Uncertainty Quantification to Enhance Probabilistic Fusion Based User Identification Using Smartphones

Rouhollah Ahmadian, Mehdi Ghatee, Johan Wahlström, Hadi Zare

***Abstract*—User identification through smartphones and wearable sensors holds promise but faces challenges from similarity and variability in user activities. Visualization of smartphone acceleration signals revealed users' signals exhibit high similarity, as activities share a common underlying structure. For example, walking elicits a repeated general pattern. Therefore, user identification relies on subtle distinguishing factors in fine activity details. At times, patterns are near-indistinguishable between users. To address this, we developed a method leveraging the assumption that prediction uncertainty increases for non-separable samples. The input data is divided into subsequences, each independently predicted by a convolutional neural network. Predictions are fused through a weighted averaging scheme, where weights quantify prediction uncertainty using the Monte Carlo Dropout method. Through experiments on five real-world datasets, the study demonstrates improved performance in identifying users across a range of activities compared to existing methods. It was also directly compared to state-of-the-art methods using two well-known datasets, improving accuracy by 1.29% in one case and 7.98% in the other. These findings validate the effectiveness of the new approach for continuous user identification, even when faced with unpredictable user behavior.**

***Index Terms*—Uncertainty Quantification, Monte-Carlo Dropout, Probabilistic Fusion, User Identification, Biometric Recognition, Signal Processing, Deep Learning**

I. Introduction

USER identification is the process of identifying a person within a group of people. Previously, users were identified using simple methods such as ID cards, passwords, and PINs. Then more advanced methods such as fingerprint, face, iris, and finger vein pattern recognition were used [1], [2]. However, these approaches relied on information that needed to be remembered or carried by the user, making them susceptible to hacking or theft [3]. Additionally, they did not provide continuous monitoring of users [4]. Furthermore, some of these methods compromised user privacy or could be exploited to enable deception [3], [5]. Modern user identification systems now leverage the embedded sensors in smartphones and wearable devices [6], [7]. The number of smartphones and wearable devices in use is growing significantly based on statistics [2]. These devices often contain highly sensitive personal information. Therefore, reliable automatic identification of authorized device users is extremely important.

User identification has widespread applications in authentication, access control, security monitoring, and systems that regulate entry and exit [8]. It can also enhance diverse areas such as intelligent transportation, healthcare, and usage-based insurance. Within transportation, identification prevents unauthorized individuals from assuming the identities of registered ride-share drivers [5], [9]. In healthcare, accurate recognition of device users is crucial given the sensitive personal data increasingly collected by wearable technologies - improper identification could jeopardize patient privacy and confidentiality [8]. Identification also allows for customized insurance policies in usage-based models, where continuous recognition of driver habits forms the basis for tailored coverage [1]. Smart authentication methods are a priority for major tech companies. Google's Smart Lock, introduced in 2014 for Android, is an influential example. It passively identifies users through on-body detection, frequently visited locations, paired devices, facial recognition, and voice matching [2]. When done correctly, identification benefits various systems by enhancing security, convenience, and user experience across numerous interactive applications.

Segmentation using a sliding window is a key preprocessing step commonly used in previous user identification research. Each segment is considered as an instance in the classification process. In this context, selecting the optimal window length is an important factor [6]. Signals collected from smartphones and wearables consist of a sequence of activities like walking and running. Our previous study divided activity patterns into two categories: primary and mixed [10]. A primary pattern represents the shortest meaningful example of a single action, such as brushing teeth. A mixed pattern incorporates multiple primary patterns. To capture a mixed pattern, the sliding window length must be long enough to include several primary patterns. In the realm of user identification, an essential aspect of mixed patterns is the randomized order of primary patterns, as real-life activity sequences often exhibit randomness [10]. Enlarging the sliding window length has the effect of reducing similarity between instances produced by the sliding window. This decrease in similarity characterizes the phenomenon known as concept drift [10]. Specifically, Fig. 1 illustrates how

Manuscript received ....; revised .....; accepted .... The Associate Editor for this paper was .... (Corresponding author: Mehdi Ghatee.)

R. Ahmadian is a Ph.D. student with the Department of Computer Science, Amirkabir University of Technology, Tehran, 15875-4413, Iran (e-mail: rahmadian@aut.ac.ir).

M. Ghatee is a Full Professor with the Department of Computer Science, Amirkabir University of Technology, Tehran 15875-4413, Iran (e-mail: ghatee@aut.ac.ir).

J. Wahlström is an Assistant Professor with the Department of Computer Science, University of Exeter, Exeter, EX4 4QF, UK (e-mail: j.wahlstrom@exeter.ac.uk).

H. Zare is an Associate Professor with the Faculty of New Sciences and Technologies, University of Tehran, Tehran, Iran (e-mail: h.zare@ut.ac.ir).

the average dynamic time warping (DTW) distance between instances increases alongside expanded sliding window length. The growing average distance serves as an indicator of diminishing similarity.

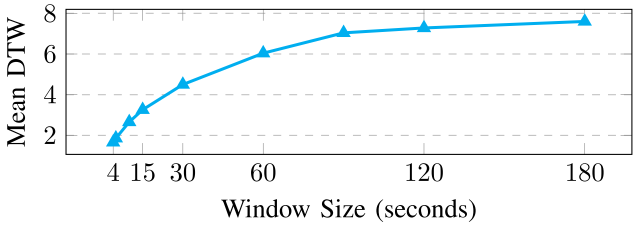

Fig. 1. The average DTW distances between instances

Human activity recognition focuses on identifying basic patterns, but identifying users based solely on these patterns presents challenges because the overall structure of a pattern is similar across users [11], [4]. The key to user identification lies in subtle differences found in finer details [10]. When the sliding window length is small, it is possible to produce samples that closely resemble different labeled classes. Sliding windows, as transitional functions, do not consistently preserve the inherent identity of the data and are not invariant to label information [10]. Therefore, labeling instances created via sliding windows using the labels from the original data is a simpler approach compared to alternatives. To illustrate this, Fig. 2 depicts instances transformed by t-SNE, generated with a 4-second sliding window. The instances are colored based on two perspectives: activity type and user identity. The figure highlights the difficulty in separating data based on user identity compared to activity. It indicates overlapping probability distributions of users, suggesting that the assigned labels for such samples may lack definitive confidence. Using smaller window sizes could potentially undermine the reliability of classification labels associated with the resulting sample data [10]. In conclusion, this research extends our previous work by proposing a solution to mitigate the impact of uncertainty in the final decision-making process.

