu⁴, Dezheng Zhang^{1,3} and Aziguli Wulamu^{1,3}

¹ School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, China

² Surgical Simulation Research Laboratory, Department of Surgery, University of Alberta, Edmonton, Alberta, Canada

³ Beijing Key Laboratory of Knowledge Engineering for Materials Science, Beijing, China

⁴ Institute of Sports Medicine, Peking University Third Hospital, Beijing, China

* **Correspondence:** Email: bin.zheng@ualberta.ca, guoqinwei@vip.sina.com.

Abstract: Obtaining massive amounts of training data is often crucial for computer-assisted diagnosis using deep learning. Unfortunately, patient data is often small due to varied constraints. We develop a new approach to extract significant features from a small clinical gait analysis dataset to improve computer-assisted diagnosis of Chronic Ankle Instability (CAI) patients. In this paper, we present an approach for augmenting spatiotemporal and kinematic characteristics using the Dual Generative Adversarial Networks (Dual-GAN) to train a series of modified Long Short-Term Memory (LSTM) detection models making the training process more data-efficient. Namely, we use LSTM-, LSTM-Fully Convolutional Networks (FCN)-, and Convolutional LSTM-based detection models to identify the patients with CAI. The Dual-GAN enables the synthesized data to approximate the real data distribution visualized by the t-distributed Stochastic Neighbor Embedding (t-SNE) algorithm. Then we trained the proposed detection models using real data collected from a controlled laboratory study and mixed data from real and synthesized gait features. The detection models were tested in real data to validate the positive role in data augmentation as well as to demonstrate the capability and effectiveness of the modified LSTM algorithm for CAI detection using spatiotemporal and kinematic characteristics in walking. Dual-GAN generated efficient spatiotemporal and kinematic characteristics to augment the training set promoting the performance of CAI detection and the modified LSTM algorithm yielded an enhanced classification outcome to identify those CAI patients from a group of

control subjects based on gait analysis data than any previous reports.

Keywords: ankle instability; computer-assisted detection; gait features; Dual Generative Adversarial Networks (Dual-GAN); Long Short-Term Memory (LSTM)

1. Background

Chronic ankle instability (CAI) is common for patients with lateral collateral ligament injuries (LCL injuries) of the ankle [1,2]. About up to 40% of patients who have previously encountered lateral ankle sprain had a CAI problem [3]. These patients often require radical treatment for stabilizing their ankle, such as using orthoses, taping, or accepting surgical repairs on the lateral ligaments [4].

In current practice, physicians make diagnoses of CAI based on physical and medical examinations. So, medical imaging and gait kinetics on CAI is rapidly developing into a world research focus [5–8]. Deep learning algorithms are developed to improve the detection of radiographs accordingly [9]. To precisely manage CAI, physicians will need to perform gait analysis on patients. Lacking gait analysis impedes our understanding of patients' injuries and impacts on locomotion [10]. Many scientists argued that gait analysis is necessary for patients with foot and ankle ligament injuries, especially in those serious and frequent cases [11]. Gait analysis includes spatiotemporal and kinematic characteristics that can describe pathological changes in foot and ankle motion during walking [12–14]. This quantitative analysis will provide a solid foundation for deep learning to carry out an automatic, accurate, and immediate detection method for patients with ankle instability [15]. However, there are few studies on intelligent detection on gait analysis for patients with CAI now.

For this issue, our study aimed to augment gait features in CAI patients with highly representative, yet fully private and synthetic data for intelligent detection. To achieve this goal, we first employed the Dual Generative Adversarial Networks (Dual-GAN), a dataset augmentation method, to synthesize the significant spatiotemporal and kinematic characteristics measured in our previous study [16,17]. The Dual-GAN was trained by the injury and control group independently to learn the probability distribution of spatiotemporal and kinematic characteristics in walking. Then we used the t-distribution Stochastic Neighbor Embedding (t-SNE) algorithm to evaluate the relevance between the real and synthesized features visually. These real and synthesized data were fed to the Long Short-Term Memory (LSTM)-, LSTM- Fully Convolutional Networks (FCN)-, and Convolutional LSTM-based CAI detection models under different strategies for training. Finally, the confusion matrix was used to profile the diagnosis results using these three intelligent CAI detection models.

2. Related works

The Generative Adversarial Networks (GAN) proposed by Goodfellow et al. is an unsupervised deep learning model playing a Zero-Sum game between two competition networks (generator and discriminator). The generator attempts to synthesize samples to confuse the discriminator, while the discriminator is a binary classifier to detect the synthesized samples from real-synthesis mixed samples as much as possible. In current studies, several improved GANs were proposed aiming for the state-of-the-art application. The study of improved GANs mainly focused on optimizing the network structure and the objective function.

Several altered network structures have been presented in GAN to improve performance. Radford et al. [18] presented the Deep Convolutional GAN (DCGAN) using the transposed/general deep convolutional layers to replace the fully connected layers in the generator/discriminator to increase the performance. The DCGAN uses the fractional-stride convolution for pooling to improve their stability. Zhang et al. [19] synthesized photo-realistic images from text with Stack GAN (SGAN). The SGAN includes two conditional GANs decomposing a hard problem into sub-problems. The Stage-I GAN is used to synthesize low-resolution images. The Stage-II GAN takes Stage-I results as inputs and generated high-resolution data with realistic details. Yi et al. [20] proposed the Dual-GAN for mutual translation between two domains. Dual learning trains two opposite translators (source-to-target domain and target-to-source domain) simultaneously by minimizing the reconstruction loss via these coupled GANs. The coupled GANs form a nested feedback loop to reinforce the learning. Compared to the classic GAN model, the Dual-GAN model is easier to converge, as well as it has better stability and performance in data synthesizing. Moreover, several studies, such as Triple-GAN for hyperspectral classification on small training dataset [21], Star GAN for multi-domain image-to-image translation [22], Big-BiGAN progressing in image generation quality to improve representation learning performance [23], and Cycle-Dubur GAN to improve the image quality of chest cone-beam computed tomography images [24], etc., revealed that the altered networks make the GANs available for a wide range of image applications.

Furthermore, the improved objective function and penalty function are used to solve vanishing gradient and mode collapse in GAN. F-divergence is a general divergence measure between two given probability distributions including Kullback-Leibler divergence, Jensen-Shannon divergence, etc. Nowozin et al. showed a unified framework f-GAN and discussed the benefits for GAN using various f-divergence functions on model training [25]. F-GAN offers a powerful approach to measure complex distributions without factorizing assumptions and inspires the following researchers. The researchers try to design modified GANs to work well, fast, and concisely. Compared to the cross-entropy loss function, GAN with the least-squares loss function makes the synthesized samples closer to the real data. The least-squares loss function provides a smooth and non-saturating gradient in the discriminator forcing the synthesized samples toward the decision boundary [26]. Wasserstein GAN replaces the Jensen-Shannon divergence with Wasserstein distance to deal with the mode collapse and provide a meaningful loss function [27]. However, the training process and the convergence of Wasserstein GAN are more time-consuming than the original GAN. Given these issues, Ishaan Gulrajani et al. proposed gradient penalty instead of weight clipping in Wasserstein GAN and demonstrated that the Wasserstein GAN with gradient penalty (WGAN-GP) performs a stable training [28].

Various GANs were widely applied in image synthesizing for machine learning of small sample sets in recent studies, particularly in brain images, while rarely in time-series synthesizing. Bidirectional GAN, a novel end-to-end network, uses image contexts and latent vectors to optimize brain MR-to-PET synthesis for Alzheimer's Disease (AD) [29,30]. Multi-directional perception GAN (MP-GAN) can visualize the morphological features indicating the severity of AD by the capture capability of salient global features [31]. Tensor-train, High-order pooling, and Semi-supervised learning-based GAN (THS-GAN) is proposed to assess mild cognitive impairment and AD [32]. Compared to image synthesizing, time-series synthesizing will learn their distribution and dynamics synchronously in time-series embedding space [33,34].

3. Participants and data preparation

3.1. Participants

This controlled laboratory study was conducted at the Peking University Third Hospital. The methods used in this study were reviewed and approved by the Institutional Research Board at the Peking University Third Hospital. Each participant provided informed, written consent before entering the study.

Three patients (all male, mean age of 34 years, range: 32–37 years) diagnosed with CAI resulting from the Grade III LCL injuries of the ankle (severe, complete rupture of the ligament fibers) and ten paired control subjects (all male, mean age of 28 years, range: 24–32 years) were recruited. The control subjects from the student and staff of the hospital were no known lower extremity abnormalities, previous injuries, or surgeries. There was a total of 211 normal gait cycles and 30 injured gait cycles from the participants. The training dataset for Dual-GAN consisted of 132 normal gait cycles for six control subjects and 20 injured gait cycles from two patients with CAI. The rest of the data from another four control subjects and a patient was used to test the proposed model.

3.2. Data acquisition and pre-processing

As shown in Figure 1, the reflective markers were stuck on the shank and foot following the Heidelberg Foot Measurement Model (HFMM) [35]. A visual anthropometric model was built in the software Vicon Nexus 1.8.5 (Vicon Motion Systems Ltd., Oxford, UK). The Vicon MX Motion Capture System (Vicon Motion Systems Ltd., Oxford, UK) with eight cameras was used to track the markers within the capture volume and record their three-dimensional coordinate data at 100 Hz. The raw streaming data during walking were exported as .csv files for future analysis.



