ScooterID: Posture-based Continuous User Identification from Mobility Scooter Rides

Devan Shah, Ruoqi Huang, Nisha Vinayaga-Sureshkanth, Tingting Chen, Murtuza Jadliwala

***Abstract*—Mobility scooters serve as a powerful last-mile transportation tool for people with mobility challenges. Given the unique riding behavior and posture of mobility scooter riders, such user-specific mobility scooter ride data has tremendous potential towards the design of continuous user identification and authentication mechanisms. However, there have been no prior research efforts in the literature exploring this unique modality for the design of continuous user identification techniques. To address this gap, this paper proposes *ScooterID*, the first framework which employs rider posture data collected from cameras on mobility scooters to continuously identify (and authenticate) users/riders. As part of this framework, a machine learning based model comprising of a spatio-temporal Graph Convolutional Network and a body-part-informed encoder is designed to effectively capture a user's subtle upper-body movements during mobility scooter rides into discriminating embedding vectors. These embeddings can then be used to reliably and continuously identify and authenticate users/riders. Experiments with real-world mobility scooter ride data show that *ScooterID* achieves high levels of authentication accuracy with few enrollment video samples. *ScooterID* also performs efficiently on resource-constrained devices (e.g., Raspberry Pis) and is robust against adversarial perturbations to authentication inputs.**

***Index Terms*—Siamese Networks, Authentication, Machine Learning**

I. Introduction

The use of micro-mobility vehicles, which provide a convenient and versatile last-mile transportation mode, is increasing in urban areas [1], [2]. Among these, battery-powered mobility scooters (as shown in Figure 1a) are especially popular among the elderly population and people with disabilities or mobility challenges and are considered essential medical devices. Moreover, mobility scooters are capable of generating user-specific ride data, which has many potential applications. In this work, we explore the use of riding posture as a biometric for continuously authenticating or identifying mobility scooter users. This approach is advantageous both in addressing the security challenges of the target system and in enabling various personalized services.

Recently, a wave of mobility scooter thefts has been reported, driven by financial gain or personal use [3], [4]. Although one-time authentication methods, such as passwords or physical keys, are easy to deploy, they fall short when mobility scooter users must leave their devices unattended in public places without turning them off, for example, when receiving medical attention at hospitals or clinics [4]. Continuous authentication would be very advantageous in the above scenario. It enhances security by re-authenticating users during sessions after one-time authentication may fail, and it improves usability by being seamless and transparent, without requiring the legitimate users' attention or notifying them. This method is particularly well-suited to the needs of mobility scooter users, the majority of whom are senior citizens [5]. Such an authentication method is expected to supplement primary one-time means of authentication, such as passwords, tokens, and physical keys.

Another primary motivation for this study is to enable various personalized services on mobility scooters by identifying users through continuous posture data. In shared-scooter scenarios, where multiple users utilize the same mobility scooter at transportation hubs or community centers, it is crucial to understand the specific rider's preferences and behavior patterns to offer personalized services. For instance, when recommending accessible sidewalk navigation routes [6], a young user recovering from a car accident may prefer a shorter but more crowded route, while a senior citizen with slower reaction time or other conditions may need a longer but safer option. Our user identification system lays the foundation for such services by continuously building user profiles based on real-time posture data.

The research literature is full of continuous user authentication techniques, which employ a variety of input modalities such as touch, posture, speech and eye gaze captured by means of sensors, such as capacitive touch sensors, motion sensors, microphones, cameras and biofeedback (pressure and EEG) sensors [7]–[10]. However, the big question here is: *which sensor and input modality is appropriate for use as a biometric in our mobility scooter riding scenario?* For instance, several of the sensors mentioned above are not appropriate for a mobility scooter setting. In particular, adding touch and biofeedback (pressure or EEG) sensors to mobility scooters could potentially disrupt the riding functionality for users or cause unexpected discomfort during mobility scooter rides. Microphone and motion sensors could be potential candidates for this application, however, in mobility scooter riding scenarios, readings from microphones and motion sensor readings are adversely impacted by the constantly changing ambient or environmental conditions such as background noise, road restrictions and moving objects (e.g, vehicles and

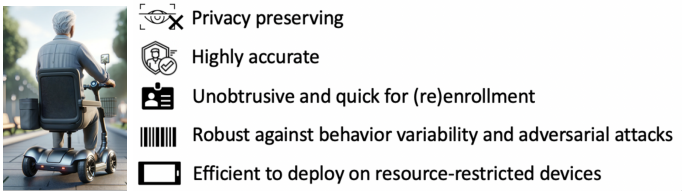
D. Shah is with Princeton University, Princeton, New Jersey, USA. Email: ds6237@princeton.edu

R. Huang is with California State Polytechnic University, Pomona, Pomona, California, USA. Email: ruoqihuang@cpp.edu

N. Vinayaga-Sureshkanth is with The University of Texas at San Antonio, San Antonio, Texas, USA. Email: nisha.vinayagasureshkanth@my.utsa.edu

T. Chen is with California State Polytechnic University, Pomona, Pomona, California, USA. Email: tingtingchen@cpp.edu

M. Jadliwala is with The University of Texas at San Antonio, San Antonio, Texas, USA. Email: murtuza.jadliwala@utsa.edu



(a) Scenario (b) *ScooterID* Design Goals/Challenges

Fig. 1: *ScooterID* Design Goals

pedestrians), which in turn, may adversely impact the user identification/authentication performance.

Cameras, in contrast, are relatively unobtrusive, do not require constant interaction with the user/rider, are relatively unimpaired by environmental conditions during rides and can continuously capture a significant amount of user information which could be used in creating unique biometric signatures, and thus, are a promising sensor choice for passive and continuous user identification/authentication in the mobility scooter setup. User privacy, on the other hand, is a significant challenge when employing cameras for any application, and any authentication scheme using users' video data should respect their privacy, as much as possible. To address the privacy concerns associate with usage of videa data, in this work we only use non-sensitive body posture-related information and features to generate biometrics. Moreover, we avoid the storage of any information which can be used to reconstruct any personally identifiable information in the video data.

Temporal posture data captures intricacies in a rider's physical biases and movement tendencies, which are abundant in human movement. Famously, walking patterns have been called the "sixth vital sign" [11], reflecting physiological changes and commonly used in diagnoses. Gait patterns are also commonly used in forensics and person identification [12], [13]. Based on the success of full gait analysis, we explore the potential of upper-body movement as a new biometric. It is currently not known, and there has been no prior work on, which features of the upper-body posture and movement patterns can be employed to accurately and reliably discern a user's identity during mobility scooter rides. An additional challenge in this regard is robustness against variability in the riding behavior of users and security against spoofing by malicious adversaries. We envision deep machine learning models playing a critical role in learning to correlate appropriate features from upper-body posture and movements (during mobility scooter rides) to user/rider identity. However, to design appropriate deep learning models we will require a large and representative dataset of mobility scooter rides for training and validation tasks. Unfortunately, currently there are no existing datasets of mobility scooter riding videos available in the literature. We also anticipate the proposed user identification and authentication framework, and the related classification models, to be deployed on resource-constrained edge devices. Thus, designing an efficient framework to run on those devices is a critical requirement, and yet another design challenge.

In this work, we present *ScooterID*, a deep learning framework for riding behavior and posture based continuous identification and authentication of mobility scooter riders. To achieve the design goals of this framework (summarized in Figure 1b), we extract the rider's upper-body keypoints (joints) from video frames and continuously process sequential coordinates while they ride the scooter to authenticate the rider. We propose a novel deep model for generating user-specific embedding vectors based on the spatio-temporal representation of posture features and design a Siamese network to learn a distance function between such embedding vectors. In the keypoint coordinates extraction, we explore three different human pose estimation models [14] with variations in output dimensions, to study their impact on user identification and authentication accuracy and system efficiency. In the deep model design, we leverage Graph Convolutional Networks (GCN) [15] and a bodypart-informed hierarchical encoding structure which have great capacity in extracting features representing unique spatial correlations of upper-body keypoints in time series when driving mobility scooters. Mobility scooter riders may have mobility challenges and medical conditions (e.g., stroke or Parkinson's disease) which can cause unique asymmetric movement patterns showing only in a local spatial region. Correspondingly, in the deep auto-encoder design of our framework, we group upper-body joints from local physiological regions and create a hierarchy to represent the unique movement pattern of each upper-body region and their interconnections in the final rider embedding. To tackle the challenge of riding behavior variability, our carefully-designed Siamese network can enable easy and efficient updates of riders' embeddings and maintain a high level of identification and authentication accuracy over time.

In summary, the contributions of this paper are as follows:

- We propose *ScooterID*, a novel continuous identification and authentication system for mobility scooter riders based on biometrics derived from their upper-body posture and riding behavior. The design of *ScooterID* consists of a Siamese network with a novel rider embedding generation model comprised of a spatio-temporal GCN and a pyramid bodypart-informed encoder to extract users' features that effectively capture their subtle patterns in upper-body movement while riding mobility scooters.
- We evaluate the performance of *ScooterID* by conducting experiments with real-world mobility scooter ride data collected from 42 volunteers. Our empirical results show that *ScooterID* achieves high levels of identification/authentication accuracy and is efficient enough for deployment in resource-constrained systems; *ScooterID* is also advantageous over relevant state-of-the-art models for behavior-based authentication as it is able to address user behavior variability with an easy and efficient re-enrollment mechanism.
- We also evaluate the robustness of *ScooterID* against a variety of input manipulation and adversarial scenarios, including pixel-level modifications, posture displacements and generative scenarios focusing on spatio-temporal attacks.