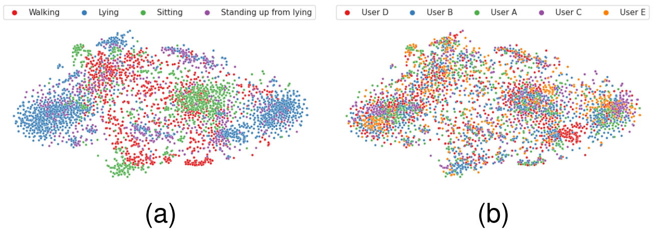

Fig. 2. t-SNE visualization of instances colored by activity type (a) and user identity (b)

This research aims to solve the problem of user identification reliability being undermined by high similarity between individuals' activity patterns. The hypothesis is that accounting for prediction uncertainty can enhance identification accuracy. The main differences between the proposed algorithm and existing methods are that it leverages subsequence modeling and uncertainty-aware fusion to account for similarities in sensor data patterns. The key contributions of this research are:

- Developing a subsequence-based modeling approach: 1) The input data is divided into overlapping fixed-length subsequences to capture temporal dependencies at a local level. 2) Each subsequence is predicted independently using a 2D CNN.
- Quantifying prediction uncertainty using Monte Carlo Dropout: 1) Dropout is applied after each layer of the CNN during test time. 2) This allows estimating prediction uncertainty from the variation in results from different dropout masks. 3) Uncertainty is quantified using the entropy in predictions.
- Uncertainty-aware weighted averaging fusion: 1) Predictions from the CNN are fused using a weighted averaging scheme. 2) Weights are determined by the inverse of the quantified prediction uncertainty. 3) This gives more importance to subsequences with lower uncertainty.
- Five publicly available datasets, [12], [13], [14], [15], [3], were used to comprehensively evaluate the model.
- Direct comparisons to two state-of-the-art models using benchmark datasets [15], [3] demonstrated improved accuracy of 1.29% over the first model and 7.98% over the second.

The paper is organized as follows: Section 2 reviews previous research on user identification, considering various sensors, data types, and methodologies. Section 3 delves into the theory of uncertainty quantification. Section 4 presents the proposed methodology for mitigating the impact of uncertainty, and Section 5 showcases the experimental results obtained from datasets.

II. Literature

Methods of verifying identity can be organized into three categories: knowledge-based, physiological biometrics, and behavioral patterns approaches [1]. Knowledge-based authentication directly requires users to provide information like passwords, PINs or graphical codes for confirmation [2], [4]. Physiological biometrics utilizes innate physical attribute and machine learning to discern users based on traits such as EEG, ECG, periocular, fingerprint, iris, or facial features [4], [16]. Behavioral patterns encompassing features such as gait, handwaving, keystroke, and touchscreen use are also examined [11], [17]. One advantage of behavioral patterns is its ability to verify passively [4]. Authentication system types can further be divided into users authentication, which evaluates whether an user is known or unknown, as well as user identification, which recognizes the specific person present [1].

Sensors within smartphones and wearable devices can be categorized into several groups, including environmental, positioning, healthcare-related, and motion sensors. Environmental sensors such as microphones, thermometers, and barometers provide contextual information about the user's surroundings that may aid in identification [18], [19]. Location sensors like GPS are also widely used for this purpose [4]. However, privacy is a potential concern with some of these approaches [5]. In addition, smartwatches can monitor health metrics such as heart rate, and electrocardiograms, enabling identification as well [20], [21]. Furthermore, motion sensors consisting of accelerometers, and gyroscopes are capable of capturing behavioral patterns involving activities such as walking, and jogging - techniques employed in previous works [4].

Previous research categorized motion-based user identification approaches into walking gait and body gesture recognition based on behavioral patterns [1]. Gait recognition analyzes walking patterns using smartphones and wearable sensors attached at different body locations to continuously monitor users [12]. One study used WiFi signals with integrated and weighted features to recognize patterns from gait and respiration [19]. Another investigated smartphone accelerometer data, formulating identification as an image classification task using spectrotemporal representations. A custom Convolutional Neural Network (CNN) model was evaluated on a large dataset [22]. More recently, a system was introduced applying a model to an activity gait dataset captured in real-world environments [23]. This system recognized subjects within uncontrolled contexts by analyzing gait patterns. Furthermore, significant authentication research focused on body gestures like various hand motions [1]. Some gestures examined for identity verification included typing, eating, and folding clothes [24]. For example, a hand-waving approach was presented for an unlocking system utilizing a smartwatch [17]. Another system recognized arm gestures, identity, and verification using inertial sensors in a custom wristband with neural network processing [25]. Additional work proposed a novel framework for identification using hand motion during walking [6]. It selected high-quality features through optimal evaluation and correlation-based selection algorithms. Again, unique lip biometrics on smartphones is a secure user authentication method [26].