Figure 1. Tracking movements of subjects using a Vicon MX Motion Capture System. Markers are stuck on the shank and foot following the HFMM. LEP/MEP, lateral/medial epicondyle; TTU, tibial tuberosity; SH1/2, two points on the medial side of the tibia; LML/MML, lateral/medial malleolus; LCL/CCL/MCL, lateral/dorsal/medial calcaneus; NAV, navicular; PMT1/PMT5, proximal end of the first and fifth metatarsals; DMT1/2/5, distal end of the first/second/fifth metatarsals; HLX, hallux.

The raw data was filtered by a low-pass zero phase shift first-order Butterworth filter and segmented into gait cycles. The gait cycles were normalized to 100 sample points by linear interpolation for comparison among subjects.

3.3. Spatiotemporal and kinematic characteristics

In this study, we selected basic gait variables, velocity and micro-adjustment variables, and range of motion (ROM) variables as spatiotemporal and kinematic characteristics for CAI detection.

3.3.1. Basic gait variables

The basic gait variables reported in our previous study, including the percentage of first/second rocker phase in the gait cycle (%), the stride length (mm), and the duration (s/stride), were significantly different between the patients with LCL injuries of the ankle and control subjects [16].

3.3.2. Velocity and micro-adjustment variables

Five markers covering knee, ankle, and foot in HFMM were chosen for measuring velocity and micro-adjustment variables, including TTU, LML, CCL, DMT2, and HLX.

The results [16] revealed that in the gait cycle significant differences in velocity between patients with CAI of the ankle and control subjects involved the maximum velocity (V_{\max} , mm/10⁻²s) of the knee (TTU), ankle (LCL), and foot (CCL, DMT2, HLX); the percentage of time to maximum velocity (TV_{\max} , %) of the ankle (LML) and foot (CCL, DMT2, HLX); the minimum velocity (V_{\min} , mm/10⁻²s) of the TTU, LML, and DMT2; the percentage of time to minimum velocity (TV_{\min} , %) of the TTU.

In the 2nd rocker phase, the significant differences in micro-adjustment and velocity between the two groups included the number of valleys of the TTU, LML, and DMT2; the number of peaks of the TTU, LML, CCL, and DMT2; the mean velocity of the TTU, LML, DMT2, and HLX.

3.3.3. Range of motion variables

According to our research, eight ROM variables calculated in [17], including tibiotalar flexion, forefoot/ankle abduction, medial arch angle, lateral arch angle, subtalar rotation, forefoot/ankle supination, MT I-V angle, and lateral malleolus scale, all made a certain contribution to automatic detection for the ankle instability.

4. Data augmentation and CAI detection model

4.1. Dual Generative Adversarial Networks

4.1.1. Network architecture

The Dual-GAN is a closed-loop feedback system consisting of a pair of tasks. We can obtain detection results from unlabeled data, and then use the results to improve the machine learning models in dual tasks. The dual network structure translates in the cross-domain, which means that the network

learns the bidirectional mapping between the features in domain F and the results in domain R . The generator G_A maps feature $f (f \in F)$ to result $r (r \in R)$ and generator G_B maps result $r (r \in R)$ to features $f (f \in F)$. The discriminator D_A can recognize the difference between the generated result of G_A and the real result in domain R . The discriminator D_B can recognize the difference between the generated features of G_B and the real features in domain F .

As shown in Figure 2, the real feature f translates to the result $G_A(f, z)$ via G_A , and the discriminator D_A evaluates how appropriate the translation $G_A(f, z)$ is in domain R . Here, z and z' appeared below are both random noises. $G_A(f, z)$ then translates back to domain F using G_B , which outputs $G_B(G_A(f, z), z')$ as the reconstructed f . Similarly, the real result r is translated as $G_B(r, z')$ and then reconstructed as $G_A(G_B(r, z'), z)$. The discriminator D_A is trained with positive samples r and negative examples $G_A(f, z)$, whereas D_B takes f as positive samples and $G_B(r, z')$ as negative samples. The generator G_A and G_B emulate “fake” outputs to blind the correlation between discriminator D_A and D_B and minimize the reconstruction loss $\|G_A(G_B(r, z'), z) - r\|$ and $\|G_B(G_A(f, z), z') - f\|$ for optimizing the Dual-GAN.

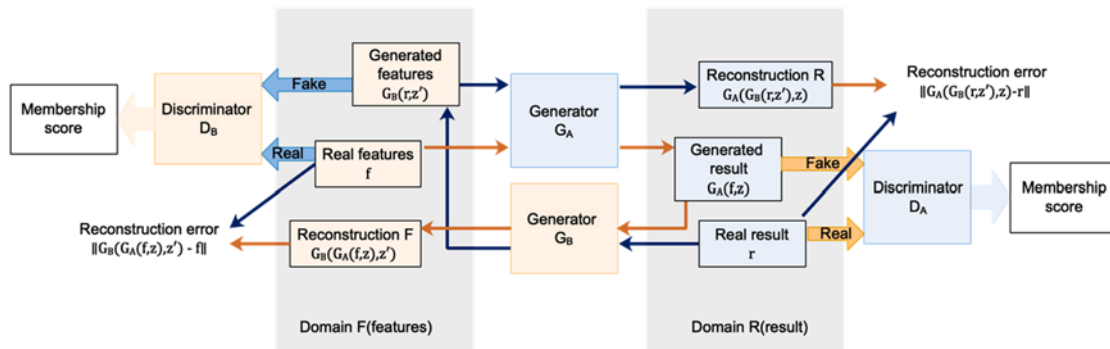


Figure 2. Dual-GAN framework architecture and data flow chart of Dual-GAN for cross-domain feature-to-result translation. The model is trained for the injury group and control group independently.

We employ the gradient penalty in WGAN-GP replacing the weight clipping to insure gradient stability in this study. The loss functions of D_A and D_B are defined as:

$$L_A^d = E_{\tilde{x} \sim \mathbb{P}_g} [D_A(\tilde{x})] - E_{x \sim \mathbb{P}_r} [D_A(x)] + \lambda E_{\tilde{x} \sim \mathbb{P}_{\tilde{x}}} [(\|\nabla_{\tilde{x}} D_A(\tilde{x})\|_2 - 1)^2] \quad (1)$$

$$L_B^d = E_{\tilde{x} \sim \mathbb{P}_g} [D_B(\tilde{x})] - E_{x \sim \mathbb{P}_f} [D_B(x)] + \lambda E_{\tilde{x} \sim \mathbb{P}_{\tilde{x}}} [(\|\nabla_{\tilde{x}} D_B(\tilde{x})\|_2 - 1)^2] \quad (2)$$

Here, loss function L_A^d is the objective of D_A , and loss function L_B^d is the objective of D_B . λ is a hyperparameter, and it is set to 10 in this study. We also define $\mathbb{P}_{\tilde{x}}$ as sampling uniformly along straight lines between pairs of points sampled from the data distribution (\mathbb{P}_f or \mathbb{P}_r) and the corresponding generator distribution \mathbb{P}_g .

Generator G_A and G_B use the same loss function:

$$l^g(f, r) = \lambda_F \|f - G_B(G_A(f, z), z')\| + \lambda_R \|r - G_A(G_B(r, z'), z)\| - D_B(G_B(r, z')) - D_A(G_A(f, z)) \quad (3)$$

where $f \in F$; $r \in R$; λ_F , λ_R are two constant parameters within $[100.0, 1000.0]$. Here, the

Manhattan distance is used to measure the recovery error in order to force the reconstruction samples to obey the domain distribution.

4.1.2. Network configuration

Dual-GAN consists of two generators and two discriminators. The discriminators have the same structure while the generators are different. The architecture of generators G_A and G_B is shown in Figure 3.