II. Related Work

Behavioral Biometrics-Based User Identification and Authentication - Owing to the increased presence of sophisticated sensors on modern mobile and wearable devices, recent research efforts have explored using various behavioral biometrics based methods for user identification [16], [17] and authentication [10], [18]–[20]. For instance, Sitová et. al [18] authenticate smartphone users using grasping and tapping patterns, while Kumar et. al. [10] accomplish it by fusing typing, swiping and phone movement patterns. Ehatisham-ul-Haq et. al. [19] leverage smartphone accelerometer, gyroscope, and magnetometer sensor data to determine whether walking, sitting, standing, running, or using a staircase offer distinguishing enough patterns for successful continuous authentication. These works show promising trends in leveraging different input modalities in providing real time and on-going user identification and authentication services in a ubiquitous computing environment, as a complement to the conventional one-shot or knowledge-based authentication schemes such as passwords.

Some behavioral biometrics-based authentication efforts in the literature have focused on identifying discriminating movement patterns to serve as biometric signatures [21]–[24]. For example, Bhalla et al. [25] leverage head movement preferences detectable from an AR headset to successfully authenticate a user from ambient movement every 3 seconds. Zhang et. al. [23] instead choose to authenticate VR users continuously via eye movement in response to implicit visual stimuli, achieving notable accuracy and adversarial robustness. Some other research efforts (e.g. [24]) have employed computer mouse dynamics for continuous authentication. These works have validated that the natural difference in each user's movement during some specific tasks, as captured by mobile and wearable device sensors, can be employed as effective biometrics for robust and continuous identification and authentication across multiple application scenarios.

However, most of the above efforts rely on sensors that are hard to deploy in a mobility scooter riding scenario without significantly interfering with users' riding activity or downgrading their comfort level. Vision-based behavioral modeling that employs a camera is a passive and less disruptive modality, but it needs an appropriate mobility scooter riding dataset for model design which, unfortunately, is currently unavailable in the research literature. To overcome this, we build a novel continuous user identification and authentication system from the ground up by collecting and using our own real-world mobility scooter riding posture data.

Deep Models for Continuous Authentication and Gait Analysis - For continuous authentication in dynamic environments, deep learning models show great promise due to their powerful feature learning capabilities. Some works have applied Recurrent Neural Network (RNN) models [26] in learning the sequential patterns of user behaviors e.g., [27]–[29]. Coskun et. al. [29] leverage a novel attention-based Long Short Term Memory (LSTM) Siamese network for human motion analysis and person identification. Convolutional Neural Networks (CNN) have also been widely applied in continuous behavioral biometrics-based authentication [30]–[33]. Fereidooni et. al. [33] perform continuous authentication for users of a mobile banking app with a 1D CNN in a Siamese network by using accelerometer, gyroscope and magnetometer data. Cardaioli et. al. [32] use user face images and employ a 2D CNN to authenticate users continuously despite camera blurring, offering reliable authentication and addressing user privacy. However, these prior works primarily leverage deep models to generate user representations out of temporal or simple spatial patterns (e.g., sequence of touch point coordinates). When extracting features from riding postures, we require more complicated networks to learn from sequences of higher dimensional video data that are correlated spatially by the human body confinement.

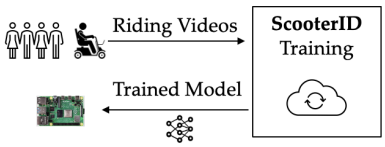
Deep gait analysis [34], [35], which is the study of recognizing how people walk using deep models, is also relevant to our work. Computer vision-based gait analysis attempts to learn user movement features from analysis of skeletons or silhouettes of subjects, but for different purposes such as sport training, health assessment, etc. Models for predicting clinical scores and analyzing gait in Parkinson's frequently leverage graph convolutional networks [36], [37]. Despite these existing works in deep gait analysis, mobility scooter riding involves upper-body movements which are drastically different than the full body engagement when people walk. Moreover, no existing deep gait analysis work has been tailed to meet the performance requirement of identification and authentication systems. Our work focuses on building a robust biometric for identifying and authenticating mobility scooter users using their unique upper-body movement features.

III. *ScooterID* Framework

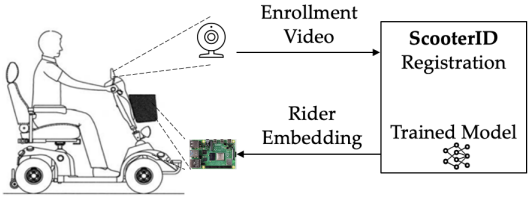
A. System Overview

ScooterID uses mobility scooter riders' upper-body posture to create unique user-specific signatures which can be used for identification and authentication. *ScooterID* employs a camera, capable of producing video frames of at least 196×196 pixels, installed on the handle of the scooter facing the upper-body of the rider. To protect privacy of users, we avoid storing information processed from user facial data, as such data (e.g., embeddings produced by neural networks) can be used to reconstruct a user's face [38]. Thus, in our initial processing of video frames, all user facial data is removed and future processing solely relies on torso information.

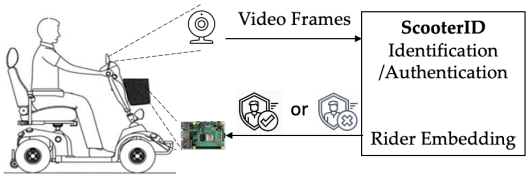
Similar to other deep learning-based authentication systems using behavioral biometrics (e.g., [39]), the *ScooterID* system is comprised of three stages, i.e., *training*, *registration* and *identification/authentication*. As shown in Figure 2, *ScooterID* leverages an existing repository of mobility scooter users' riding videos to train a carefully architected machine learning model which produces embeddings from input video segments. After the model is trained, and is available to the users via API access to a cloud server or local deployment, a new user can register into the identification/authentication system. To enroll in the system, a user rides the mobility scooter for between 40 and 135 seconds and video segment samples of the user riding are recorded and processed using



(a) Model Training Stage



(b) Rider Registration Stage



(c) Identification/Authentication Stage

Fig. 2: *ScooterID* System Model.

the trained model to produce the rider embedding, which is stored on-device. For mobility scooters with single users, in the authentication stage, when a user is riding the mobility scooter, the video frames from the camera are sampled and passed to *ScooterID*, which then yields an authentication result of "match" or "no match" by computing the similarity between the current rider's generated embedding and the enrolled user's stored embedding and leveraging thresholds determined by withheld test sets and user preference. For mobility scooters shared among multiple users, the identification stage is similar to authentication, except that the current rider's embedding will be compared with all the stored embedding belonged to n different users, and identified as one of them based on distance.

ScooterID uses only the videos of riding postures as input. The continuous video stream of the mobility scooter rider is sampled and processed into T-frame segments, each segment producing one identification/authentication decision. We leverage existing pose estimation techniques [40]–[42] to gather the user skeleton information from each processed video frame which depicts the spatial relation among the user's upper-body keypoints. Then the keypoint coordinates detected from T-frame video segments form a sequence which is fed into the subsequent neural network as input.

The identification and authentication functionality of *ScooterID*, requires to either classify the current rider as one of the registered users, or distinguish between a registered user and an unauthorized rider not registered with that scooter. To this end, we employ a **Siamese Network Architecture** [43], which can effectively learn a similarity function to compute the distance (closeness) between the embedding of the registered riders and that of the current rider. For training the Siamese network, different users' riding video segments are grouped into triplets. Three input samples are fed to the network in succession with the weights being identical. The loss values are calculated using all three input samples and then back-propagated. In the identification/authentication stage, *ScooterID* employs the trained Siamese Network to output the distance (e.g., Euclidean distance) between an input pair (i.e., current rider's and the registered user's embeddings).

As shown in Figure 3, *ScooterID* system design includes the following main components: 1) input data acquisition, 2) pose estimation to generate upper-body keypoint coordinates, 3) a graph convolutional neural network to extract spatio-temporal features of upper-body keypoints movement, 4) a bodypart-informed deep encoder to generate an embedding for the user, 5) a triplet loss function for model training, and a distance-based classifier for the final identification and authentication result. The last step, i.e., distance-based classification step, can be customized based on users' preferences of appropriate distance threshold values which in turn determines system sensitivity and usability.

B. Pose Estimation for ScooterID

The first step towards abstracting discriminating features to represent different users in *ScooterID* is to perform *human pose estimation* using the sampled frames from the rider's video. Specifically, human pose estimation is used to detect/predict *keypoints* (on human body) such as hands, head and elbow in 2D or 3D coordinate system from an image of the human body. Figure 4 shows a rider's upper-body keypoints detected by the pose estimation in 2D and 3D representation. Intuitively, 3D coordinates of keypoints contain more information of upper-body posture compared to 2D coordinates. In Figure 4b, the red arrows illustrate the movement of the two arms when making a left turn while riding the mobility scooter. In this scenario, 3D representation may have advantages in capturing the posture spatial feature in anterior and posterior movements than 2D coordinates. 2D coordinates may not be able to fully capture the detailed movement of wrists and other keypoints in the depth dimension. However, on the other hand, it may cause other upper-body keypoint coordinates' change correspondingly, which can be reflected in their 2D coordinates.

For *ScooterID*, we experiment with three existing human pose estimation models to gather keypoints. In particular, we employ MoveNet [42], MediaPipe [40], and Yolov7 [41], among which MoveNet and Yolov7 produce 2D keypoint coordinates while MediaPipe yields 3D coordinates. The objectives of testing three pose estimation models for *ScooterID* are: (i) to investigate the feasibility of running the models on a resource-constrained platform such as Raspberry Pi, based on two popular machine learning frameworks, i.e., Pytorch (Yolov7) and Tensorflow (MediaPipe and MoveNet), and (ii) to determine if a 3D coordinate representation of keypoints can be more advantageous in accurately computing the final embeddings for the identification/authentication task, compared to 2D coordinates. In Section IV, we will present the related results in more details.