Data preprocessing plays a vital role in extracting relevant information from raw data [1]. It involves cleaning incomplete or noisy samples, segmentation, and data augmentation [27]. Techniques for filtering noise include signal smoothing and normalization [11], [18]. Moreover, a data augmentation module is proposed to enhance user authentication performance [27]. Segmentation aims to delineate meaningful sections from sequential data. Both rule-based and dynamic segmentation approaches have been explored [28], [5]. Rule-based techniques employ fixed thresholds or windows but are sensitive to inputs. Therefore, dynamic segmentation adapts to split streams adaptively by recognizing patterns, such as touch gestures, to isolate significant actions [28]. Feature engineering endeavors to capture user behavior by extracting meaningful attributes from data [28]. These attributes encompass statistical, spectral, and temporal features, which capture overall and dynamic patterns from macro and signal perspectives [11], [3]. Recent studies have utilized randomized CNNs incorporating statistical features like entropy in accelerometer data analysis [7]. Dimensionality reduction and feature selection techniques can be applied to construct optimized subsets [8]. However, deep learning-based methods have gained attention as they can automatically learn representations, addressing the limitations of manually specifying attributes [18], [29].

Based on the classification algorithms used, behavior-based user identification systems can be categorized into two main types: shallow and deep learning models [1]. Shallow approaches rely on hand-crafted features and supervised classification algorithms, which require time-consuming feature engineering and lack robustness to changes in user behavior [1]. Supervised algorithms are commonly used include K-Nearest Neighbours (KNN), Support Vector Machines (SVM), One-Class SVM (OSVM), Multi Layer Perceptrons (MLP), Gaussian Mixture Model (GMM), and Random Forest (RF) [30], [31].

Deep learning methods offer promising improvements in performance. They encompass various techniques such as CNN [2], Recurrent Neural Networks (RNN) [15], Generative Adversarial Networks (GAN) [32], and Attention Mechanism (AM) [3]. CNNs are widely used for user authentication [22]. GANs are utilized to generate realistic biometric data [5]. RNNs and their variants (LSTM, GRU, ConvLSTM) excel in processing sequential behavior data effectively [3]. AM, which includes self-attention and squeeze-and-excitation network, automatically assigns higher weights to informative segments of feature maps [3].

Furthermore, previous research has explored the utilization of decision fusion techniques for user identification [33]. These techniques can be categorized into two main approaches: ensemble learning and probabilistic fusion [8], [34]. In ensemble learning, multiple classifiers' predictions are combined to predict an instance [3]. On the other hand, probabilistic fusion involves dividing an instance into subsequences and generating a prediction for each subsequence [9]. These individual predictions are then combined using a fusion function. Notably, averaging and majority voting fusion functions have been commonly employed in previous studies to aggregate multiple initial decisions [6], [10]. Consequently, these learning approaches display remarkable potential in enhancing the accuracy and security of user identification systems.

While previous work has made progress with deep learning and fusion, shortcomings remain in addressing inherent behavioral pattern similarities. Most techniques treat predictions equally without considering uncertainty, undermining reliability for highly comparable patterns. Previous probabilistic fusion primarily used simple averaging/voting, without exploring robust combination methods. This study proposes a novel uncertainty-aware fusion approach leveraging deep learning and uncertainty quantification. By weighting predictions based on uncertainty, the goal is to enhance identification performance when handling ambiguities from similarities. In addition, as motion data from smartphones and wearables is more accessible than healthcare data, this research focuses on leveraging various motion sensors to enable continuous identification.

III. Preliminaries

According to a comprehensive survey study on estimating uncertainty in deep neural networks [35], different approaches have been identified, including single deterministic, Bayesian, ensemble, and test-time augmentation methods. Single deterministic methods involve using a deterministic network and performing a single forward pass, but they may lack robustness due to reliance on a single opinion. Ensemble methods combine predictions from multiple networks but require significant computational resources. Test-time augmentation methods generate multiple predictions by augmenting input data, although this approach can lead to incorrect predictions. Bayesian methods, on the other hand, utilize Bayesian learning in combination with DNNs to estimate uncertainty, offering accurate and expressive posteriors. Despite challenges in specifying priors and marginalizing parameters, Bayesian methods show promise for robust uncertainty estimation in DNNs. Hence, the focus of this research is specifically on one of the Bayesian methods.

In Bayesian modeling, the uncertainty of a classification model arises from two sources: data uncertainty (*aleatoric*) and model uncertainty (*epistemic*). Data uncertainty refers to the inherent uncertainty in the data, while model uncertainty relates to uncertainty in the model's parameters [36]. To explore and quantify these uncertainties, we employ a Bayesian framework. In Bayesian modeling, we consider a joint distribution, denoted as $p(x, y)$, over the input features x and labels y. We want to estimate the predictive uncertainty, denoted as $p(y = \omega_c|x^*, \mathcal{D})$, of the model given a new input x^* and the training dataset $\mathcal{D} = \{x_j, y_j\}_{j=1}^{N} \sim p(x, y)$. To compute the predictive uncertainty, we need to integrate over the model's parameters θ and the data [36]:

$$p(y = \omega_c|x^*, \mathcal{D}) = \int p(y = \omega_c|x^*, \theta)p(\theta|\mathcal{D})\ d\theta \tag{1}$$

However, obtaining the true posterior distribution $p(\theta|\mathcal{D})$ using Bayes' rule is often intractable due to the complexity of the models and the high-dimensional parameter space. To overcome this challenge, variational approximations are employed. An approximating distribution, denoted as $q(\theta)$, is obtained, which can be more easily worked with. The goal is to find an approximation that is close to the true posterior distribution $p(\theta|\mathcal{D})$. This allows estimation of the predictive uncertainty by integrating the model's predictions, $p(y = \omega_c|x^*, \theta)$, over the approximated posterior distribution $q(\theta)$. However, even with the variational approximation, computing the exact integral for neural networks remains computationally infeasible. To address this, Monte Carlo dropout [37] can be employed. During evaluation, the Dropout layer is kept active, allowing for multiple forward passes through the network. Each forward pass corresponds to a different set of dropped out neurons, leading to diverse predictions. Therefore, we are indeed able to approximate the predictive distribution. Next, the uncertainty associated with the predictions can now be quantified. One common approach is to calculate the entropy of the expected softmax output [35]:

$$\mathbb{UM}(x, \theta) = \mathcal{H}(\mathbb{E}_{\theta\sim p(\theta)}[p(y = \omega_c|x, \theta)]) \tag{2}$$

$$\mathcal{H}(p) = -\sum_{i=1}^{c} p[i]\ log_2(p[i])$$

where c is the number of classes, $\mathcal{H}$ denotes entropy, and $\mathbb{E}_{\theta\sim p(\theta)}[p(y|x, \theta)]$ represents the expected softmax output. The entropy measures the uncertainty in the predicted class probabilities. Higher entropy indicates higher uncertainty, while lower entropy suggests more confident predictions [35].