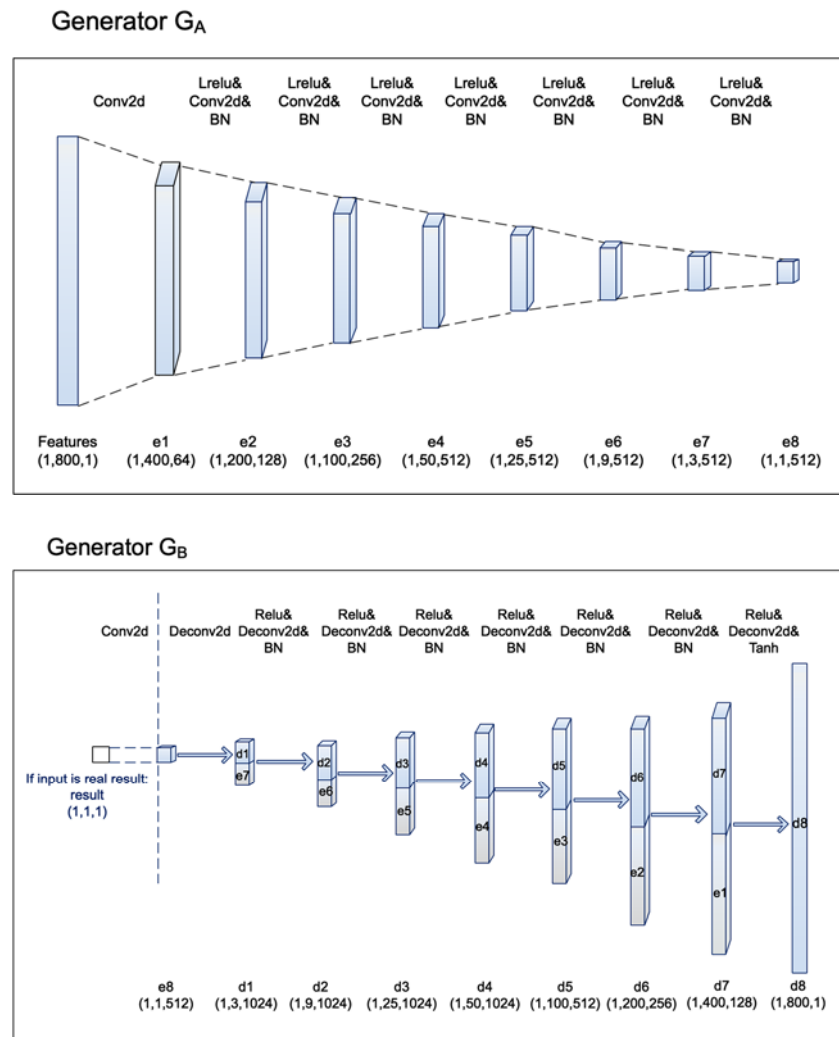


Figure 3. The architecture of generators G_A and G_B .

Generator G_A adopts the FCN. The input of G_A is a feature vector, and the output is diagnosis result vector for CAI. Specifically, G_A is built with a convolutional layer with 5×5 kernel-size and 2 stride-size, 4 convolutional layers with 3×3 kernel-size and 2 stride-size, as well as 3 convolutional layers with 3×3 kernel-size and 3 stride-size. Each layer includes the leaky rectified linear unit (LReLU) activation function and batch normalization (BN) except for the first layer.

Generator G_B also uses the FCN. But the input of G_B is diagnosis result vector for CAI, and the

output is a feature vector. When the input is a real result vector, a 1×1 convolutional layer is set for matching the dimension between the real result vector and G_B . Generator G_B is built with a convolutional layer with 1×1 kernel-size, 3 deconvolutional layers with 3×3 kernel-size and 3 stride-size, 4 deconvolutional layers with 3×3 kernel-size and 3 stride-size, as well as a convolutional layer with 5×5 kernel-size and 2 stride-size. Each layer included the rectified linear unit (ReLU) activation function and BN except for the last layer. The last layer configures a TanHyperbolic (tanh) activation function to output the generated feature.

The generators G_A and G_B make up a U-shaped generator net with skip connections for learning both high- and low-level features. So, we can establish contact between the gait characteristics and CAI detection results through these skip connections.

The architecture of discriminators D_A and D_B is shown in Figure 4. When the input is a real result vector, a 1×1 convolutional layer is set for matching the dimension between the real and generated result vectors. Discriminators D_A and D_B consist of a convolutional layer with 5×5 kernel-size and 2 stride-size, 4 convolutional layers with 3×3 kernel-size and 2 stride-size, as well as a fully connected layer. The LReLU activation function follows each convolutional layer, and BN is configured at layers 2–5.

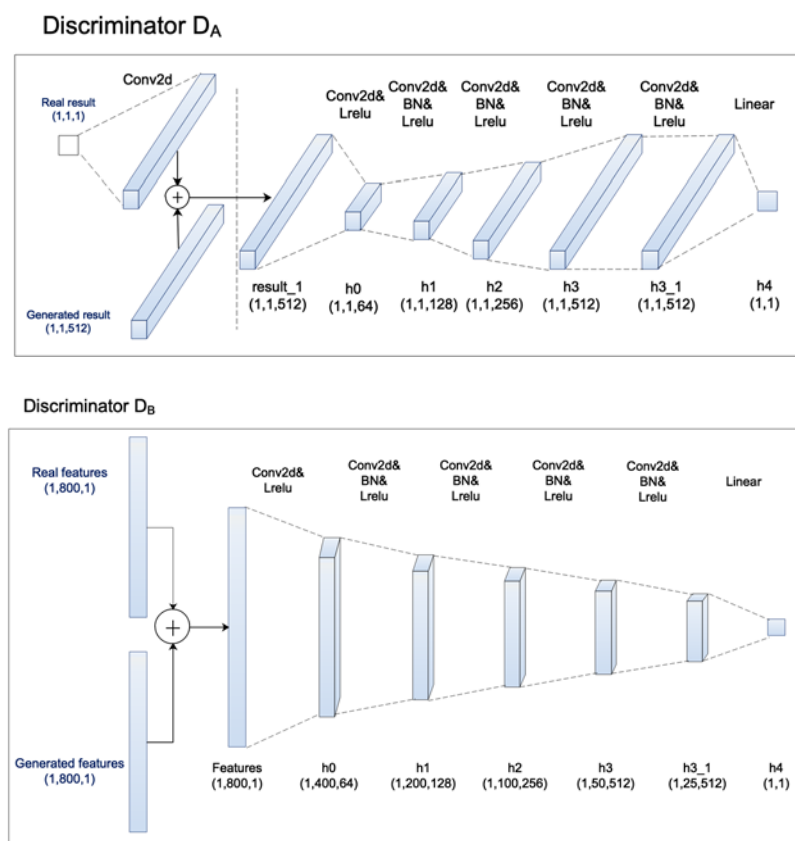


Figure 4. The architecture of discriminators D_A and D_B .

4.1.3. Model training

We used mini-batch Stochastic Gradient Descent and applied the Root Mean Square Propagation

(RMSProp) solver to optimize this Dual-GAN model. The batch size was set to 8 and the dropout rate was 0.3. The learning rate α and the attenuation rate β were set to 0.001 and 0.9, respectively. We adopted Manhattan distance to measure the recovery error between synthesized samples and real data. The gradient penalty replaced weight clipping in the loss function for high-quality data generation. The Dual-GAN training procedure is shown in Figure 5.

-
- Algorithm 1:** Dual-GANs training procedure. We use default values of $\lambda = 10$, $\alpha = 0.001$, $\beta = 0.9$.
-
- 1 Require: Feature set F, Result set R. A GAN A with generator parameters θ_r and discriminator with parameters ω_r . A GAN B with generator parameters θ_f and discriminator with parameters ω_f . The gradient penalty coefficient λ , the number of critic iterations per generator iteration n_{critic} , the batch size m , RMSProp hyperparameters α , β . Randomly initialize θ_i , ω_i , $i \in \{f, r\}$.
 - 2 repeat
 - 3 samples $\{f^{(k)}\}_{k=1}^m \subseteq F$, $\{r^{(k)}\}_{k=1}^m \subseteq R$.
 sample a random number $\epsilon \sim F[0, 1]$.
 $\tilde{x}_r \leftarrow G_A(\{f^{(k)}\}_{k=1}^m)$.
 $\tilde{x}_f \leftarrow G_B(\{r^{(k)}\}_{k=1}^m)$.
 $\bar{x}_r \leftarrow \epsilon x_r + (1 - \epsilon)\tilde{x}_r$
 $\bar{x}_f \leftarrow \epsilon x_f + (1 - \epsilon)\tilde{x}_f$
 $L_A^d \leftarrow D_A(\bar{x}_r) - D_A(\{r^{(k)}\}_{k=1}^m) + \lambda(\|\nabla_{\bar{x}_r} D_A(\bar{x}_r)\|_2 - 1)^2$
 $L_B^d \leftarrow D_B(\bar{x}_f) - D_B(\{f^{(k)}\}_{k=1}^m) + \lambda(\|\nabla_{\bar{x}_f} D_B(\bar{x}_f)\|_2 - 1)^2$
 $\omega_r \leftarrow RMSProp(\nabla_{\omega_r} \frac{1}{m} L_A^d, \omega_r, \alpha, \beta)$
 $\omega_f \leftarrow RMSProp(\nabla_{\omega_f} \frac{1}{m} L_B^d, \omega_f, \alpha, \beta)$
 samples $\{f^{(k)}\}_{k=1}^m \subseteq F$, $\{r^{(k)}\}_{k=1}^m \subseteq R$.
 $\theta_r \leftarrow RMSProp(\nabla_{\theta_r} \frac{-1}{m} D_A(G_A(\{f^{(k)}\}_{k=1}^m)), \theta_r, \alpha, \beta)$
 $\theta_f \leftarrow RMSProp(\nabla_{\theta_f} \frac{-1}{m} D_B(G_B(\{r^{(k)}\}_{k=1}^m)), \theta_f, \alpha, \beta)$
 - 4 until convergence;
-

Figure 5. Dual-GAN training procedure.