We process video frames from the camera one at a time, with a stride length s depending on the processing speed of

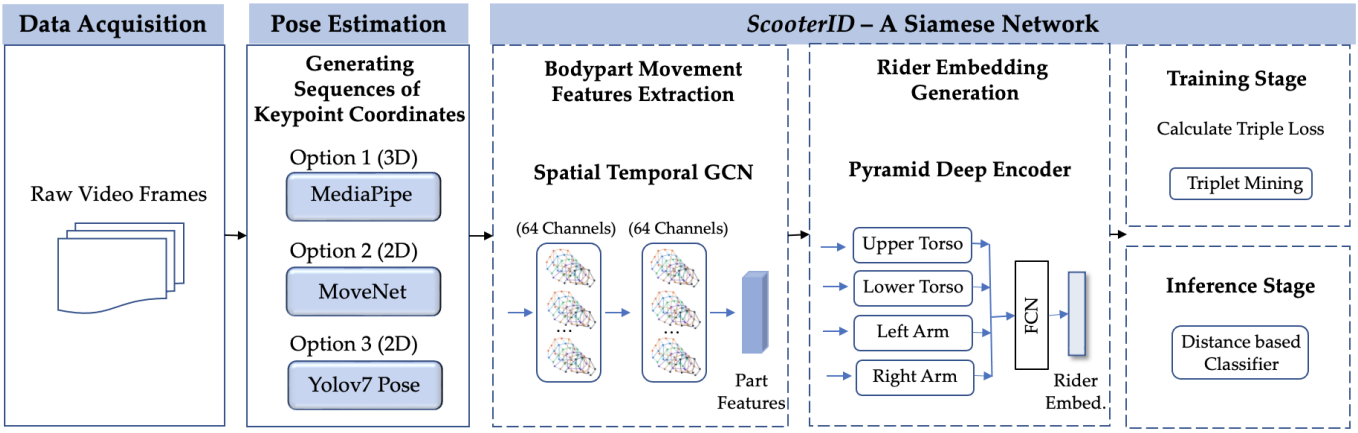
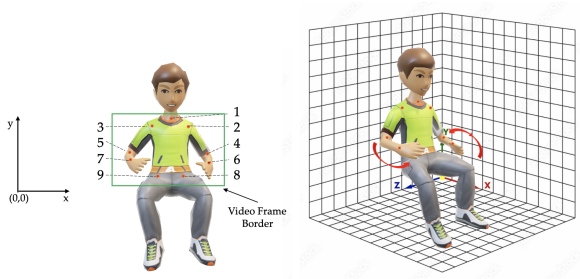


Fig. 3: System Framework of *ScooterID* that Authenticates Mobility Scooter Riders based on Riding Postures



(a) 2D Representation (b) 3D Representation

Fig. 4: 2D vs 3D representations of upper-body keypoints: neck, left and right shoulders, left and right elbows, left and right wrists, left and right hip.

the chosen pose estimation model. The pose estimation result of each frame is expected to include the coordinates of the nine keypoints as shown in Figure 4. For frames where the pose estimation models have low confidence, we use the coordinates from prior frames. We group the N $(= 9)$ keypoints from T frames into a sequence, and use the sequences as inputs to the embedding generation model which comprises of a spatio-temporal upper-body GCN and a bodypart-informed deep encoder.

C. Spatio-Temporal Upper-body GCN

To capture the spatial and temporal characteristics of the movement of upper-body keypoints during scooter riding, using the sequence of keypoint coordinates, we construct a *spatio-temporal graph* as shown in Figure 5. In particular, we construct an undirected spatio-temporal graph $G = (V, E)$ based on the sequence of upper-body keypoint coordinates, with N keypoints (with either 2D or 3D coordinates), and T video frames. The set of nodes is denoted as $V = \{v_{i,t} | 1 \leq i \leq N, 1 \leq t \leq T\}$, where i is the keypoint index, and t is the frame index. We associate each node with a 2D or 3D vector of features (coordinate values). In the spatio-temporal graph, we define two types of edges: (i) *spatial edges* – connecting nodes where the corresponding upper-body keypoints are connected physically, such as, left elbow and left shoulder (connected by left upper arm); (ii) *temporal edges* – connecting the same keypoint in neighboring frames. Formally, $E = \{\{v_{i,t}, v_{j,t}\} | \{i, j\} \in P, 1 \leq t \leq T\} \cup \{\{v_{i,t}, v_{i,t+1}\} | 1 \leq i \leq N, 1 \leq t \leq T - 1\}$, where P is a set of physically connected keypoint pairs. We let $\boldsymbol{A}$ denote the adjacency matrix of the spatio-temporal graph $G = (V, E)$.

Then, based on the graph G, we form a *Graph Convolutional Neural Network (GCN)* composed of two graph convolution layers, each followed by the batch normalization and ReLU activation, with implementation adapted from [44] [45]. The layer-wise propagation rule is represented as

$$g(\boldsymbol{H}^{(l)}, \boldsymbol{A}) = \sigma(\hat{\boldsymbol{D}}^{-1/2} \hat{\boldsymbol{A}} \hat{\boldsymbol{D}}^{-1/2} \boldsymbol{H}^{(l)} \boldsymbol{W}^{(l)}) \quad (1)$$

with $\hat{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}$ denoting the adjacency matrix with inserted self-loops, where $\boldsymbol{I}$ is the identity matrix. $\hat{\boldsymbol{D}}$ is the diagonal node degree matrix of $\hat{\boldsymbol{A}}$. The identity matrix $\boldsymbol{I}$ is added to ensure each node is included in the convolution process and $\hat{\boldsymbol{D}}^{-1/2} \hat{\boldsymbol{A}} \hat{\boldsymbol{D}}^{-1/2}$ is for normalizing $\hat{\boldsymbol{A}}$. $\boldsymbol{W}$ is the connection parameter matrix, and $\boldsymbol{H}^{(l)}$ is the node in hidden layer l. The edge weights in the GCN are set to 1, and our activation function σ is ReLU. This layer-wise propogation rule allows the flow of state between adjacent key-point nodes and, when repeated n times, allows each node to develop an understanding of its local region of keypoints at most n edges away.

With Eq. (1) indicating the aggregating node representations from their direct neighborhood, GCN has a clear meaning of vertex localization. In our GCN, we have two layers with 64 channels each, extending the receptive region for each resultant joint embedding, while being less computationally intensive than a single layer with more edges. The edges in the GCN enable the network structure to learn the spatial and temporal relations of different keypoints in upper-body movement during mobility scooter riding. The GCN produces an encoding composed of the correlated keypoints movement features as its output.

D. Bodypart-Informed Deep Encoder

Following the GCN, as described above, we leverage the spatial hierarchy of the upper-body, which is tied to the

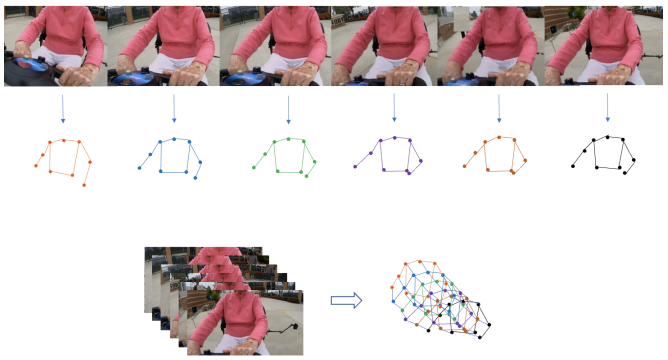


Fig. 5: Generation of a spatio-temporal graph from a sequence of rider's video frames and their corresponding a skeletal subgraphs with 9 keypoint nodes and edges representing physical connections on the body.

riding posture, to design the deep encoder for generating rider embeddings. The encoder allows increased data processing and more abstract pattern recognition while maintaining model efficiency due to localized scope. We group the upper-body keypoints into four regional subsets, corresponding to four body parts, i.e., the upper torso (neck, left shoulder, right shoulder), lower torso (left hip, right hip), left arm (left shoulder, left elbow, left wrist), and right arm (right shoulder, right elbow, right wrist).

Each of the body part is processed by a deep encoder that consists of five residual convolutional layers each followed by the batch normalization and a ReLU activation. The residual connections prevent over-processing and address the vanishing gradient problem. After the second and fourth convolution layer, there is a 1D max pooling layer. Each encoder produces a d-dimensional segment embedding via global average pooling. Given these four embeddings, a two-layer fully connected network with a ReLU activation then produces a single embedding for the T-frame video sample.

In other words, *ScooterID* movement processing begins locally at each keypoint, before employing the results of that layer at a broader level with the regional convolutions, and culminating in a final fully connected network to generate the rider's embedding. Such a pyramidal architecture has several advantages. First, it allows the model to focus on one body part at a time, constructing embedding vectors for each bodypart corresponding to the physical constraints in the body movement, denoted as part features in Figure 6. Second, such segmented convolutions are more efficient compared to full convolutions, especially in a computationally constrained setting/device.

E. Model Configuration

Before discussing the loss function used in our network, we summarize our rider embedding generation model with more configuration details in this subsection. Figure 6a depicts the detailed architecture of *ScooterID*'s rider embedding generation model, which takes a sequence of keypoint coordinates as input and outputs an embedding vector. The input dimension is $(T, 9, 3)$ for 9 keypoints with 3D coordinates, with T being the number of frames that are used to make an identification/authentication decision. The model has two layers of graph convolutions in the Spatial Temporal GCN. Each graph convolutional layer is followed by a Batch Normalization and a ReLU activation function. The output of each graph convolutional layer is of dimension $(T, 9, 64)$. Each of the 9 keypoints is represented with a feature of size 64. The $(T, 9, 64)$ output of the second graph convolutional layer is taken by four bodypart encoders as input. The encoders extract features for different upperbody parts, i.e., upper torso, lower torso, left arm and right arm, each outputting a feature of dimension 64. After combining the four bodypart features with the fully connected layers with ReLU activation in the final step, the rider's final embedding of size 64 is produced.