IV. Methodology

This section introduces a novel architecture for user identification as shown in Fig. 3. It contains an inner classifier which can be any classification algorithm. The inner classifier is first trained on the input data. During inference, the input is divided into multiple grains or segments. Each grain is independently predicted by the inner classifier. The prediction results are then combined using a fusion function to generate the final decision. Our previous research found averaging predictions effective by reducing variance while maintaining bias to enhance performance [10]. In this study, the fusion function incorporates prediction uncertainty - it quantifies the confidence of each grain prediction and factors this into the fusion process. The following subsections explain each component in detail.

A. Preprocessing Module

This module extracted inertial signals from device sensors. To ensure data quality, outliers are initially removed, and missing values are replaced with averages during a cleaning step. The dataset is then divided into three parts, namely training, validation, and testing, with proportions of 50%, 20%, and 30%, respectively. Also, the data is normalized using the Gaussian transformation, where $x' = \frac{x-\bar{x}}{\sigma}$. Here, $\bar{x}$ represents the average and σ represents the standard deviation, both of which are obtained from the training data. Subsequently, the segmentation process takes place, utilizing hopping windows. These windows possess a fixed length and move through the data with a specified step size. As the window progresses, it generates distinct instances. It is worth noting that if the hopping window length is denoted as w and the data contains t signals, the resulting instances will have dimensions of $t \times w$. Furthermore, the quantity of generated instances, represented as n, is not only determined by the window length but also influenced by the window overlap, indicated by r. So a decreased window length (w) and an increased window overlap (r) will lead to a higher number of generated instances (n). To facilitate the learning, the training instances are shuffled. However, it is essential to maintain the original order of instances during the fusion. Therefore, the validation and testing data are not shuffled.

B. Training Inner Classifier

The experiments showed that convolutional neural networks performed best as the inner classifier. CNNs excel at directly learning representations from the data distribution, unlike methods that require engineering features manually. Fig. 4 details the specific CNN model architecture employed as the inner classifier. To achieve this architecture, it began simply and was carefully scaled by validation-guided experimentation, adding layers and neurons until achieving suitable accuracy. Model parameters resulted from trial-and-error tuning using validation data. The training process aims to optimize the inner

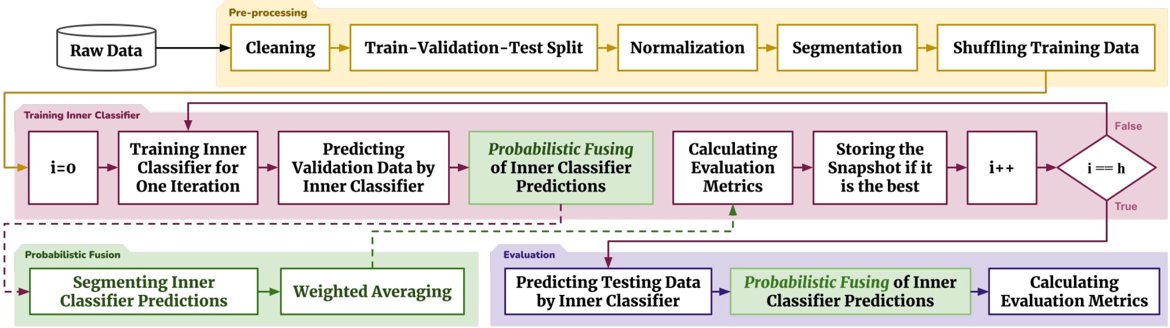
Raw Data
Pre-processing
Cleaning
Train-Validation-Test Split
Normalization
Segmentation
Shuffling Training Data
Training Inner Classifier
i=0
Training Inner Classifier for One Iteration
Predicting Validation Data by Inner Classifier
Probabilistic Fusing of Inner Classifier Predictions
Calculating Evaluation Metrics
Storing the Snapshot if it is the best
i++
i == h
False
True
Probabilistic Fusion
Segmenting Inner Classifier Predictions
Weighted Averaging
Evaluation
Predicting Testing Data by Inner Classifier
Probabilistic Fusing of Inner Classifier Predictions
Calculating Evaluation Metrics

Fig. 3. The architecture of the training and testing process of the proposed model.

classifier. Furthermore, our previous research showed that errors from the inner classifier's predictions are not directly correlated with errors from fused results [10]. Therefore, the inner classifier's errors can decrease in an iteration while the error after fusion increases, and vice versa. To this end, we proved that monitoring the inner classifier based on the error after fusion is essential during the training process [10].

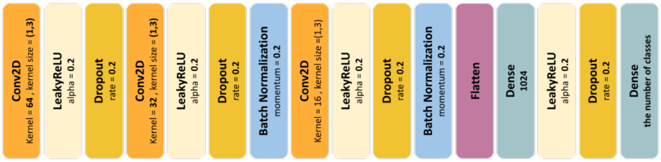

Fig. 4. The architecture of the CNN classifier.