We randomly selected a group of real characteristics and synthesized samples using Dual-GAN for visualization (Figure 6). The parameters of Dual-GAN are initialized at epoch 0. After 50 epochs, the network begins to learn the general trend of spatiotemporal and kinematic characteristics in the gait cycle. After 100 epochs, the generator can synthesize an unsmooth curve closing to real characteristics. While training after 150–200 epochs, the synthesized sample resembles the real data, but it seems relatively oscillatory. After 250 epochs, the Dual-GAN is convergence, and the synthesized sample coordinate with the real gait characteristics. In the following section, we will feed the synthesized samples to the detection model and verify the synthesized samples using the t-SNE algorithm.

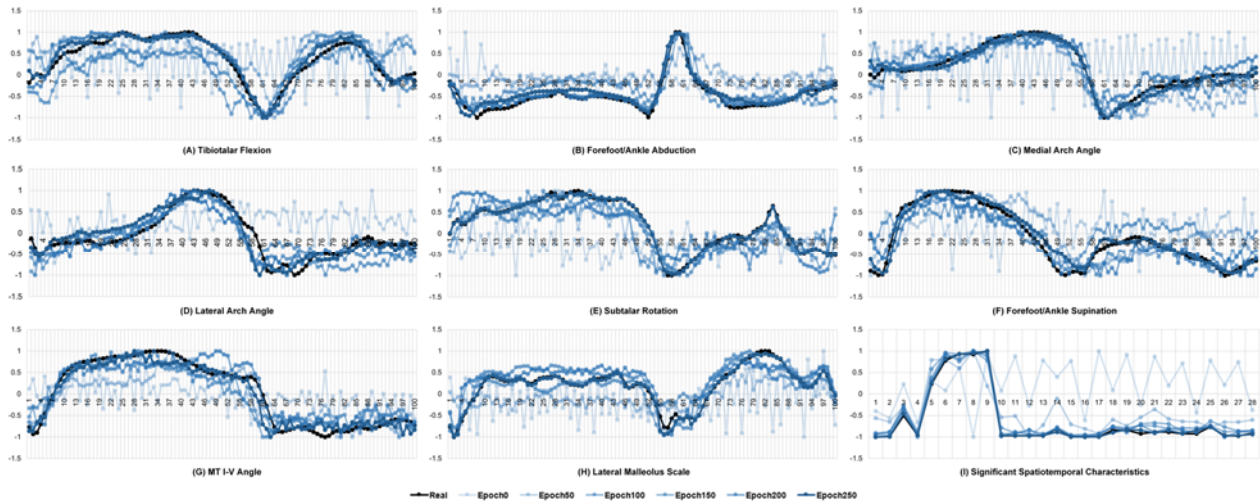


Figure 6. A set of synthesized spatiotemporal and kinematic characteristics for the patients group selected randomly from the Dual-GAN results (at 0, 50, 100, 150, 200, and 250 epochs), and a set of real characteristics from a patient with CAI.

4.2. CAI detection

The network architecture based on LSTM learns the long-term sequence characteristics of spatiotemporal and kinematic variables well. We adopted three LSTM-based deep learning models, including LSTM, LSTM-FCN, and Convolutional LSTM to detection CAI (Figure 7) [36–39]. The synthesized and real data were mixed for training the detection model.

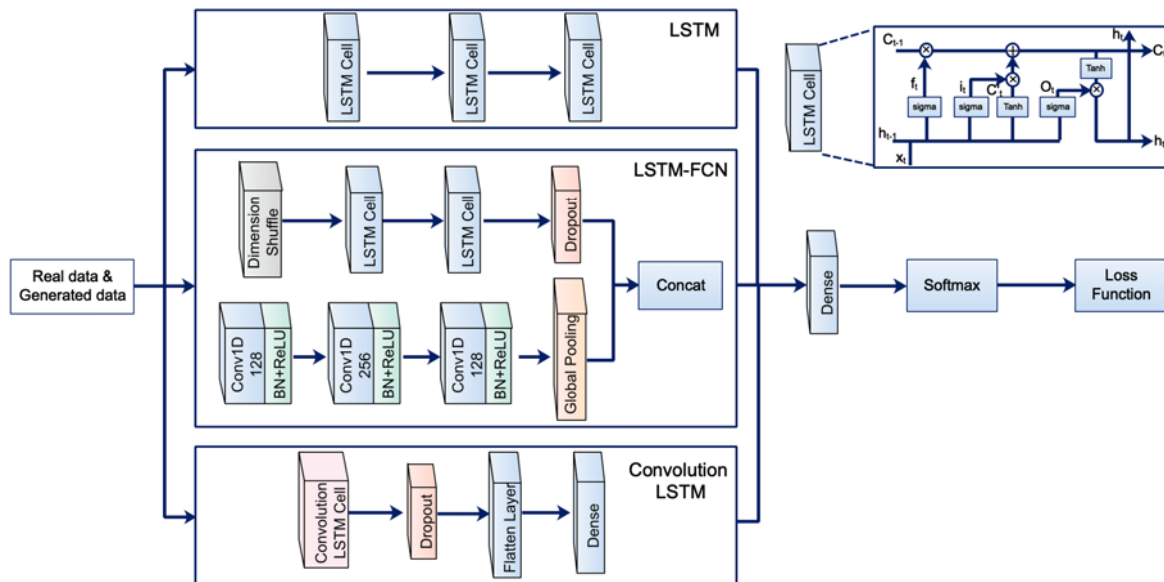


Figure 7. The net structure of three CAI detection models.

4.2.1. LSTM-based detection model

LSTM can solve the problem of gradient disappearance, especially in the long-distance dependence task. We used LSTM to CAI detection in this study. As shown in Figure 7, the LSTM cell is composed of an input gate, an output gate, and a forget gate. The cell traces the dependency between features in the input sequence. The input gate selectively retains the information from the previous time step. The forget gate gives up some input information from the previous node. The output gate controls the current cell state output to the next cell. The training procedure of the LSTM-based detection model is shown in Figure 8. The activation function of LSTM is the logistic sigmoid function. The input gate i , forget gate f , and output gate O are:

$$i_t = \sigma(W_i \cdot [h_{t-1}, X_t] + b_i) \quad (4)$$

$$f_t = \sigma(W_f \cdot [h_{t-1}, X_t] + b_f) \quad (5)$$

$$C_t = f_t C_{t-1} + i_t \tanh(W_c \cdot [h_{t-1}, X_t] + b_c) \quad (6)$$

$$O_t = \sigma(W_o \cdot [h_{t-1}, X_t] + b_o) \quad (7)$$

$$h_t = O_t \tanh(C_t) \quad (8)$$

where X_t is the input of current time; h_{t-1} is the output in the previous time; C_{t-1} is the cell status in the previous time; W is the connection weight; b is bias item; σ is a sigmoid activation function; \tanh is a hyperbolic tangent function.

Algorithm 2: LSTM-based detection model training procedure. We use default values of $\alpha = 0.001$, $\beta = 0.9$.

- 1 Require: LSTM with parameters θ_{lstm} , the batch size bs , the epoch $nEpochs$, the label of training samples y , the probability of predicting samples p , real samples f , reconstructed samples \tilde{x}_f , feature set F , RMSProp hyperparameters α , β .
 - 2 **for** $e = 1$ to $nEpochs$ **do**
 - 3 samples $\{\tilde{x}_f^{(k)}\}_{k=1}^{bs} \subseteq F$, $\{f^{(k)}\}_{k=1}^{bs} \subseteq F$, $\{y^{(k)}\}_{k=1}^{bs}$
 $\{p^{(k)}\}_{k=1}^{bs} \leftarrow LSTM(samples, \theta_{lstm}, \alpha, \beta)$
 $Loss = BinaryCrossEntropy(\{y^{(k)}\}_{k=1}^{bs}, \{p^{(k)}\}_{k=1}^{bs})$
 $\theta_{lstm} \leftarrow RMSProp(\nabla_{\theta_{lstm}} \frac{-1}{bs} \sum_{k=1}^{bs} Loss, \theta_{lstm}, \alpha, \beta)$
 - 4 **endfor**
-

Figure 8. Training procedure of LSTM-based detection model.

4.2.2. LSTM-FCN-based detection model

As shown in Figure 7, the LSTM-FCN model in this study includes a fully convolutional block and a LSTM block. The fully convolutional block is comprised of three stacked temporal convolutional

blocks and a global average pooling layer. The LSTM block consists of a dimension shuffle layer, two general LSTM sub-modules, and a dropout layer. The dimension shuffle layer can reduce the training time improving the model's efficiency. The training procedure of the LSTM-FCN-based detection model is shown in Figure 9.

Algorithm 3: LSTM-FCN-based detection model training procedure. We use default values of $\alpha = 0.001$, $\beta = 0.9$.

- 1 Require: LSTM-FCN with parameters $\theta_{lstm-FCN}$, the batch size bs , the epoch $nEpochs$, the label of training samples y , the probability of predicting samples p , real samples f , reconstructed samples \tilde{x}_f , feature set F , RMSProp hyperparameters α , β .
- 2 **for** $e = 1$ to $nEpochs$ **do**
- 3 samples $\{\tilde{x}_f^{(k)}\}_{k=1}^{bs} \subseteq F$, $\{f^{(k)}\}_{k=1}^{bs} \subseteq F$, $\{y^{(k)}\}_{k=1}^{bs}$
 $\{p^{(k)}\}_{k=1}^{bs} \leftarrow LSTM-FCN(samples, \theta_{lstm-FCN}, \alpha, \beta)$
 $Loss = BinaryCrossEntropy(\{y^{(k)}\}_{k=1}^{bs}, \{p^{(k)}\}_{k=1}^{bs})$
 $\theta_{lstm-FCN} \leftarrow RMSProp(\nabla_{\theta_{lstm-FCN}} \frac{-1}{bs} \sum_{k=1}^{bs} Loss, \theta_{lstm-FCN}, \alpha, \beta)$
- 4 **endfor**

Figure 9. Training procedure of LSTM-FCN-based detection model.