Figure 6b shows the detailed architecture of the bodypart encoder model, as described above. Since there are 3 keypoints in each part, with $64 * 3$, the input to the encoders is of dimension $(192, T)$, except for the lower torso being $(128, T)$ with 2 keypoints only. The part encoder consists of residual convolutional layers with ReLU, max-pooling layers and a terminating global average layer.

F. Loss Function and Triplet Mining

In order to make the embedding discriminating enough to perform mobility scooter rider identification and authentication, in our model training, we use the *triplet metric loss function* [46]. The primary motivation for utilizing triplet metric loss is to encourage the model to map video samples from the same rider near to each other. The triplet metric loss function, as described in Eq. (2), minimizes the Euclidean distance between embeddings from the same user and maximizes the distance between embeddings from different users.

$$L(x^a, x^p, x^n) = \max(||f(x^a) - f(x^p)||_2 - ||f(x^a) - f(x^n)||_2 + \alpha, 0) \tag{2}$$

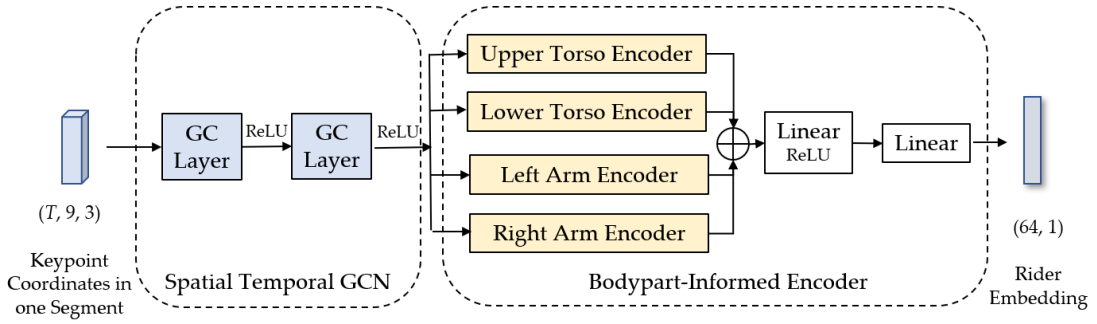
Here, x^a and x^p represent embedding vectors from the same user, denoted as the anchor embedding and positive embedding respectively. x^n is an embedding from a different user. α represents the margin between positive and negative pairs, and f is the model function. To minimize the loss, the model aims to satisfy the following:

$$||f(x^a) - f(x^p)||_2 + \alpha < ||f(x^a) - f(x^n)||_2 \tag{3}$$

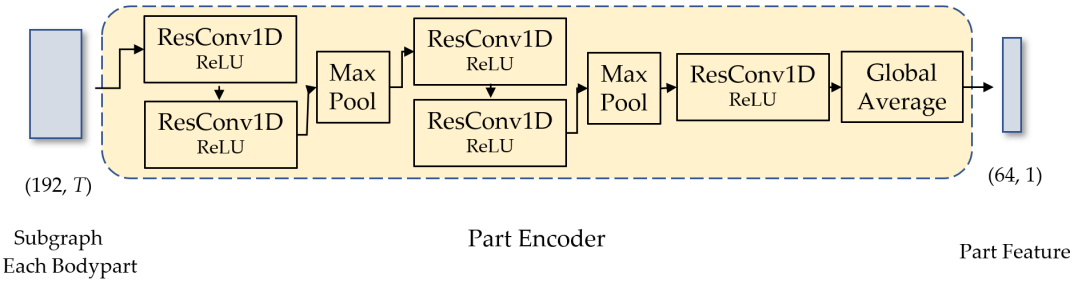
This implies that embeddings from the same rider will be pulled closer, whereas those from different riders will be further away, with an enforced margin of α between samples from different classes.

Rather than using all possible triplets of anchor, positive and negative samples in the training dataset, we accelerate model training by performing easy-positive and semi-hard-negative *Triplet Mining* [47]. As illustrated in Figure 7a, for each anchor in a batch, we choose from the batch the closest positive embedding to the anchor sample and the closest negative embedding that is not closer than the positive embedding to produce challenging triplets to train the model.

By employing the triplet loss mechanism in training, the distance between the same rider's embeddings decreases and that between different riders' embeddings increases. In this

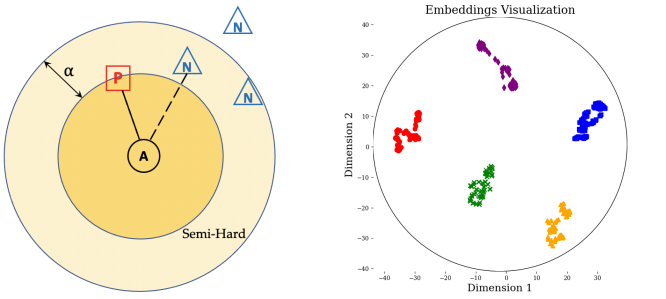


(a) Rider Embedding Generation Overall Architecture



(b) BodyPart Movement Encoder

Fig. 6: Model Configuration



(a) Illustration of triplet mining with easy positive and semi-hard negative

(b) t-SNE visualization of 5 riders' embeddings after triplet mining

Fig. 7: Training Visualization

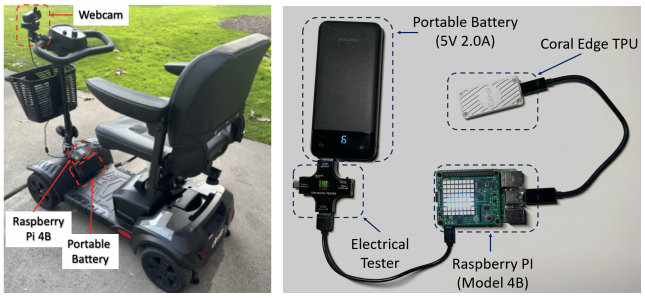
way, the embeddings can capture the subtle differences in distinct riders' upper-body riding behavior to distinguish them, and thus produce reliable identification/authentication results. Figure 7b shows the distribution of embeddings belonging to five different riders in one group (100 embeddings for each rider), with t-SNE [48] visualization, after triplet loss based training. As we can see, embeddings for the same rider with the same color are clustered closely while those for different riders are far apart, indicating that riders are separable in the embedding space transformed from the upper-body riding behaviors.

IV. Evaluation

ScooterID can conduct two different tasks, i.e., user authentication and user identification. To thoroughly assess *ScooterID*, for the user authentication task, we carry out comprehensive evaluations to assess it's accuracy, reliability and efficiency under a variety of settings and operational parameters. Due to limitation of space, for the user identification task, we only include basic performance evaluation results in Section IV-H. But because both user authentication and identification in *ScooterID* are based on embedding distances generated by the same deep learning framework, we expect results for user identification similar to those reported for authentication in various settings.

A. Data Collection

We collect mobility scooter riding data from 42 (9 female, 33 male) on-campus participants with ages ranging from 18 to 90, weights from 140 to 200 lbs, and heights ranging from 66 to 73 $inches$. In order to have the most diverse group of participants possible, we do not screen participants based on their races, prior mobility scooter riding experience, body shapes, or health conditions. They are preinformed with the same set of riding tasks, and the participation is based on the volunteers' willingness and self assessment. To collect the video frames of participants' upper-body movements when riding the mobility scooter, we mount a web camera on the handle facing the rider and focusing on the areas below the neck, as showcased in Figure 8a. Participants spent between $14 - 18$ minutes to complete the riding tasks on private university roads, including forward riding, backwards riding, $45°$ and $90°$ left and right turns, $360°$ rotations, both on-pavement and on-grass riding, and sudden acceleration and deceleration with gradual inclines and descents. After filtering out participants who either did not follow the tasks correctly or concluded the tasks early, we were able to record riding video data from 35 different/unique participants. Overall, our



(a) System setup on mobility scooter (b) Resource utilization test setup on a Raspberry Pi 4B.

Fig. 8: Illustration of Experiment Setup

evaluation dataset contains approximately 15 hours and 41 minutes of recorded mobility scooter riding video footage, with a total of 1.69 million frames, which will be publicly-available soon. Our data collection and experiments have been approved by the university's Institutional Review Board or IRB.

B. Experiment Setup

We implement our proposed *ScooterID* system in Python. The main Python libraries that we used for the implementation of our model architecture and training phase include PyTorch 2.1, PyTorch Geometric 2.4, and PyTorch Metric Learning 2.3. The Adam optimizer is used, and the learning rate is set as 0.001 for 50 epochs of training based on empirical observation. We utilize a Tewiky TW-05 Webcam with wide angle lens [49] to capture riding videos and the on-device evaluations are performed on a Raspberry Pi 4B comprising of a quad-core Cortex-A72 (ARM v8) CPU and 4GB Ram memory running on Raspberry Pi OS (64 bit). Identification/Authentication models are trained on computing clusters at the authors' institutions and on Google Cloud Computing. The computation devices used include Intel Skylake 32-core CPUs, DL160 CPUs, Nvidia A100 GPUs, and Tesla P100 GPUs.

For each test, we divide the dataset of 35 participants into 7 subsets and each subset contains 5 participants' data. We apply 7-fold cross validation - the riding video data in 6 subsets (from 30 participants) is used as training data and 1 subset with 5 participants' data (14%) is used as test data in the evaluation. From the videos used for testing, we gather authentication enrollment samples across the first 135 seconds of each scooter use. The evaluation results are the average of the cross-validation. We test the model with $T = 40$ frames per authentication decision and 10 fps video input.

C. Performance Metrics

We assess the authentication accuracy of *ScooterID* using the following metrics.