Now, let's delve into how to train the inner classifier, building upon our previous work investigating neural network optimization procedures [10], [38]. Given that neural network training is an iterative process, let's assume it is repeated h times. In each iteration, initially the network undergoes training for a single iteration, employing training instances as input. Afterward, the model's performance is assessed by utilizing validation data. This evaluation involves feeding the validation data into the inner classifier. This produces predictions of shape $n \times c$, where n is the number of validation instances, generated in the segmentation step, covering c classes. To further analyze these predictions, they are partitioned using a fixed-length sliding window referred to as the *decision window*. This window has a length of s and zero overlap. Thus, the resulting dimensions are $o \times s \times c$, where $o = \frac{n}{s}$ indicates the number of decision windows. Within each decision window, the fusion method described in Section IV-C is applied to combine the s predictions into a shape of $o \times c$. This fused output is evaluated by the *Mean Penalized Brier Score* (MS_{PBS}) metric [38]:

$$MS_{PBS} = \frac{1}{o}\sum_{i=1}^{o} S_{PBS}(q^{(i)}, y^{(i)}) \tag{3}$$

$$S_{PBS}(q, y) = \left(\sum_{i=1}^{c}(y[i] - q[i])^2\right) + \begin{cases} \frac{c-1}{c} & q \in \xi \\ 0 & \text{otherwise} \end{cases}$$

where q represents predictions, y indicates ground-truth labels, and the term $q \in \xi$ implies that q is a wrong prediction (false-positives or false-negatives). Notably, our research [38] demonstrates the high effectiveness of this metric in evaluating classification models through both theoretical proofs and extensive experiments. The MS_{PBS} combines the characteristics of F-measures and Mean Square Error (MSE) by penalizing false-positive and false-negative predictions. Thus it enables more precise identification of optimal model checkpoints to maximize classification performance during training.

Finally, the current evaluation results obtained based on Eq. (3) are compared to the best previous iterations. For higher cases, the network weights are temporarily stored in a checkpoint. After full training, the final weights retrieved are those with minimum MS_{PBS}, optimizing the inner classifier. Comparing iteration scores and checkpoints, the best weights guide the model to maximum accuracy per the validation metric.

C. Probabilistic Fusion Function

To estimate the probability of a specific class label, ω_c, given a new input X segmented into n subsequences, we apply Theorem 1 and Assumption 1.

Theorem 1. *For any input instance $X = (x_1, \dots, x_s)$ we have:*

$$p(y = \omega_c|X) = \frac{\sum_{i=1}^{s} p(y = \omega_c|x_i)p(x_i)}{\sum_{i=1}^{s} p(x_i)}, \tag{4}$$

where, x_i represents the i^{th} generated subsequence of X.

Proof. By considering $p(\omega_c|X)$ as an alternative representation for $p(y = \omega_c|X)$, and treating subsequences x_i as mutually exclusive events. As we segment X into subsequences x_i:

$$p(\omega_c|X) = p(\omega_c| \cup_{i=1}^{s} x_i) \tag{5}$$

Using the definition of conditional probability:

$$p(\omega_c| \cup_{i=1}^{s} x_i) = \frac{p(\omega_c \cap \cup_{i=1}^{s} x_i)}{p(\cup_{i=1}^{s} x_i)} = \frac{p(\cup_{i=1}^{s}(\omega_c \cap x_i))}{p(\cup_{i=1}^{s} x_i)} \tag{6}$$

By additivity of probability:

$$\frac{p(\cup_{i=1}^{s}(\omega_c \cap x_i))}{p(\cup_{i=1}^{s} x_i)} = \frac{\sum_{i=1}^{s} p(\omega_c \cap x_i)}{\sum_{i=1}^{s} p(x_i)} \tag{7}$$

By definition of conditional probability, this equals:

$$\frac{\sum_{i=1}^{s} p(\omega_c \cap x_i)}{\sum_{i=1}^{s} p(x_i)} = \frac{\sum_{i=1}^{s} p(\omega_c|x_i)p(x_i)}{\sum_{i=1}^{s} p(x_i)} \tag{8}$$

□

Assumption 1. *To simplify, the quantified uncertainty of $p(y = \omega_c|x_i)$, denoted as α_i, is assumed as an alternative to $p(x_i)$:*

$$p(y = \omega_c|X) = \frac{\sum_{i=1}^{s} \alpha_i p(y = \omega_c|x_i)}{\sum_{i=1}^{s} \alpha_i}. \tag{9}$$

In the realm of out-of-distribution (OOD) instance detection, empirical findings demonstrate a strong correlation between uncertainty quantification and the data's position within the data distribution. Specifically, as the data point diverges further from the distribution, the level of uncertainty increases [39]. Leveraging this characteristic presents an opportunity to effectively identify OOD instances [35]. By assuming that the data adheres to a normal distribution, we can establish uncertainty quantification as an estimation for the prior probability.

The fusion function $\mathcal{WA} : R^{s\times c} \rightarrow R^c$, which takes the matrix dw representing the decision window as input, calculates a vector output for the decision window using a weighted average approach based on Eq. (9). The weights are determined by the quantified uncertainties α associated with the predictions from the inner classifier for each subsequence. Therefore:

$$\mathcal{WA}(dw) = \left[\frac{1}{\sum_{i=1}^{s} \alpha_i} \sum_{i=1}^{s} \alpha_i \; dw_{i,j} \right]_{j=1}^{c} \tag{10}$$

Here, $dw_{i,j}$ represents the prediction for the j^{th} class label in the i^{th} subsequence. To determine the quantified uncertainties, we utilize the Monte-Carlo Dropout approach and the uncertainty measure given by Eq. (2). To obtain α_i, with M predictions from x_i using the Monte-Carlo Dropout approach, Eq. (2) can be rewritten as:

$$\mathbb{UM}(x, \theta) = \mathcal{H}\left(\left\{ \frac{1}{M} \sum_{i=1}^{M} p_i(y = \omega_j|x, \theta_i) \right\}_{j=1}^{c} \right) \tag{11}$$

Since the quantified uncertainty is inversely related to relevance, we apply the inverse of $\mathbb{UM}$ to obtain α_i. Considering that the maximum value of $\mathbb{UM}$ is $\log_2(c)$, we define α_i based on a linear transformation of $\mathbb{UM}$ as:

$$\alpha_i = -\frac{1}{\log_2(c)} \mathbb{UM}(x_i, \theta) + 1 \tag{12}$$

D. Evaluation

During testing, the decision window technique segments data and the proposed fusion function merges probability vectors within each window. Standard performance metrics evaluate the model. Given the time-series nature of the data, h-block cross-validation is used for model selection and hyperparameter tuning. This partitions the data into unique training, validation, and test sets over multiple iterations [10]. The data was partitioned into 50% for training, 20% for validation, and 30% for testing. Let's consider an example where we have 100 minutes of signal data per class. During the initial iteration, we allocate the first 20 minutes from each class for validation, followed by 30 minutes for testing, leaving the remainder for training. As subsequent iterations unfold, we dynamically adjust the time allocations, dedicating 10 to 30 minutes for validation, followed by a dedicated 30-minute duration for testing, and utilizing the remaining time for training. Through this iterative process, multiple experiments are conducted, which yield the mean and variance.