4.2.3. Convolutional LSTM-based detection model

The Convolutional LSTM (Figure 7) not only possesses the capacity of temporal modeling from LSTM but also can describe spatial local features. This is because the Convolutional LSTM model applies convolution structure in both input-to-state and state-to-state transitions to extract spatial features. The ‘‘peephole connection’’ in each gate is employed to supervise the cell state. The training procedure of the Convolutional LSTM-based detection model is shown in Figure 10.

Algorithm 4: Convolutional LSTM-based detection model training procedure. We use default values of $\alpha = 0.001$, $\beta = 0.9$.

- 1 Require: Convolutional LSTM with parameters $\theta_{conv-lstm}$, the batch size bs , the epoch $nEpochs$, the label of training samples y , the probability of predicting samples p , real samples f , reconstructed samples \tilde{x}_f , feature set F , RMSProp hyperparameters α , β .
- 2 **for** $e = 1$ to $nEpochs$ **do**
- 3 samples $\{\tilde{x}_f^{(k)}\}_{k=1}^{bs} \subseteq F$, $\{f^{(k)}\}_{k=1}^{bs} \subseteq F$, $\{y^{(k)}\}_{k=1}^{bs}$
 $\{p^{(k)}\}_{k=1}^{bs} \leftarrow conv-LSTM(samples, \theta_{conv-lstm}, \alpha, \beta)$
 $Loss = BinaryCrossEntropy(\{y^{(k)}\}_{k=1}^{bs}, \{p^{(k)}\}_{k=1}^{bs})$
 $\theta_{conv-lstm} \leftarrow RMSProp(\nabla_{\theta_{conv-lstm}} \frac{-1}{bs} \sum_{k=1}^{bs} Loss, \theta_{conv-lstm}, \alpha, \beta)$
- 4 **endfor**

Figure 10. Training procedure of Convolutional LSTM-based detection model.

5. Experiments and results

5.1. Data feeding strategies

We designed three data feeding strategies (DFSs) for training the CAI detection control model and comparing their outcomes. The first data feeding strategy (DFS1) only included significant spatiotemporal characteristics (basic gait variables, velocity variables, and micro-adjustment variables). The second data feeding strategy (DFS2) only included kinematic characteristics (ROM variables). The third data feeding strategy (DFS3) included both spatiotemporal and kinematic characteristics.

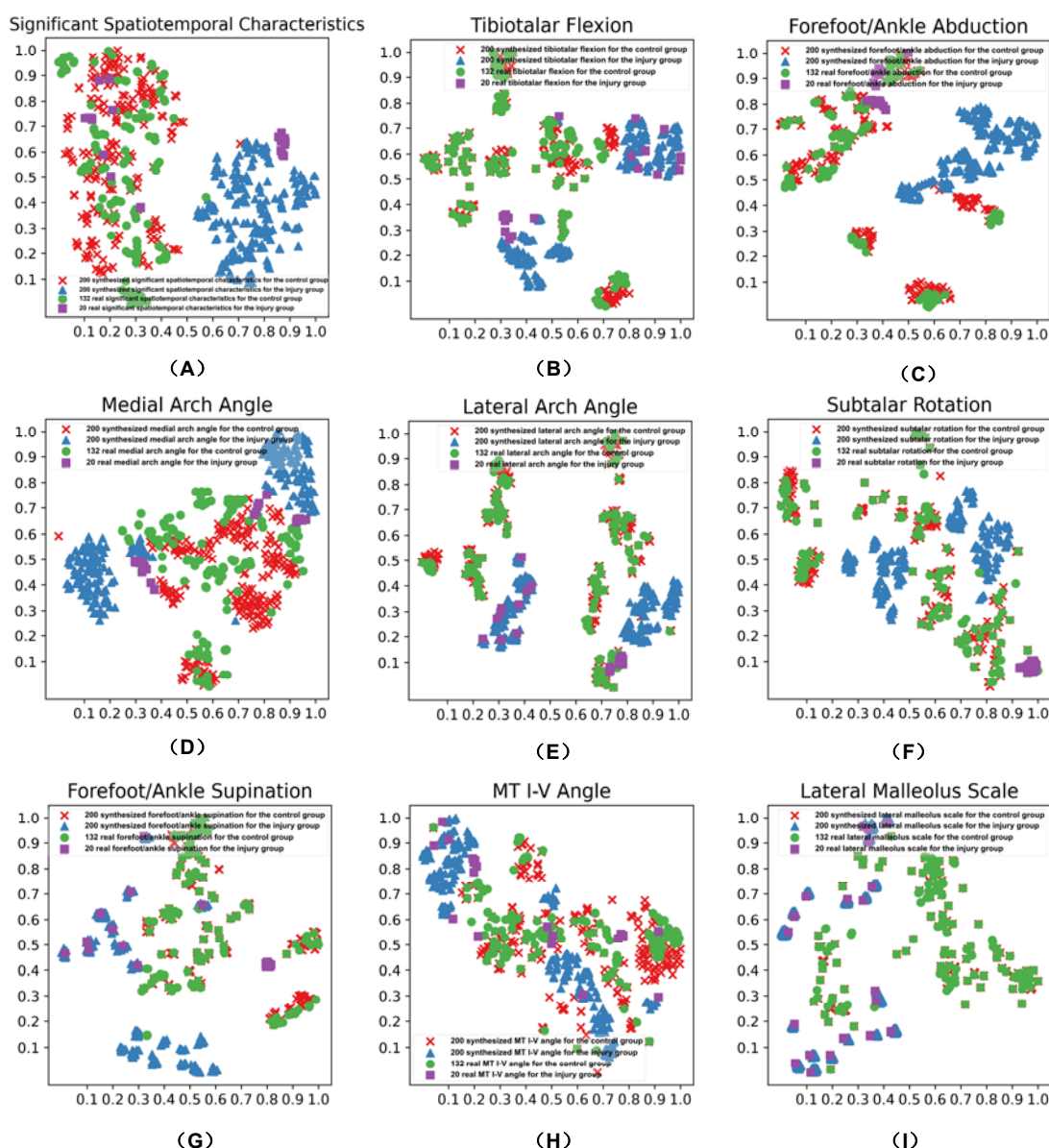


Figure 11. Verifying synthesized spatiotemporal and kinematic characteristics by t-SNE-based visualization.

Moreover, we created three sub-DFSs for comparing the effects of the synthesized data on the detection model. The first sub-DFS (DFS*-1), as same as the training set for the Dual-GAN, only included real characteristics. The second sub-DFS (DFS*-2) included the real characteristics plus 200 synthesized data for each group. The third sub-DFS (DFS*-3) included the real characteristics plus 1000 synthesized data for each group.

5.2. *t*-SNE algorithm-based synthesized data verification

The t-SNE algorithm is a nonlinear descending dimension method for visualizing the high-dimensional spatiotemporal and kinematic characteristics [40]. What t-SNE does is present a way to project high-dimensional data into a low-dimensional space [41]. The data point on the 2-dimensional plane is attracted to data it is semblable in high-dimensional space and repelled by data it is unlike [42]. Figure 11 shows the visualization of 132 real control data (●), 20 real injury data (■), 200 randomly selected synthesized control data (×), and 200 randomly selected synthesized injury data (▲) reported by t-SNE. Here, the perplexity was 30 to keep a balance between local and global features.

The synthesized data overlapped with the same classification of real data by learning the distribution from the control group and injury group, respectively. The distribution between real and synthesized data for tibiotalar flexion (B), lateral arch angle (E), forefoot/ankle supination (G), MT I-V angle (H), and lateral malleolus scale (I) were more similar than the others. Compared to patients, the synthesized data of control subjects (×) were more consistent with the real (●) due to their routine gait rhythm and pattern. The synthesized data is generally in good shape for training the CAI detection model.

5.3. Ankle instability detection

The CAI detection is a binary classification including CAI and non-CAI. Here classification metrics such as accuracy, precision, recall (sensitivity), and f1-score (balanced score) are selected to evaluate the performance of the CAI detection modes. the relevant components including *TP* (the number of correct predicting non-CAI samples), *TN* (the number of correct predicting CAI), *FP* (the number of false predicting non-CAI), and *FN* (the number of false predicting CAI). The definitions of the classification metrics are explained below.