- *False Acceptance Rate (FAR)* captures the rate at which *ScooterID* incorrectly accepts an authentication attempt by unauthorized users. In other words, FAR is the ratio of the number of video segments in which authentication attempts by unauthorized users is incorrectly accepted by *ScooterID* (as valid authentication) to the total number of tested video segments from unauthorized users.
- *False Rejection Rate (FRR)* captures the rate at which *ScooterID* incorrectly rejects an authorized user's authentication attempt. In other words, FRR is the ratio of the number of video segments in which authentication attempts by authorized users are incorrectly rejected to the total number of tested video segments from authorized users.
- *Area Under the Curve of the Receiver Operating Characteristic Curve (AUCROC)* refers to the ability of the system to discriminate between authorized and unauthorized mobility scooter riding videos across all possible thresholds where the ROC curves are generated using FAR as x coordinates and (1-FRR), and the *True Acceptance Rate*, as y coordinates.
- *Equal Error Rate (EER)* is the point when FAR and FRR are equal. By varying different thresholds, this point can be found when the curves for FAR and FRR intersect, i.e., when $EER = FAR = FRR$, meaning when the system is equally likely to incorrectly accept an unauthorized user and to incorrectly reject an authorized user. With a lower EER, the system is considered more secure and usable.
- *True Acceptance Rate (TAR)* is used to measure the accuracy of *ScooterID* when performing the user identification task. TAR is the number of correctly identified test users among a group of known users (determined by the lowest embedding distance) over the total number of tests.

Because users of mobility scooters may have various preferences in setting their authentication system alarm threshold, with different considerations for security/sensitivity and user experience, instead of FAR and FRR which depend on threshold, we use AUCROC and EER in our evaluations as is the case in many other similar biometrics-based authentication works, e.g., [28], [50].

D. ScooterID Performance

In this section, we outline the evaluation results measuring *ScooterID*'s performance in discriminating between authorized and unauthorized mobility scooter riding videos. We first demonstrate (in Figure 9) the efficacy of the distance-based mechanism used in *ScooterID* by showing an example instance of how distance between embeddings (obtained by our models) change when switching from an authentic user to an unauthorized user. The figure shows one distance value between the embeddings for every 40 frames. The figure also clearly shows the gap when the authentic user (user ID 25) is switched by an unauthorized user (user ID 1), i.e, the left side distance measure is when the authentic user is riding, while the right side measure is when the unauthorized user takes over and starts riding. It is clear from this sample distance plot that the user switch causes an abrupt and significant (more than 2×) increase of the distance between the embeddings, which can be used by *ScooterID* to detect the unauthorized takeover. Next, we evaluate the impact of parameters such as embedding vector size (dimension), amount of video segments used for

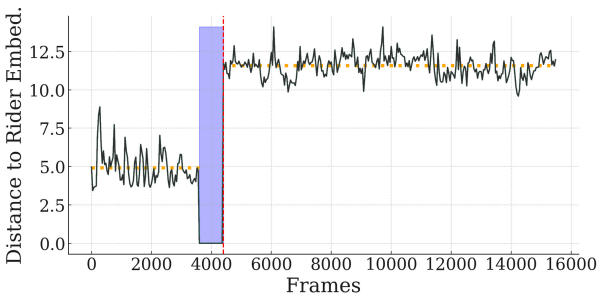


Fig. 9: Observed distance change between embeddings when switching from legitimate user (ID 25) to unauthorized rider (ID 1).

user enrollment (enrollment samples), and the choice of pose estimation model on system performance measured in terms of AUCROC and EER.

Impact of Embedding Vector Size - We vary the size of the embedding vector generated by the *ScooterID* model between 10, 30, and 60 dimensions, and evaluate the corresponding authentication accuracy with 40 enrollment samples. Figures 10a, 10b, and 10c shows the ROC curves when using Yolov7, MediaPipe and MoveNet, respectively. In each figure, we plot three ROC curves with the embedding vectors of size 10, 30, 60. From these plots we can see that, in general, the system performance is increasing with larger embedding vector sizes and no matter which pose estimation model is used, *ScooterID* achieves high levels of accuracy, as reflected by the ROC curves and AUCROC values (ranging from 0.832 to 0.995).

Impact of Number of Enrollment Samples: We investigate the impact of the number of enrollment samples used to create a new user's embedding vector on the system accuracy performance. We measure AUCROC and EER when the number of enrollment samples varies from 1 to 40 and the embedding vector size is fixed at 60. As shown in Table I, *ScooterID* yields high AUCROC values (ranging from 0.8390 to 0.9689) using very few enrollment samples across all the three different pose estimation models. Within the results shown, the peak AUCROC values (0.912, 0.901, 0.969 for MoveNet, MediaPipe, Yolov7, respectively) are all obtained when the number of enrollment samples is 40, while the increasing trend of AUCROC is not consistent with the number of enrollment samples varying from 1 to 20. For the EER values, it also shows *ScooterID* has strong performance in achieving high levels of accuracy (low EER values from 0.0319 to 0.2244) with few enrollment samples. Similar to AUCROC, we observe that more enrollment samples do not always generate more accurate models for authentication.

Impact of Pose Estimation Models: From Figure 10 and Table I, we can also evaluate the impact of pose estimation models on *ScooterID*'s performance. From these results, we observe that *ScooterID* with Yolov7 yields exceptional overall performance in learning a user's upper-body movement features while riding a mobility scooter, thus resulting in better authentication accuracy, compared to the MoveNet and

TABLE I: AUCROC and EER of *ScooterID* with Varying Numbers of Enrollment Samples

Pose Option	Number of Enrollment Samples				
	1	5	10	20	40
AUCROC					
MoveNet	0.8724	0.9119	0.9060	0.9019	0.9120
MediaPipe	0.8390	0.8787	0.8991	0.8976	0.9011
Yolov7	0.9672	0.9594	0.9667	0.9668	**0.9689**
EER					
MoveNet	0.1732	0.1396	0.1450	0.1506	0.1409
MediaPipe	0.2244	0.1918	0.1892	0.1525	0.1657
Yolov7	0.0786	0.0401	**0.0319**	0.0380	0.0771

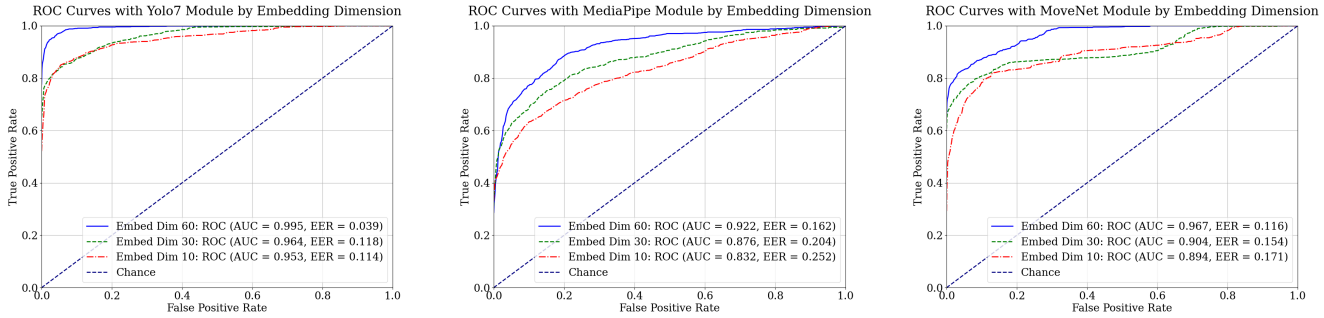
MediaPipe. We can also see that *ScooterID* with the MediaPipe pose estimation model resulted in lower accuracy compared to the other two models, although MediaPipe produces 3D coordinates of rider's upper-body keypoints. This is most probably due to the fact that the MediaPipe pose estimation model works by first locating the nose of a person and then using it to estimate the coordinates of the other landmarks (keypoints). Due to the privacy-aware nature of our design, the captured video frames do not include the riders' faces, and thus the MediaPipe model finds it challenging to effectively generate the 3D coordinates of the keypoints. Comparatively, the MoveNet model which uses the body center as the base or starting point for pose estimation performs much better. For the remainder of the evaluation results for *ScooterID*, unless otherwise stated, we assume an embedding vector dimension of 60 and an enrollment sample number of 40, with Yolov7 being employed as the pose estimation model.

E. Comparative Performance

Next, we perform an ablation study to investigate the impact of the major components of *ScooterID*'s embedding extraction architecture on its performance, and also comparatively study *ScooterID* with two relevant state-of-the-art models.

Models for Ablation Study: To study the effect of the two major components in *ScooterID*, i.e., spatio-temporal upper-body GCN and bodypart-informed deep encoder, we create two alternative models by removing these two components one at a time. In particular, we first create a *No-GCN* model, which directly takes sequences of keypoint coordinates, without the two graph convolutional layers, as input to the deep encoder with pyramid. The other components of *ScooterID*remain the same in the No-GCN model. Next, we also create a *No-Pyramid* model, which is trained without separate bodypart encoders, but instead with only one residual 1D CNN based deep encoder which takes in 9 keypoint coordinates together to create the embedding vector.

Models for Comparative Study: We also comparatively study the accuracy of *ScooterID* with two other well-known models in the literature. First is *Dynamic Time Warping (DTW)* [52], which is an algorithm for measuring similarity between two temporal sequences and has been widely applied in behavioral biometrics-based continuous authentication (e.g., [53]). The second model is based on the *attention mechanism* [54] which adds soft weights to the neural network that can



(a) Pose estimation model Yolov7 (b) Pose estimation model MediaPipe (c) Pose estimation model MoveNet

Fig. 10: ROC Curves of pose estimation models of *ScooterID* with varying embedding vector sizes.