V. Experimental Result

In this section, the proposed model is analyzed from different perspectives. Several datasets have been used for this purpose. Table I provides general informations of the datasets. The proposed model analysis utilizes UIFW, CLD, and HOP datasets, offering relevant evaluation data. Additionally, benchmarking with DB2 and HAR datasets, utilized in state-of-the-art methods [3], [15].

TABLE I
Datasets

Dataset	Ref.	c	Sensors
UIFW	[12]	13	3-axis Accelerometer
CLD	[13]	5	3-axis Coordinates
HOP	[14]	12	3-axis Accelerometer
DB2	[15]	20	3-axis Accelerometer, 3-axis Gyroscope
HAR	[3]	30	3-axis Accelerometer, 3-axis Gyroscope

1) **UIFW:** An Android smartphone captured accelerometer data in the chest pocket of 22 participants walking in natural environments. Analysis focuses on 13 individuals due to inadequate data availability. 2) **CLD:** Localization data was recorded from 5 individuals wearing tags on their ankles, belt, and chest. 3) **HOP:** Motion data from 12 healthy older adults (66-86 years) was captured using a sternum-level wearable sensor. The data exhibits sparsity and noise due to the usage of passive sensors (batteryless RFID tags). 4) **Dataset #2 (DB2):** Gait data from 20 subjects segmented with a sliding window approach (window length: 128, no overlap). Dataset split into 70% training and 30% testing sets. 5) **UCI-HAR (HAR):** Accelerometer and gyroscope data from 30 volunteers (19-48 years) preprocessed with low-pass filters. Segmented into fixed-width sliding windows (2.56s, 50% overlap). Dataset divided into 70% training and 30% testing sets.

A. Sensitivity Analysis

a) Inner Classifier: Several algorithms were analyzed to select the inner classifier. Grid search optimized hyperparameters, and models evaluated on varied window lengths. The results, presented in Fig. 5, depict the highest achieved F1-Scores for each model. Notably, the result shows that the CNN model emerges as the superior choice. It is worth noting that fusion is not used at this stage.

b) Sensors: The proposed model underwent analysis to determine how it would perform with different sensor combinations. The outcomes are presented in Fig. 6, showing the highest scores achieved. Significantly, the figure indicates optimal functioning occurred when all available sensor inputs were utilized.

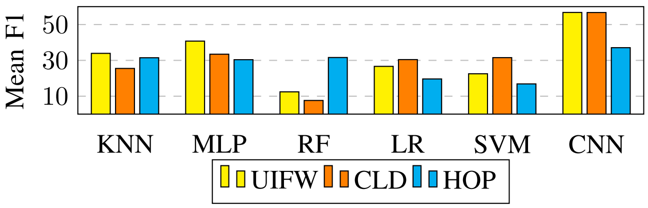

Fig. 5. Comparing F1-Scores across various classifiers.

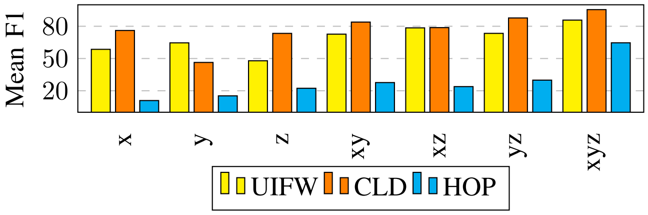

Fig. 6. F1-Scores of the proposed model in terms of various sensors.

c) Hyperparameters: Table II summarizes the optimal hyperparameters identified through grid search for the proposed model applied to each dataset. The search space considered window lengths (w) ranging from 3 to 120 seconds, window overlaps (r) of $0 - 75\%$, and decision window sizes (S) varying from 3 to 180 seconds. Also, the number of predictions in each decision window (s) is shown in parentheses. The learning rate (lr), number of epochs ($epoch$), and batch size (bs) were evaluated across values of 10^{-1} to 10^{-5}, 20 to 500, 32 to 512, respectively.

TABLE II
OPTIMAL HYPERPARAMETERS

Data	w	$r(\%)$	S (s)	lr	$epoch$	bs	M
UIFW	3	75	22 (29)	10^{-4}	50	32	300
CLD	3	75	120 (157)	10^{-4}	50	32	200
HOP	8	75	180 (87)	10^{-4}	20	32	300

Additionally, the number of Monte Carlo samples (M) used to estimate predictive uncertainty was iterated from 50 to 1000. Because larger M incurs greater computational overhead, we maximize M for each dataset until the proposed fusion method outperforms simple averaging. It is worth noting that the optimal value of M can be checked based on the convergence of Eq. (2) for the validation data. Specifically, we applied the d'Alembert ratio test, which evaluates the limit of the absolute value of the ratio between successive uncertainty estimates: $\sigma = \lim_{i\to\infty} \left| \frac{\mathbb{UM}^{(i)}}{\mathbb{UM}^{(i-1)}} \right|$, where $\mathbb{UM}^{(i)}$ represents the predictive uncertainty after the i^{th} sampling iteration. This test criterion approaches unity as the uncertainty estimates stabilize. By exploring the behavior of σ in Fig. 7, we could visually assess that for larger M, the uncertainty $\mathbb{UM}$ converged. In addition, the model was trained using the Nadam algorithm and the categorical cross-entropy loss function. All modeling work was implemented using Python alongside the TensorFlow and scikit-learn library and are publicly available on GitHub.