Accuracy is the proportion of all correctly predicted samples to total samples for measuring the performance of a detection model.

$$ACC = \frac{TP+TN}{TP+TN+FP+FN} \quad (9)$$

Precision is the proportion of all correctly predicted non-CAI samples to all predicted non-CAI samples.

$$Precision = \frac{TP}{TP+FP} \quad (10)$$

Recall is the proportion of all correctly predicted non-CAI samples to all real non-CAI samples.

$$Recall = \frac{TP}{TP+FN} \quad (11)$$

The f1-score is the harmonic mean of precision and recall measuring their difference.

$$f1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (12)$$

5.3.1. LSTM-based detection

The LSTM-based detection model in this study consisted of an input layer, two hidden layers, and an output layer. For each input/hidden layer, there were 32 neurons. The output layer was a Softmax classifier with two nodes to output the result of detection (CAI/non-CAI). We adopted a binary cross-entropy loss function after the output layer to estimate the consistency between the outcome and true value. Here the RMSProp solver was used to restrain the swing amplitude and speed up the convergence during the gradient descent. The iteration was set to 300 and the batch size was 32 in model training. The accuracy, precision, recall, and f1-score for CAI detection using this LSTM model are presented in Table 1.

Table 1. Accuracy, precision, recall and f1-score for CAI detection using LSTM model.

DFS	Measure	DFS1	DFS2	DFS3
DFS*-1	accuracy (%)	74.16	88.76	88.76
	precision	0.94	0.89	0.89
	recall	0.76	1.00	1.00
	f1-score	0.84	0.94	0.94
DFS*-2	accuracy (%)	86.52	94.38	94.38
	precision	1.00	0.94	0.94
	recall	0.85	1.00	1.00
	f1-score	0.92	0.97	0.97
DFS*-3	accuracy (%)	98.88	100	95.51
	precision	1.00	1.00	0.95
	recall	0.99	1.00	1.00
	f1-score	0.99	1.00	0.98

5.3.2. LSTM-FCN-based detection

We tried to replace the LSTM module with LSTM-FCN in the CAI detection model for improving the classification performance. In this model, the LSTM and FCN sub-module perceived the same input in different views. The LSTM sub-model included a dimension shuffle layer and two LSTM cell layers followed by a dropout layer. The shuffle layer for receiving the input was a multivariate time series with a single time step. For each LSTM cell layer, there were 32 neurons. The FCN sub-module consisted of three stacked convolutional blocks with filter sizes of 128, 256, and 128 respectively and a global pooling layer. Each convolutional block comprised a temporal convolutional layer accompanied by BN (momentum was 0.99, epsilon was 0.001) and a ReLU activation function. The output of the dropout and global pooling layer was concatenated and input into the Softmax classifier. Here the iteration was set to 50 and the batch size was 32 in model training. The accuracy, precision, recall, and f1-score for CAI detection using this LSTM model are presented in Table 2.

Table 2. Accuracy, precision, recall and f1-score for CAI detection using LSTM-FCN model.

DFS	Measure	DFS1	DFS2	DFS3
DFS*-1	accuracy (%)	83.15	88.76	88.76
	precision	0.89	0.89	0.89
	recall	0.92	1.00	1.00
	f1-score	0.91	0.94	0.94
DFS*-2	accuracy (%)	87.64	96.63	94.38
	precision	0.89	0.96	0.94
	recall	0.99	1.00	1.00
	f1-score	0.93	0.98	0.97
DFS*-3	accuracy (%)	94.38	94.38	95.51
	precision	0.97	0.94	0.95
	recall	0.96	1.00	1.00
	f1-score	0.97	0.97	0.98

5.3.3. Convolutional LSTM-based detection

Considering the temporal-spatial characteristics of gait variables, we further adopted the Convolutional LSTM-based model detecting the CAI. The convolutional layer with filter size 64 and kernel size 1×3 , and was accompanied by a ReLU activation function. The convolutional layer was followed by a dropout layer (dropout rate was 0.5) and a flattened layer for inputting a dimensional array into the Softmax classifier. Here the iteration was set to 25 and the batch size was 64 in model training. The accuracy, precision, recall, and f1-score for CAI detection using this LSTM model are presented in Table 3.

Table 3. Accuracy, precision, recall and f1-score for CAI detection using Convolutional LSTM model.

DFS	Measure	DFS1	DFS2	DFS3
DFS*-1	accuracy (%)	88.76	88.76	88.76
	precision	0.89	0.89	0.89
	recall	1.00	1.00	1.00
	f1-score	0.94	0.94	0.94
DFS*-2	accuracy (%)	87.64	96.63	98.88
	precision	1.00	1.00	1.00
	recall	0.86	0.96	0.99
	f1-score	0.93	0.98	0.99
DFS*-3	accuracy (%)	94.38	100	100
	precision	0.99	1.00	1.00
	recall	0.95	1.00	1.00
	f1-score	0.97	1.00	1.00

6. Discussion

Based on the results of spatiotemporal and kinematic characteristics augmentation using Dual-

GAN, we employed three LSTM-based models (the LSTM-based model, LSTM-FCN-based model, and Convolutional LSTM-based model) in the study to compare their performance and reveal the effect of synthesizing data on CAI detection. The detection outputs by LSTM, LSTM-FCN, and Convolutional LSTM models under three DFSs are shown in Tables 1–3 and the Receiver Operating Characteristic (ROC) curves for the CAI and non-CAI classification are shown in Figure 12.

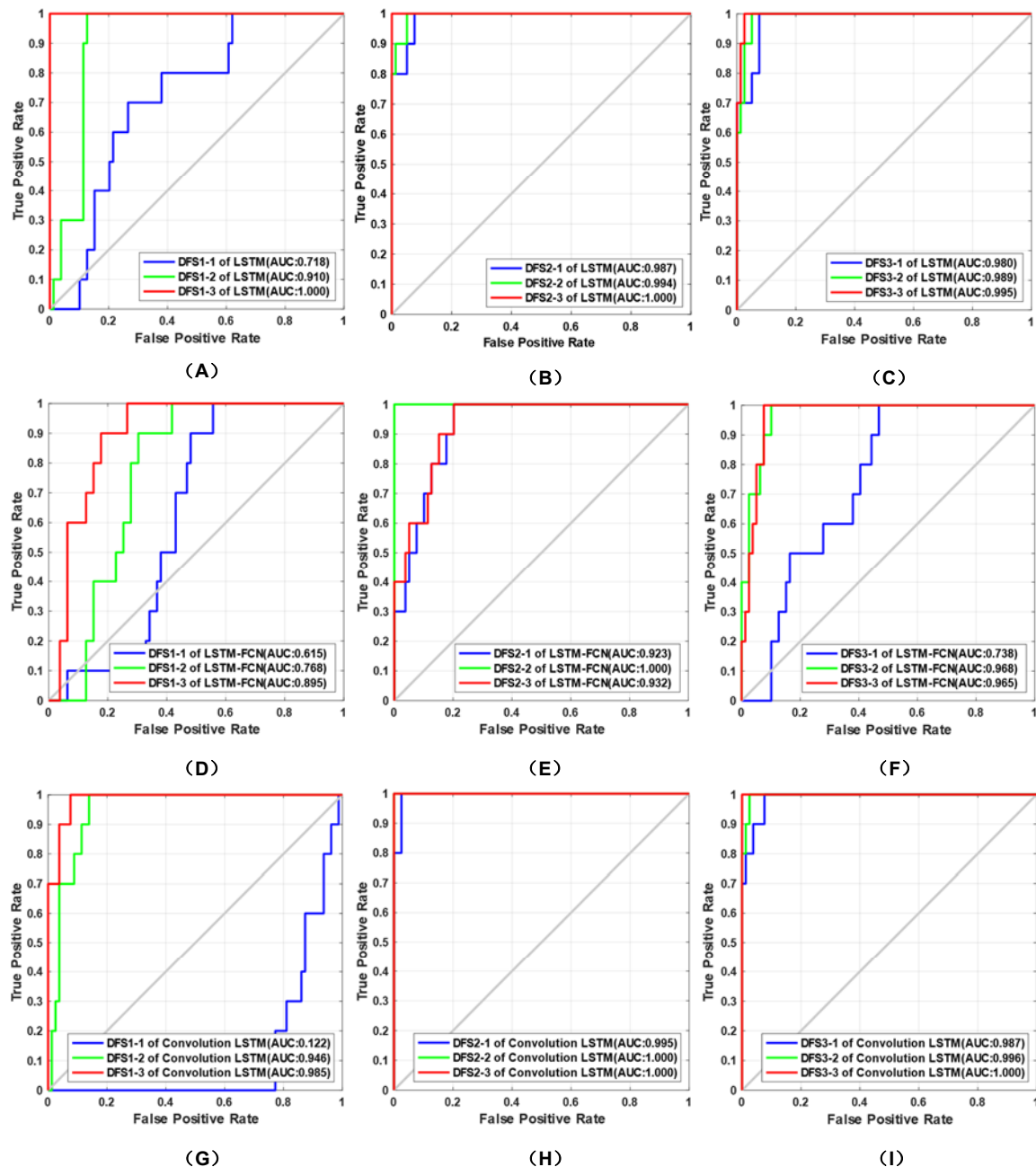


Figure 12. ROC curves for CAI and non-CAI classification using different DFSs. ROC curves for either or both spatiotemporal and kinematic characteristics using the same detection model (in rows). ROC curves for the same characteristics using LSTM-based, LSTM-FCN-based, and Convolutional LSTM-based detection model (in columns).