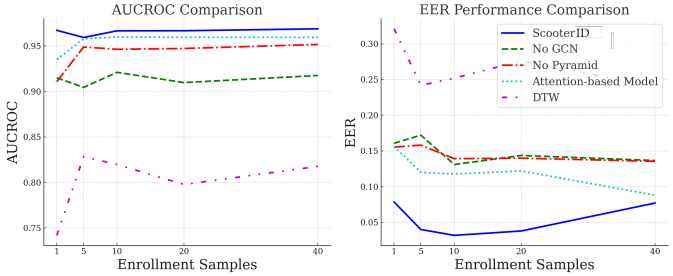


Fig. 11: Comparison of AUCROC and EER between *ScooterID* and attention-based model [51], DTW [52], No-GCN and No-Pyramid models.

emphasize features based on relevance, and is another state-of-the-art method in biometrics-based continuous authentication (e.g., [55]). For the second model, each node in the spatio-temporal graph attends to connected nodes with 10-headed dynamic graph attention with output concatenation [51].

Performance across Models: We measure the AUCROC and EER of all the 5 models (i.e., *ScooterID*, No-GCN, No-Pyramid, Attention-based Model and DTW), keeping the model parameters the same, i.e., input dimension: $(40 \times 9 \times 2)$ and embedding vector dimension 60 with various numbers of enrollment samples. As we can see from the results (Figure 11), *ScooterID* outperforms the other four models with higher AUCROC and lower EER values across all settings. This shows the significance of the GCN and the pyramid structure of the deep encoder in our architecture, as without either of them the accuracy of *ScooterID* is impacted negatively. The spatio-temporal GCN has greater positive impact on the performance of *ScooterID* than the pyramid in the deep encoder. We note that the EER value with 40 enrollment samples shows an unexpected increase compared with 20 samples. We believe this fluctuation is due to oversampling for some users with a short enrollment period. In *ScooterID* system deployment, the amount of enrollment samples can be optimized for different users and enrollment experiences to achieve best performance. At the same time we do not find that oversampling significantly affects other performance metrics. On the other hand, attention-based architecture achieves AUCROC values close to *ScooterID*, but not higher. However, based on the results of training and 1000 inference runs, the average training and inference times for the attention-based architecture is $6.847\times$ and $1.238\times$ slower than *ScooterID*, respectively.

F. Resource Utilization

We evaluate the execution efficiency of *ScooterID* on both a Raspberry Pi 4B (with Quad-core ARM v8 CPU and 4GB RAM running a 64-bit Raspberry Pi OS, as shown in Figure 8b) and an Acer Aspire A515-46 laptop (with AMD Ryzen 3 3350U Quad-Core CPU and 4GB RAM memory running Windows 11 OS). To begin our evaluation, we first break down the *ScooterID* prototype into core steps, and test the execution time and energy consumption during authentication. As a baseline for measuring energy consumption, we use a power bank of +5V 3A with capacity of $10400mAh$. We first focus on the resources utilized by the pose estimation and authentication inference tasks, not including the cameras used to capture the riding related video frames. We evaluate the feasibility of the three off-the-shelf pose estimation models in real-time authentication tasks on a resource-constraint platform such as a Raspberry Pi. Therefore, we implement multiprocessing for the pose estimation module to accelerate the process. For MoveNet implementation on the Raspberry Pi, we leverage the MoveNet.SinglePose.Lightning tflite model [56] with a Coral Edge Tensor Processing Unit (TPU) [57]. Alongside TPU, multiprocessing plays a significant role in maximizing CPU utilization for highly optimized models like MediaPipe and MoveNet. The presented results (in Table II) are averaged over 50 authentication runs.

TABLE II: Resource Consumption of *ScooterID* models inference on Acer Aspire A515-46 Laptop and Raspberry Pi 4.

	Acer Aspire		Raspberry Pi 4	
Model	**Time** (ms)	**Energy** (mJ)	**Time** (ms)	**Energy** (mJ)
MediaPipe	46.5	262	103.3	297
MoveNet	39.9	333	91.1	391
Yolov7	177.8	1785	1320	5623
GCN+Encoder	73.8	766	124	570

In Table II, the time column refers to the time consumed by each of the three pose estimation models for processing one video frame. For the GCN+Encoder, we measure the time required to generate an embedding from an input sample (a keypoint coordinates segment). As we can see, MoveNet

is the fastest option for pose estimation in both the laptop environment ($39.9ms$) and the Raspberry Pi environment ($91.1ms$), with less than $100ms$ per frame on the Raspberry Pi 4. While MediaPipe is slightly slower than MoveNet, it causes less energy consumption on both platforms ($262mJ$ and $297mJ$ respectively). A higher ratio of time versus energy consumption indicates that MediaPipe is the most energy efficient out of the three pose estimation models. Yolov7 is significantly more time and energy consuming than the other two models. The GCN and encoder implementations are sufficiently efficient, with execution times of $73.8ms$ on the laptop and $124ms$ on the Raspberry Pi, respectively.

TABLE III: Total Power Consumption of *ScooterID* on Raspberry Pi 4 with Camera

	Video Capture	***ScooterID* (MoveNet)**	***ScooterID* (MediaPipe)**	***ScooterID* (Yolov7)**
Power (W)	4.0	6.4	4.9	5.0

We also evaluate the total power consumption including the part from the camera capturing video frames. Table III shows the total power consumption experiment results of *ScooterID* on a Raspberry Pi 4 and the web camera. The power consumption of camera for video frame capture is 4.0 watts, while the *ScooterID* models (including both pose estimation and authentication inference) consume slightly more power, i.e., 6.4 watts, 4.9 watts and 5.0 watts respectively for MoveNet, MediaPipe and Yolov7. We observe a notable increase in energy consumption when a TPU is added for running MoveNet.

G. User Behavior Variability

Mobility scooter riders typically have progressive medical conditions such as stroke or neuropathy, oftentimes with impairments in upper extremities, which may cause variability in their upper-body postures. Next, we study how behavior variability of mobility scooter riders can affect the performance of *ScooterID* and determine the need for periodic authentication re-enrollment (i.e., updating registration embedding using new video segment samples). We randomly select five senior participants from the 35 volunteers and conduct a 20-week long longitudinal study. Each participant has two riding sessions. During the first session, participants are enrolled with *ScooterID* and registration embeddings are generated using video samples from their rides during the session. The second session is 1, 5, or 20 weeks after the first session and differs for each participant - participants P1 and P2 participate in the second session 1 week later, participant P3 participates 5 weeks later and participants P4 and P5 participate 20 weeks later. We test the average AUCROC and EER in the second session while using the old registration embeddings gathered in the first session. The results for these tests are outlined in Table IV.

From these results, we notice that when test sessions are one week later from the time of enrollment, *ScooterID* continues to achieve high accuracy as seen in Table IV. However, not surprisingly, when the registration is done 5 or 20 weeks earlier than the test session, the accuracy in terms of AUCROC and EER detiorate significantly, atleast for some participants. Different levels of accuracy are obtained for P4 and P5 (0.55 AUCROC vs 0.882 AUCROC), although both are using registration embeddings that are 20 weeks old. This shows that there is significant individual (riding) behavior variability in the long term, which can impact the authentication provided by *ScooterID*. These observations demonstrate the necessity of periodic re-enrollment in *ScooterID* (i.e., collecting new riding video samples to update the registration embeddings) to guarantee accurate continuous authentication. *ScooterID* has an advantageous design in this regard as it allows easy and efficient re-enrollment enabled by the Siamese network design, which does not require re-training of the embedding generation model. The re-enrollment frequency can be dynamically determined based on a system- or user-defined accuracy threshold.

TABLE IV: Average AUCROC and EER values when using old registration embeddings.

Time Apart	**1 week**		**5 weeks**	**20 weeks**	
Participants	**P1**	**P2**	**P3**	**P4**	**P5**
AUCROC	0.980	0.887	0.458	0.550	0.882
EER	0.142	0.235	0.540	0.448	0.116

TABLE V: User Identification True Acceptance Rate of *ScooterID* with Varying Numbers of Enrollment Samples

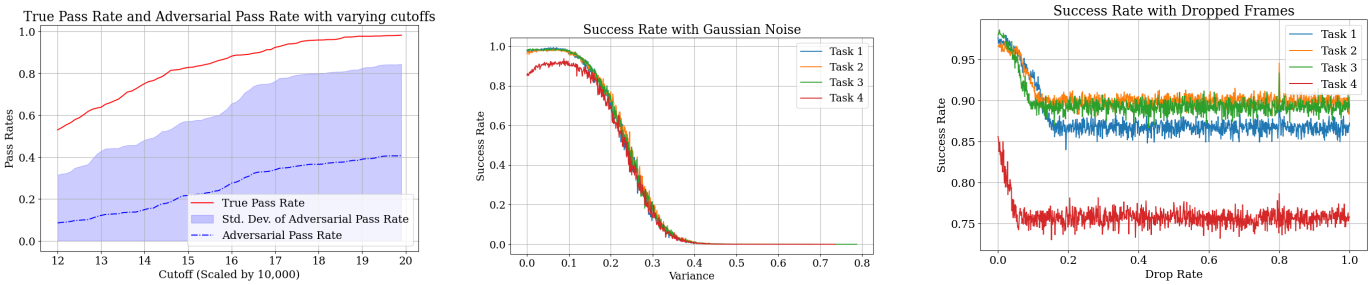
Pose Option	**Number of Enrollment Samples**				
	1	**5**	**10**	**20**	**40**
MoveNet	0.961	0.953	0.957	0.958	0.956
MediaPipe	0.791	0.771	0.797	0.785	0.78
Yolov7	0.941	0.941	0.959	0.945	0.949

H. User Identification Performance

The user identification task is needed when a mobility scooter is shared among group of users (e.g., family or in a senior community), and driving information is only used for personalized service and not for authentication. The test user is identified as one of a group of users who has the lowest embedding distance calculated by *ScooterID*. The group size is 5 in our tests. Table V shows the TAR values of *ScooterID* when various enrolment sample sizes are used and different pose estimation options are deployed. Overall, *ScooterID* achieves high level of true acceptance rate with MoveNet and Yolov7, in the range of 0.941 to 0.961. However, MediaPipe is outperformed in these tests for the user identification task with TARs under 0.80. The results are obtained as an average of 1000 tests per model across the high performance variation of each model.