d) Training Procedure: In the methodology section, it was mentioned that the prediction errors from the inner classifier and the fused outputs do not have a direct relationship. Therefore, during the training of the inner classifier, the fusion error of validation data based on Eq. (3) is the evaluation metric for monitoring. To examine this claim, Fig. 8 shows the MS_{PBS} curves for three datasets, comparing the inner classifier predictions (blue line) to the fused results (red line). Firstly, the figure illustrates that the minimum errors before and after fusion do not coincide. Moreover, the gray vertical lines in the figure indicate points where the pre-fusion error increases while the post-fusion error decreases, and vice versa. Together, these insights validate our methodology of monitoring the fusion error rather than inner classifier errors alone during training, as it leads to the most relevant optimization of the overall probabilistic model.

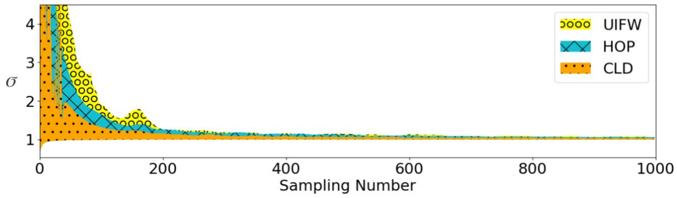

Fig. 7. Convergence of predictive uncertainty $\mathbb{UM}$ on validation data as the number of Monte Carlo samples (M) increases.

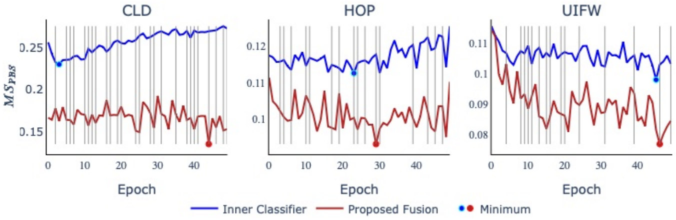

Fig. 8. Analysis of the MS_{PBS} before (blue) and after (red) fusion.

B. Performance of Uncertainty Quantification

a) Baseline Fusion Methods: In this step, the effectiveness of the proposed fusion model, Eq. (10), is compared with two baseline fusion models, averaging and majority voting. Table III reports the results of this comparison using two key metrics: F1-Score and MSE. As shown, the proposed fusion approach outperforms the baseline methods.

TABLE III
FUSION METHOD ANALYSIS

Dataset	**Fusion Method**	**MSE**	**F1**
UIFW	Majority Voting	0.0149 (± 0.008)	85.57 (± 0.08)
	Averaging	0.0147 (± 0.008)	85.74 (± 0.09)
	Our ($\mathcal{WA}$)	**0.0141 (± 0.008)**	**88.10 (± 0.09)**
CLD	Majority Voting	0.0314 (± 0.015)	95.30 (± 0.05)
	Averaging	0.0329 (± 0.016)	95.49 (± 0.05)
	Our ($\mathcal{WA}$)	**0.0250 (± 0.012)**	**98.67 (± 0.03)**
HOP	Majority Voting	0.0388 (± 0.007)	62.39 (± 0.08)
	Averaging	0.0385 (± 0.005)	64.63 (± 0.08)
	Our ($\mathcal{WA}$)	**0.0363 (± 0.006)**	**65.11 (± 0.08)**

b) Baseline Classification Models: One of the main competing approaches to fusion models for user identification is deep recurrent neural networks. Recurrent models aim to learn temporal dependencies between subsequences or activities over time. However, we believe that for this task, the subsequences may not always be strongly dependent on each other and could often be considered independently. Therefore, fusion models that treat subsequences independently, such as by averaging, may be better suited. To validate this, we compared our probabilistic fusion approach against recurrent state-of-the-art approaches including GRU, LSTM, BiLSTM, and ConvLSTM [3], [15], [28]. Table IV reports the results of this comparison using two important metrics: F1-Score and MSE. As shown, the proposed model outperformed each of the individual baseline models according to both evaluation metrics. Furthermore, our fusion technique led to improved results over using the inner classifier models directly without fusion, as discussed in Section V-A0a.

c) Benchmark: An empirical comparison to related work [15], [3] was conducted using the DB2 and HAR datasets, which were also evaluated in those studies. To ensure a fair evaluation, we followed the same dataset configurations as in the prior work: both datasets had been pre-segmented into standardized train/test splits. Furthermore, 20% of the training data was held out for validation. A grid search was conducted to determine the best hyperparameters for the fusion model. The optimal values found were: $w = 32s$, $r = 24s$, $s = 128s$, $lr = 10^{-4}$, $batch\ size = 32$, $epochs = 50$, $M = 200$. Table V reports results versus [15], [3] on common classification metrics: MSE, F1-Score, Recall, Precision, and Accuracy - some of which were not reported previously. On the DB2 and HAR datasets, our model outperformed the individual baselines [15] and [3] on all metrics evaluated in prior works. Additionally, our model exceeds the baselines on F1-Score, Recall, and Accuracy, reported here for the first time to enable more comprehensive benchmarking. This controlled experimental evaluation demonstrates the proposed fusion method achieves state-of-the-art user identification accuracy.