We took a step to optimize the feeding by pre-selecting crucial features for intelligent detection in our previous study [16,17]. The spatiotemporal and kinematic characteristics in gait were both effective to predict CAI. However, the detection performance using the model trained by only real spatiotemporal characteristics (DFS1) was unsatisfactory (Area Under Curve (AUC): 0.122–0.718). When we used real kinematic characteristics (DFS2) for training, the outcome was improved (AUC for DFS2: 0.923–0.995). When we added real kinematic characteristics to real spatiotemporal characteristics (DFS3) for training, the outcome was also good (AUC for DFS3: 0.738–0.987). In general, the training set of only the real data did not yield adequate classification outcomes. The accuracy of DFS2 and DFS3 were all 88.76% for LSTM, LSTM-FCN, and Convolutional LSTM-based detection models. The accuracy, precision, recall, and f1-score were moderate.

So, we further included 200 and 1000 synthesized characteristics in the training set, and the detection accuracy, precision, recall, and f1-score improved dramatically. Adding 200 synthesized data for training, the outcome of the detection model was improved. When we included 1000 synthesized data, the detection accuracy, precision, recall, and f1-score improved dramatically for either or both spatiotemporal and kinematic characteristics. The volume of training samples is crucial for the models' detection performance. Spatiotemporal and kinematic characteristics augmentation using Dual-GANs can improve the diagnostic accuracy of the CAI detection models presented in this study.

When we added 200 synthesized data to the real data set training the LSTM-based CAI detection model (Table 1), the accuracy was improved (from 74.16% to 86.62% for DFS1; from 88.76% to 94.38% for DFS2 and DFS3). After adding 1000 synthesized data, the accuracy for the three DFSs was more than 95% (DFS1: 98.88%, DFS2: 100%, DFS3: 95.51%), and the precision, recall, and f1-score also look good.

For LSTM-FCN-based CAI detection model (Table 2), the accuracy was improved (from 83.15% to 87.64% for DFS1; from 88.76% to 96.63% for DFS2; from 88.76% to 94.38% for DFS3) because of adding 200 synthesized data (DFS*-2). Adding 1000 synthesized data (DFS*-3) enhanced the detection accuracy of DFS1 (from 87.64% to 94.38%) and DFS3 (from 94.38% to 95.51%), but there was no positive effect on DFS2.

The results in Table 3 reveal that the performance of the Convolutional LSTM-based classifier is better than LSTM and LSTM-FCN-based CAI detection model. The Convolutional LSTM-based detection model showed better performance for spatiotemporal and kinematic characteristics. Adding more synthesized data helped the Convolutional LSTM-based classifier enhance the detection performance. when we added 1000 synthesized data to the training set, the accuracy of DFS1, DFS2, and DFS3 was up to 94.38, 100, and 100% respectively. Their precision, recall, and score were all superior. The detection results were good, but several limitations of the current study need to be noted. First, the high-quality gait information from the optical motion capture system helped us find several significant spatiotemporal and kinematic characteristics of CAI enhancing the detection efficiency, but the data collection is time-consuming. At present, wearable devices, such as smartphones, wearable biofeedback sensors, and sensing fabrics, are widely applied to gait monitoring [43,44]. Unlike laboratory-based motion tracking, wearable devices are convenient to record daily gait data. Based on our current results, we will explore effective gait features from the data recording by wearable devices. It is possible to get more practical applications for our studies, such as multi-sensor-based human activity recognition, bipedal walking robot, exoskeleton, et al. [45–48]. Second, this paper focused on the performance of LSTM-based models for ankle instability detection. We will explore more CAI detection models based on characteristics augmenting and combine them to improve the detection effectiveness in our further research.

7. Conclusions

In this paper, we first present an approach for augmenting spatiotemporal and kinematic characteristics using the Dual-GAN to train a series of modified LSTM detection models to make the training process more data-efficient. The Dual-GAN enables the synthesized data to approximate the real data distribution visualized by the t-SNE algorithm.

Then we use LSTM-, LSTM-FCN-, and Convolutional LSTM-based detection models training by the real data collected from the controlled laboratory study and mixed data from real and synthesized gait features respectively to identify the patients with CAI. The detection models were tested in real data to validate the positive role in data augmentation as well as to demonstrate the capability and effectiveness of the modified LSTM algorithm for CAI detection using spatiotemporal and kinematic characteristics in walking.

The experimental results show that the proposed data augmentation method promoted the detection performance in cases of the limited real dataset and the modified LSTM algorithm yielded an enhanced classification outcome to identify those CAI patients from a group of control subjects based on gait analysis data than any previous reports. This is because the Dual-GAN addresses the insufficient limitation of the training set, and the Convolutional LSTM-based detection model works well both in spatial and temporal features during walking.

The techniques proposed in this study develop a new way to extract significant features from a small clinical gait analysis dataset to improve computer-assisted diagnosis on CAI patients. This is a concrete step toward the long-term goal to develop artificial intelligence-based instruments for clinical diagnosis and rehabilitation. In subsequent studies, we will expand our research to posture control in more sports situations, such as running, jumping, and cutting, et al., and improve the performance of deep learning models to enhance the medical treatment in sports medicine.

Acknowledgments

The publication of this paper is funded by the Peking University Third Hospital—Haidian Innovation and Transformation Project under Grant Y74482-09, the National Natural Science Foundation of China under Grant 61801019, the China Scholarship Council under Grant 201906465021, and the Fundamental Research Funds for the University of Science and Technology Beijing under Grant FRF-BD-19-012A.

Conflict of interest

The authors declare there is no conflict of interest.

References

1. B. Burgesson, M. Glazebrook, S. Guillo, K. Matsui, M. D. Pastor, F. Peña, et al., Ankle instability (ICL 7), in *ESSKA Instructional Course Lecture Book: Barcelona 2016* (eds. R. Becker, G. M. M. J. Kerkhoffs, P. E. Gelber, M. Denti, R. Seil), Springer Berlin Heidelberg, Berlin, Heidelberg, (2016), 89–99. https://doi.org/10.1007/978-3-662-49114-0_7

2. M. H. Leonard, Injuries of the lateral ligaments of the ankle—a clinical and experimental study, *J. Bone Joint Surg. Am.*, **31** (1949), 373–377. <https://doi.org/10.2106/00004623-194931020-00013>
3. E. Kemler, K. M. Thijs, I. Badenbroek, I. G. L. van de Port, A. W. Hoes, F. J. G. Backx, Long-term prognosis of acute lateral ankle ligamentous sprains: High incidence of recurrences and residual symptoms, *Fam. Pract.*, **33** (2016), 596–600. <https://doi.org/10.1093/fampra/cmw076>
4. C. J. Powden, J. M. Hoch, M. C. Hoch, Rehabilitation and improvement of health-related quality-of-life detriments in individuals with chronic ankle instability: A meta-analysis, *J. Athl. Training*, **52** (2017), 753–765. <https://doi.org/10.4085/1062-6050-52.5.01>
5. R. Guo, X. Cheng, Z. C. Hou, J. Z. Ma, W. Q. Zheng, X. M. Wu, et al., A shoe-integrated sensor system for long-term center of pressure evaluation, *IEEE Sens. J.*, **21** (2021), 27037–27044. <https://doi.org/10.1109/JSEN.2021.3116249>
6. S. Mollà-Casanova, M. Inglés, P. Serra-Añó, Effects of balance training on functionality, ankle instability, and dynamic balance outcomes in people with chronic ankle instability: Systematic review and meta-analysis, *Clin. Rehabil.*, **35** (2021), 1694–1709. <https://doi.org/10.1177/02692155211022009>
7. K. G. Migel, E. A. Wikstrom, Immediate effects of vibration biofeedback on ankle kinematics in people with chronic ankle instability, *Clin. Biomech.*, **90** (2021), 105495. <https://doi.org/10.1016/j.clinbiomech.2021.105495>
8. S.-W. Kim, H. G. Jung, J. S. Lee, Ligament stabilization improved clinical and radiographic outcomes for individuals with chronic ankle instability and medial ankle osteoarthritis, *Knee Surg. Sports Tr. A.*, **28** (2020), 3294–3300. <https://doi.org/10.1007/s00167-020-05845-5>
9. S. Ashkani-Esfahani, R. Mojahed-Yazdi, R. Bhimani, G. M. Kerkhoffs, M. Maas, C. W. DiGiovanni, et al., Deep learning algorithms improve the detection of subtle lisfranc malalignments on weightbearing radiographs, *Foot Ankle Int.*, 2022. <https://doi.org/10.1177/10711007221093574>
10. K. Kipp, R. M. Palmieri-Smith, Differences in kinematic control of ankle joint motions in people with chronic ankle instability, *Clin. Biomech.*, **28** (2013), 562–567. <https://doi.org/10.1016/j.clinbiomech.2013.03.008>
11. R. M. Koldenhoven, J. Hart, S. Saliba, M. F. Abel, J. Hertel, Gait kinematics & kinetics at three walking speeds in individuals with chronic ankle instability and ankle sprain copers, *Gait Posture*, **74** (2019), 169–175. <https://doi.org/10.1016/j.gaitpost.2019.09.010>
12. T. Balasukumaran, U. Gottlieb, S. Springer, Spatiotemporal gait characteristics and ankle kinematics of backward walking in people with chronic ankle instability, *Sci. Rep.*, **10** (2020), 11515. <https://doi.org/10.1038/s41598-020-68385-5>
13. G. Andreopoulou, D. J. Mahad, T. H. Mercer, M. L. van der Linden, Test-retest reliability and minimal detectable change of ankle kinematics and spatiotemporal parameters in MS population, *Gait Posture*, **74** (2019), 218–222. <https://doi.org/10.1016/j.gaitpost.2019.09.015>
14. B. Stansfield, K. Hawkins, S. Adams, D. Church, Spatiotemporal and kinematic characteristics of gait initiation across a wide speed range, *Gait Posture*, **61** (2018), 331–338. <https://doi.org/10.1016/j.gaitpost.2018.02.003>
15. S. Ashkani-Esfahani, R. Mojahed Yazdi, R. Bhimani, G. M. Kerkhoffs, M. Maas, D. Guss, et al., Assessment of ankle fractures using deep learning algorithms and convolutional neural network, **7** (2021), 2473011421S00091. <https://doi.org/10.1177/2473011421S00091>