V. *ScooterID* Robustness

Next, we evaluate the robustness of *ScooterID* against adversarial threats. We specifically focus on two adversarial goals: (i) getting authenticated as a legitimate user, and (ii) preventing legitimate users from being authenticated. For accomplishing the first goal, we assume that the adversary attempts to carry out a *mimicry attack*, which involves generation of a (riding) video to mimic legitimate users. For the latter goal, the adversary aims to carry out a *denial-of-service attack*, which involves video manipulation to prevent legitimate users from



(a) Generative videos (b) Perturbed keypoints (c) Missing frames

Fig. 12: Authentication success rates of *ScooterID* under adversarial conditions.

getting authenticated. We measure *ScooterID*'s robustness under these two types of attack scenarios using the metric of *pass rate* or *success rate*, which denotes the proportion of samples identified as belonging to an enrolled legitimate user in the system.

Mimicry Attacks: For these attacks, we assume that the adversary has a few images of legitimate users (riding the scooter) and can query the *ScooterID* model with engineered/doctored videos of legitimate users, generated using open-source or proprietary generative AI tools such as Gen-2 [58] and Stable Diffusion [59] in order to be authenticated as a legitimate user in the system. For our attack experiments, we use one randomly sampled video frame of a legitimate user driving a mobility scooter as the starting frame for the prediction, along with a text description of the intended driving behavior to generate a 16-second video using Gen-2 [58]. In total, we generate 21 such fake videos, test the average video segment pass rate varying the classification cut-offs and compare it with the pass rate of real video segments of the same legitimate user. Figure 12a shows that the generated fake video segments are rejected with a probability of about 60% at the distance cut-off achieving the EER, whereas the real video segments from the registered user are rejected with a probability of only about 2%. This shows that overall *ScooterID* is reasonably resistant to mimicry attacks using videos created by state-of-the-art generative models. The false pass rate may be improved by incorporating of robustness against diffusion-model-based mimicry attacks into the model design. We leave it to the future work.

Denial of Service Attacks (Videos): In this attack scenario, we assume that the adversary who intends to reduce the efficacy of *ScooterID* may have access to the input videos. The adversary is assumed to be capable of: (i) replacing the original video (file) with a modified video, and/or (ii) injecting adversarial perturbations into the live video feed deployment. Given that identification/authentication is real-time and continuous in nature, the adversary may choose to adopt standard pixel-level or temporal perturbations to the legitimate user videos. We attempt to mimic the adversary by applying *Gaussian noise* and *adversarial snow* to videos related to four different tasks encompassing track driving, road driving, off-road driving, and acceleration testing. The adversarial samples are then tested using *ScooterID* and are found to have minimal impact on authentication results, with the adversarial modifications leading to a sub-1% average decrement in true positive rate (success rate). The finding shows that *ScooterID* is robust against standard noise perturbations to the input videos.

Denial of Service (Keypoints): Here the adversary may have access to the keypoint data, rather than video data. To mimic the adversary, we simulate both the addition of Gaussian noise directly to all the keypoint coordinates, and randomly removing some frames' keypoint data (frame dropping). We replace keypoint coordinates of dropped frames with those from the most recent remaining frame, so that the amount of temporal information in coordinates data is reduced while the sequence length remains the same. Figure 12b shows the success rate of authentication, when Gaussian noise with different variances is directly added to the coordinates data. We note that *ScooterID* is robust to Gaussian noise up to 0.1 variance with 0 mean on data normalized between 0 and 1. Figure 12c shows that *ScooterID* demonstrates high robustness to dropped keypoints coordinates with success rates in between 75%-90% regardless of frame drop rates.

VI. Discussion and Limitations

Impact of Environmental Conditions. Due to limitations in the permitted testing environment, *ScooterID* has not been tested in all possible environments that mobility scooter riders may encounter, for example in low-light, or foggy conditions or on very rugged landscapes. In these cases, *ScooterID*'s performance is associated with the robustness of the human pose estimation models that we apply. Although MediaPipe, Yolov7, and MoveNet models were not trained or tested with data from all possible environmental conditions, there are robust transformer-based human pose estimation models (e.g., [60]–[62]) which have been tested using datasets with diverse backgrounds and environmental conditions such as MPI-INF-3DHP [63]. Applying these human pose estimation models in *ScooterID* could potentially improve the robustness of the system in diverse backgrounds and environmental conditions, but may downgrade the efficiency when *ScooterID* is run on resource constrained devices. Also, as mobility scooters cannot be safely driven on muddy or off-road terrains, we do not consider riding on such terrains in our tests. Scenarios to consider for testing our scheme's robustness would include sloped pavement and transitions from sidewalk-road at intersections, both of which we have collected as part of our dataset.

Completeness of Keypoints in Frame. In certain riding postures, not all keypoints will be in the frame of the camera, leading to several underdetermined key point locations. This concern is especially common with wrist keypoints, for example, when riders recover from stroke and only use one hand to hold the handle. We have included such driving patterns in the training dataset to ensure model performance despite missing keypoints. To address the longer absence of keypoints, *ScooterID* interpolates linearly from prior keypoint data to predict the current keypoint location. This interpolation procedure will likewise occur for cases where keypoints are occluded, such as if a user reaches into their pocket. However, with substantial and consistent omissions, it can lead to a degradation in performance due to the interpolation error. This issue can be addressed by employing a camera with a smaller focal length for a wider camera angle. As pose detection models have been trained to varying focal lengths and have normalized output, this will not cause issues in the *ScooterID* pipeline.

Camera Position and Angle Uncertainty. During riding, it is possible for camera to drift in position or angle. In addition, when users need to charge the camera and repeatedly remove it from the mount, we anticipate that in some cases the user may not properly secure the camera. For slight shifts that lead to occasional out-of-frame keypoints, the same interpolation procedure described above will estimate missing data. For significant drifts, a function can be added to the *ScooterID* system which can alert the user that the majority of keypoints are no longer in the frame and cease identification/authentication temporarily until the matter is addressed.

Privacy Concerns. Despite our intent not to capture facial data through the camera, this is not always avoidable. However, *ScooterID* is designed to process data captured by the camera immediately on a local device such as a Raspberry Pi and convert them into keypoint coordinate data. All stored data on local devices is either the keypoint coordinate data or the embedding vectors extracted from the posture data. The accidentally captured face information from riders getting on or off the mobility scooters or adjusting their sitting positions will not be saved. Thus, even with full access to the models employed and data stored, an attacker would be unable to reconstruct users' facial information. Obtaining the originally captured videos requires physical access to the camera or local computing devices, which is not practically feasible for the adversary [64].

Re-enrollment after Prolonged Periods. Over a prolonged period of time, it is possible that patient movement and posture patterns will change. However, for frequent users of *ScooterID*, after we have performed authentication, we can sample a small set of embeddings after each ride to use to refresh the existing enrollment embeddings. In doing so, we can ensure that, over a shorter period, the enrollment samples are completely refreshed without any intervention by the user. If the user experiences a sudden shift in riding pattern or has a prolonged period without using the mobility scooter, the user can opt to re-enroll.

VII. Conclusion

This paper proposes a new deep learning based continuous user identification and authentication framework for mobility scooter riders, called *ScooterID*. *ScooterID* utilizes only videos of users' upperbody movement while riding the mobility scooters to create a rider embedding and performs user identification/authentication by checking the distance between the registered embedding(s) and the one most recently computed (for the identification/authentication task). Our proposed deep learning model leverages spatio-temporal graph convolutions before a hierarchical encoding structure to produce embeddings and is trained with a Triplet Metric Loss function. Our comprehensive experimental results, using real mobility scooter riders' data, show that *ScooterID* achieves high levels of accuracy and efficiency, and demonstrates significant advantages compared to other architectures.

VIII. Acknowledgement

This research was supported by NSF awards #20508266, #2318671 and #2318672, and Google exploreCSR Summer Research 2023.

References

[1] R. L. Abduljabbar, S. Liyanage, and H. Dia, "The role of micro-mobility in shaping sustainable cities: A systematic literature review," *Transportation research part D: transport and environment*, vol. 92, p. 102734, 2021.

[2] D. J. Reck and K. W. Axhausen, "Who uses shared micro-mobility services? empirical evidence from zurich, switzerland," *Transportation Research Part D: Transport and Environment*, vol. 94, p. 102803, 2021.

[3] M. Mobility. How to prevent mobility scooter theft. https://marcsmobility.com/blog/how-prevent-mobility-scooter-theft.

[4] G. M. POLICE. Youths pictured riding mobility scooter after it was stolen from 85-year-old woman in salford. https://www.itv.com/news/granada/2023-10-19/woman-left-stranded-after-mobility-scooter-stolen-and-damaged-by-youths.

[5] B. of Transportation Statistics. Travel patterns of american adults with disabilities. https://www.bts.gov/travel-patterns-with-disabilities.

[6] Accessmap, a project to enable accessible, safe sidewalk trip planning for people with limited mobility. https://www.accessmap.io.

[7] A. R. Elshenaway and S. K. Guirguis, "Adaptive thresholds of eeg brain signals for iot devices authentication," *IEEE Access*, vol. 9, pp. 100 294–100 307, 2021.

[8] Y. Liang, S. Samtani, B. Guo, and Z. Yu, "Behavioral biometrics for continuous authentication in the internet-of-things era: An artificial intelligence perspective," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 9128–9143, 2020.

[9] S. Andrew, S. Watson, T. Oh, and G. W. Tigwell, "A review of literature on accessibility and authentication techniques," in *Proceedings of the 22nd International ACM SIGACCESS Conference on Computers and Accessibility*, 2020, pp. 1–4.

[10] R. Kumar, V. V. Phoha, and A. Serwadda, "Continuous authentication of smartphone users by fusing typing, swiping, and phone movement patterns," in *2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, 2016, pp. 1–8.

[11] S. Fritz and M. Lusardi, "White paper:"walking speed: the sixth vital sign"," *Journal of geriatric physical therapy*, vol. 32, no. 2, pp. 2–5, 2009.

[12] M. H. Khan, M. S. Farid, and M. Grzegorzek, "Vision-based approaches towards person identification using gait," *Computer Science Review*, vol. 42, p. 100432, 2021.

[13] D. Gafurov, "Performance and security analysis of gait-based user authentication," 2008.

[14] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2d human pose estimation: New benchmark and state of the art analysis," in *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, 2014, pp. 3686–3693.

[15] N. Li, X. Zhao, and C. Ma. (2020) Jointsgate: Gait recognition based on graph convolutional networks and joints relationship pyramid mapping. [Online]. Available: https://arxiv.org/pdf/2005.08625.pdf

[16] J. M. Kim, G. Choi, and S. Pan, "User identification system based on 2d cqt spectrogram of emg with adaptive frequency resolution adjustment," *Scientific Reports*, vol. 14, no. 1, p. 1340, 2024.

[17] S. Schneegass, Y. Oualil, and A. Bulling, "Skullconduct: Biometric user identification on eyewear computers using bone conduction through the skull," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 2016, pp. 1379–1384.

[18] Z. Sitová, J. Šeděnka, Q. Yang, G. Peng, G. Zhou, P. Gasti, and K. S. Balagani, "Hmog: New behavioral biometric features for continuous authentication of smartphone users," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 5, pp. 877–892, 2016.

[19] M. E. ul Haq, M. Awais Azam, U. Naeem, Y. Amin, and J. Loo, "Continuous authentication of smartphone users based on activity pattern recognition using passive mobile sensing," *Journal of Network and Computer Applications*, vol. 109, pp. 24–35, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804518300717

[20] H. Saevanee, N. Clarke, S. Furnell, and V. Biscione, "Continuous user authentication using multi-modal biometrics," *Computers Security*, vol. 53, pp. 234–246, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404815000875

[21] M. Lee, J. Ryu, and I. Youn, "Biometric personal identification based on gait analysis using surface emg signals," in *2017 2nd IEEE International Conference on Computational Intelligence and Applications (ICCIA)*. IEEE, 2017, pp. 318–321.

[22] M. Sivasamy, V. Sastry, and N. Gopalan, "Vrcauth: Continuous authentication of users in virtual reality environment using head-movement," in *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, 2020, pp. 518–523.

[23] Y. Zhang, W. Hu, W. Xu, C. T. Chou, and J. Hu, "Continuous authentication using eye movement response of implicit visual stimuli," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, pp. 1–22, 01 2018.

[24] S. Mondal and P. Bours, "Continuous authentication using mouse dynamics," in *2013 International Conference of the BIOSIG Special Interest Group (BIOSIG)*, 2013, pp. 1–12.

[25] A. Bhalla, I. Sluganovic, K. Krawiecka, and I. Martinovic, "Movear: Continuous biometric authentication for augmented reality headsets," in *Proceedings of the 7th ACM on Cyber-Physical System Security Workshop*, ser. CPSS '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 41–52. [Online]. Available: https://doi.org/10.1145/3457339.3457983

[26] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[27] G. Le Lan and V. Frey, "Securing smartphone handwritten pin codes with recurrent neural networks," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 2612–2616.

[28] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, and J. Ortega-Garcia, "Biotouchpass2: Touchscreen password biometrics using time-aligned recurrent neural networks," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2616–2628, 2020.

[29] H. Coskun, D. J. Tan, S. Conjeti, N. Navab, and F. Tombari, "Human motion analysis with deep metric learning," 2018.

[30] D. Shi, D. Tao, J. Wang, M. Yao, Z. Wang, H. Chen, and S. Helal, "Fine-grained and context-aware behavioral biometrics for pattern lock on smartphones," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 1, pp. 1–30, 2021.

[31] S. Lai, L. Jin, Y. Zhu, Z. Li, and L. Lin, "Synsig2vec: Forgery-free learning of dynamic signature representations by sigma lognormal-based synthesis and 1d cnn," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 10, pp. 6472–6485, 2021.

[32] M. Cardaioli *et al.*, "Privacy-friendly de-authentication with blufade: Blurred face detection," in *2022 IEEE International Conference on Pervasive Comp. and Comm. (PerCom)*, Pisa, Italy, 2022, pp. 197–206.

[33] H. Fereidooni, J. König, P. Rieger, M. Chilese, B. Gökbakan, M. Finke, A. Dmitrienko, and A.-R. Sadeghi, "Authentisense: A scalable behavioral biometrics authentication scheme using few-shot learning for mobile platforms," 2023.

[34] A. Sepas-Moghaddam and A. Etemad, "Deep gait recognition: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 1, pp. 264–284, 2022.

[35] A. Nambiar, A. Bernardino, and J. C. Nascimento, "Gait-based person re-identification: A survey," *ACM Computing Surveys (CSUR)*, vol. 52, no. 2, pp. 1–34, 2019.

[36] K. Hu, Z. Wang, S. Mei, K. A. Ehgoetz Martens, T. Yao, S. J. G. Lewis, and D. D. Feng, "Vision-based freezing of gait detection with anatomic directed graph representation," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 4, pp. 1215–1225, 2020.

[37] A. Sabo, S. Mehdizadeh, A. Iaboni, and B. Taati, "Estimating parkinsonism severity in natural gait videos of older adults with dementia," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 5, pp. 2288–2298, 2022.

[38] E. Vendrow and J. Vendrow, "Realistic face reconstruction from deep embeddings," in *NeurIPS 2021 Workshop Privacy in Machine Learning*, 2021.

[39] Y. Song and Z. Cai, "Integrating handcrafted features with deep representations for smartphone authentication," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 6, no. 1, pp. 1–27, 2022.

[40] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee, W.-T. Chang, W. Hua, M. Georg, and M. Grundmann, "Mediapipe: A framework for building perception pipelines," 2019.

[41] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," *arXiv preprint arXiv:2207.02696*, 2022.

[42] T. Hub. Movenet singlepose lightning. https://tfhub.dev/google/movenet/singlepose/lightning/4.

[43] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1701–1708.

[44] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2017.

[45] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline," 2016.

[46] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification." *Journal of machine learning research*, vol. 10, no. 2, 2009.

[47] B. Harwood, V. Kumar BG, G. Carneiro, I. Reid, and T. Drummond, "Smart mining for deep metric learning," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2821–2829.

[48] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

[49] Tewiky. Tewiky tw-05 webcam. https://www.amazon.com/Microphone-Rotatable-Computer-Conferencing-Streaming/dp/B08VJ25PL1.

[50] S. Vhaduri, S. V. Dibbo, and W. Cheung, "Hiauth: A hierarchical implicit authentication system for iot wearables using multiple biometrics," *IEEE Access*, vol. 9, pp. 116 395–116 406, 2021.

[51] S. Brody, U. Alon, and E. Yahav, "How attentive are graph attention networks?" 2022.

[52] J. Wu, J. Konrad, and P. Ishwar, "Dynamic time warping for gesture-based user identification and authentication with kinect," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 2371–2375.

[53] H. Kong, L. Lu, J. Yu, Y. Chen, and F. Tang, "Continuous authentication through finger gesture interaction for smart homes using wifi," *IEEE Transactions on Mobile Computing*, vol. 20, no. 11, pp. 3148–3162, 2020.

[54] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[55] M. Hu, K. Zhang, R. You, and B. Tu, "Relative attention-based one-class adversarial autoencoder for continuous authentication of smartphone users," *arXiv preprint arXiv:2210.16819*, 2022.

[56] G. Coral. Movenet single pose lightning tflite model. https://raw.githubusercontent.com/google-coral/test_data/master/movenet_single_pose_lightning_ptq_edgetpu.tflite.

[57] ——. Edge tpu usb accelerator wa1. https://coral.ai/products/accelerator/#description.

[58] "RunwayML Gen 2," https://research.runwayml.com/gen2, 2023, [Online; accessed 19-Nov-2023].

[59] "Stability AI," https://stability.ai/, 2023, [Online; accessed 19-Nov-2023].

[60] W. Li, H. Liu, H. Tang, P. Wang, and L. Van Gool, "Mhformer: Multi-hypothesis transformer for 3d human pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 13 147–13 156.

[61] H. Liu, J.-Y. He, Z.-Q. Cheng, W. Xiang, Q. Yang, W. Chai, G. Wang, X. Bao, B. Luo, Y. Geng *et al.*, "Posynda: Multi-hypothesis pose synthesis domain adaptation for robust 3d human pose estimation," in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, pp. 5542–5551.

[62] W. Li, M. Liu, H. Liu, P. Wang, J. Cai, and N. Sebe, "Hourglass tokenizer for efficient transformer-based 3d human pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 604–613.

[63] D. Mehta, H. Rhodin, D. Casas, P. Fua, O. Sotnychenko, W. Xu, and C. Theobalt, "Monocular 3d human pose estimation in the wild using improved cnn supervision," in *2017 international conference on 3D vision (3DV)*. IEEE, 2017, pp. 506–516.

[64] H. Wei, H. Tang, X. Jia, Z. Wang, H. Yu, Z. Li, S. Satoh, L. Van Gool, and Z. Wang, "Physical adversarial attack meets computer vision: A decade survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.