16. L. Xin, Z. Dezheng, Z. Bin, G. Qinwei, Z. Zhongshi, Gait kinematics of patients with lateral collateral ligament injuries of ankle, 2021. <https://doi.org/10.21203/rs.3.rs-22139/v1>
17. X. Liu, C. Zhao, B. Zheng, Q. Guo, Z. Zhang, A. Wulamu, et al., Synthesizing foot and ankle kinematic characteristics for lateral collateral ligament injuries detection, *IEEE Access*, **8** (2020), 188429–188440. <https://doi.org/10.1109/ACCESS.2020.3029616>
18. A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, preprint, arXiv:1511.06434, 2015.
19. H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, et al., StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks, (2017), 5908–5916. <https://doi.org/10.1109/ICCV.2017.629>
20. Z. L. Yi, H. Zhang, P. Tan, M. L. Gong, DualGAN: Unsupervised dual learning for image-to-image translation, in *Proceedings of the IEEE International Conference on Computer Vision*, (2017), 2868–2876. <https://doi.org/10.1109/Iccv.2017.310>
21. X. Wang, K. Tan, Y. Chen, CapsNet and Triple-GANs towards hyperspectral classification, in *2018 Fifth International Workshop on Earth Observation and Remote Sensing Applications*, (2018), 194–197. <https://doi.org/10.1109/EORSA.2018.8598574>
22. Y. Choi, M. Choi, M. Kim, J. W. Ha, S. Kim, J. Choo, StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2018), 8789–8797. <https://doi.org/10.1109/Cvpr.2018.00916>
23. J. Donahue, K. Simonyan, Large scale adversarial representation learning, in *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, 2019.
24. H. J. Tien, H. C. Yang, P. W. Shueng, J. C. Chen, Cone-beam CT image quality improvement using Cycle-Deblur consistent adversarial networks (Cycle-Deblur GAN) for chest CT imaging in breast cancer patients, *Sci. Rep.*, **11** (2021). <https://doi.org/10.1038/s41598-020-80803-2>
25. S. Nowozin, B. Cseke, R. Tomioka, f-GAN: Training generative neural samplers using variational divergence minimization, in *Advances in Neural Information Processing Systems 29*, 2016.
26. I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A. Courville, Improved training of wasserstein GANs, in *Advances in Neural Information Processing Systems 30 (Nips 2017)*, 2017.
27. M. Arjovsky, S. Chintala, L. Bottou, Wasserstein GAN, 2017. <https://doi.org/abs/1701.07875>
28. X. D. Mao, Q. Li, H. R. Xie, R. Y. K. Lau, Z. Wang, S. P. Smolley, Least squares generative adversarial networks, in *Proceedings of the IEEE International Conference on Computer Vision*, (2017), 2794–2802. <https://doi.org/10.1109/Iccv.2017.304>
29. S. Hu, Y. Shen, S. Wang, B. Lei, Brain MR to PET synthesis via bidirectional generative adversarial network, in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, Cham, (2020), 698–707.
30. S. Hu, B. Lei, S. Wang, Y. Wang, Z. Feng, Y. Shen, Bidirectional mapping generative adversarial networks for brain MR to PET synthesis, *IEEE. Trans. Med. Imaging*, **41** (2022), 145–157. <https://doi.org/10.1109/TMI.2021.3107013>
31. W. Yu, B. Lei, S. Wang, Y. Liu, Z. Feng, Y. Hu, et al., Morphological feature visualization of alzheimer’s disease via multidirectional perception GAN, *IEEE Trans. Neural Networks Learn. Syst.*, (2022), 1–15. <https://doi.org/10.1109/TNNLS.2021.3118369>

32. W. Yu, B. Lei, M. K. Ng, A. C. Cheung, Y. Shen, S. Wang, Tensorizing GAN with high-order pooling for alzheimer's disease assessment, *IEEE Trans. Neural Networks Learn. Syst.*, (2021), 1–15. <https://doi.org/10.1109/TNNLS.2021.3063516>
33. F. Pollastri, F. Bolelli, R. Paredes, C. Grana, Augmenting data with GANs to segment melanoma skin lesions, *Multimedia Tools Appl.*, **79** (2020), 15575–15592. <https://doi.org/10.1007/s11042-019-7717-y>
34. J. Yoon, D. Jarrett, M. Schaar, Time-series generative adversarial networks, in *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, 2019.
35. J. Simon, L. Doederlein, A. S. McIntosh, D. Metaxiotis, H. G. Bock, S. I. Wolf, The Heidelberg foot measurement method: Development, description and assessment, *Gait Posture*, **23** (2006), 411–424. <https://doi.org/10.1016/j.gaitpost.2005.07.003>
36. A. Graves, Long short-term memory, in *Supervised Sequence Labelling with Recurrent Neural Networks*, Springer Berlin Heidelberg, Berlin, Heidelberg, (2012), 37–45. https://doi.org/10.1007/978-3-642-24797-2_4
37. F. Karim, S. Majumdar, H. Darabi, S. Chen, LSTM fully convolutional networks for time series classification, *IEEE Access*, **6** (2018), 1662–1669. <https://doi.org/10.1109/ACCESS.2017.2779939>
38. T. N. Sainath, O. Vinyals, A. Senior, H. Sak, Convolutional, long short-term memory, fully connected deep neural networks, in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, (2015), 4580–4584. <https://doi.org/10.1109/ICASSP.2015.7178838>
39. E. Tsironi, P. Barros, C. Weber, S. Wermter, An analysis of convolutional long short-term memory recurrent neural networks for gesture recognition, *Neurocomputing*, **268** (2017), 76–86. <https://doi.org/10.1016/j.neucom.2016.12.088>
40. L. Van der Maaten, G. Hinton, Visualizing data using t-SNE, *J. Mach. Learn. Res.*, **9** (2008), 2579–2605.
41. M. Wattenberg, F. Viégas, I. Johnson, How to use t-SNE effectively, *Distill*, **1** (2016), e2. <https://doi.org/10.23915/distill.00002>
42. S. Arora, W. Hu, P. K. Kothari, An analysis of the T-SNE algorithm for data visualization, in *Conference On Learning Theory*, (2018), 1455–1462.
43. G. Marta, F. Simona, C. Andrea, B. Dario, S. Stefano, V. Federico, et al., Wearable biofeedback suit to promote and monitor aquatic exercises: A feasibility study, *IEEE Trans. Instrum. Meas.*, **69** (2020), 1219–1231. <https://doi.org/10.1109/TIM.2019.2911756>
44. A. R. Anwary, H. Yu, M. Vassallo, Optimal foot location for placing wearable IMU sensors and automatic feature extraction for gait analysis, *IEEE Sens. J.*, **18** (2018), 2555–2567. <https://doi.org/10.1109/JSEN.2017.2786587>
45. S. Qiu, H. Zhao, N. Jiang, Z. Wang, L. Liu, Y. An, et al., Multi-sensor information fusion based on machine learning for real applications in human activity recognition: State-of-the-art and research challenges, *Inf. Fusion*, **80** (2022), 241–265. <https://doi.org/10.1016/j.inffus.2021.11.006>
46. Z. Sun, Y. Tian, H. Li, J. Wang, A superlinear convergence feasible sequential quadratic programming algorithm for bipedal dynamic walking robot via discrete mechanics and optimal control, *Optim. Control. Appl. Methods*, **37** (2016), 1139–1161. <https://doi.org/10.1002/oca.2228>
47. Z. Sun, T. Shi, L. Wei, Y. Sun, K. Liu, L. Jin, Noise-suppressing zeroing neural network for online solving time-varying nonlinear optimization problem: A control-based approach, *Neural Comput. Appl.*, **32** (2020), 11505–11520. <https://doi.org/10.1007/s00521-019-04639-2>

48. Z. Sun, F. Li, B. Zhang, Y. Sun, L. Jin, Different modified zeroing neural dynamics with inherent tolerance to noises for time-varying reciprocal problems: A control-theoretic approach, *Neurocomputing*, **337** (2019), 165–179. <https://doi.org/10.1016/j.neucom.2019.01.064>



AIMS Press

©2022 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>).