TABLE IV
COMPARING WITH RECURRENT MODELS

Dataset	Model	MSE	F1
UIFW	GRU	0.0490 (± 0.009)	54.56 (± 0.07)
	LSTM	0.0493 (± 0.008)	56.10 (± 0.07)
	BiLSTM	0.0437 (± 0.011)	66.56 (± 0.06)
	ConvLSTM	0.0574 (± 0.014)	48.18 (± 0.08)
	Our	**0.0147 (± 0.008)**	**85.74 (± 0.09)**
CLD	GRU	0.1576 (± 0.015)	39.65 (± 0.06)
	LSTM	0.1823 (± 0.011)	37.25 (± 0.04)
	BiLSTM	0.1408 (± 0.029)	52.22 (± 0.12)
	ConvLSTM	0.1831 (± 0.013)	36.33 (± 0.03)
	Our	**0.0329 (± 0.016)**	**95.49 (± 0.05)**
HOP	GRU	0.0687 (± 0.008)	36.84 (± 0.07)
	LSTM	0.0699 (± 0.007)	36.45 (± 0.06)
	BiLSTM	0.0779 (± 0.008)	36.91 (± 0.06)
	ConvLSTM	0.0633 (± 0.006)	32.06 (± 0.06)
	Our	**0.0385 (± 0.005)**	**64.63 (± 0.08)**

TABLE V
BENCHMARK

Data	Ref.	MSE	F1	Recall	Precision	Accuracy
DB2	[15]	-	-	-	-	97.33
	Our	0.0012	97.23	96.98	98.89	**98.62**
HAR	[3]	-	91.18	91.27	-	91.31
	Our	0.0078	**99.26**	**99.27**	99.31	**99.29**

VI. CONCLUSION

This research introduced a probabilistic fusion approach for identifying users using smartphones. The study showed that distinguishing individuals is accomplished through subtle variations in details of activity patterns, though some activities remain challenging to differentiate. To address this, uncertainty quantification by Monte Carlo Dropout was applied to predictions from a CNN classifier regarding divided subsequences, where higher uncertainty implied lower separability. Predictions were then combined using weighted averaging, with weights inversely corresponding to uncertainty levels. Experiments using five datasets indicated the proposed fusion model performed superiorly compared to baseline and recurrent models. By evaluating two datasets from previous state-of-the-art methods, our approach surpassed prior best accuracy levels by over 1.29% and 7.98% respectively. However, this research faced certain limitations. First, fixed-length sliding windows were employed for segmentation whereas dynamically adjusting window sizes based on activity duration could enhance performance. Additionally, the work focused on identification rather than authentication. Authentication presents a binary problem of authenticating one's identity. For future work, alternative uncertainty methods like Markov Chain Monte Carlo (MCMC) and variational inference could be explored. Furthermore, noisy label techniques may aid the analysis of difficult-to-separate subsequences (see, e.g., [36]).

REFERENCES

[1] 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.

[2] Z. Guo, J. Cao, X. Bai, A. Li, B. Niu, and H. Li, "Shake, shake, i know who you are: Authentication through smart wearable devices," *IEEE Sensors Journal*, 2023.

[3] F. Luo, S. Khan, Y. Huang, and K. Wu, "Activity-based person identification using multimodal wearable sensor data," *IEEE Internet of Things Journal*, vol. 10, no. 2, pp. 1711–1723, 2022.

[4] M. Abuhamad, A. Abusnaina, D. Nyang, and D. Mohaisen, "Sensor-based continuous authentication of smartphones' users using behavioral biometrics: A contemporary survey," *IEEE Internet of Things Journal*, vol. 8, no. 1, pp. 65–84, 2020.

[5] R. Ahmadian, M. Ghatee, and J. Wahlström, "Discrete wavelet transform for generative adversarial network to identify drivers using gyroscope and accelerometer sensors," *IEEE Sensors Journal*, vol. 22, no. 7, pp. 6879–6886, 2022.

[6] S. R. V. Sudhakar, N. Kayastha, and K. Sha, "Actid: An efficient framework for activity sensor based user identification," *Computers & Security*, vol. 108, p. 102319, 2021.

[7] A. Oğuz and Ö. F. Ertuğrul, "Human identification based on accelerometer sensors obtained by mobile phone data," *Biomedical signal processing and control*, vol. 77, p. 103847, 2022.

[8] F. Sun, W. Zang, R. Gravina, G. Fortino, and Y. Li, "Gait-based identification for elderly users in wearable healthcare systems," *Information fusion*, vol. 53, pp. 134–144, 2020.

[9] R. Ahmadian, M. Ghatee, and J. Wahlström, "Driver identification by an ensemble of cnns obtained from majority-voting model selection," in *International Conference on Artificial Intelligence and Smart Vehicles*. Springer, 2023, pp. 120–136.

[10] R. Ahmadian, M. Ghatee, and J. Wahlstrom, "Training of neural networks to classify spatiotemporal data by probabilistic fusion on hopping windows: Theory and experiments," *Available at SSRN 4616995*.

[11] W. Xu, Y. Shen, C. Luo, J. Li, W. Li, and A. Y. Zomaya, "Gait-watch: A gait-based context-aware authentication system for smart watch via sparse coding," *Ad Hoc Networks*, vol. 107, p. 102218, 2020.

[12] P. Casale, O. Pujol, and P. Radeva, "Personalization and user verification in wearable systems using biometric walking patterns," *Personal and Ubiquitous Computing*, vol. 16, pp. 563–580, 2012.

[13] B. Kaluža, V. Mirchevska, E. Dovgan, M. Luštrek, and M. Gams, "An agent-based approach to care in independent living," in *International joint conference on ambient intelligence*. Springer, 2010, pp. 177–186.

[14] R. L. S. Torres, D. C. Ranasinghe, Q. Shi, and A. P. Sample, "Sensor enabled wearable rfid technology for mitigating the risk of falls near beds," in *2013 IEEE international conference on RFID (RFID)*. IEEE, 2013, pp. 191–198.

[15] Q. Zou, Y. Wang, Q. Wang, Y. Zhao, and Q. Li, "Deep learning-based gait recognition using smartphones in the wild," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3197–3212, 2020.

[16] Y. Sun, F. P.-W. Lo, and B. Lo, "Eeg-based user identification system using 1d-convolutional long short-term memory neural networks," *Expert Systems with Applications*, vol. 125, pp. 259–267, 2019.

[17] C. Shen, Z. Wang, C. Si, Y. Chen, and X. Su, "Waving gesture analysis for user authentication in the mobile environment," *IEEE Network*, vol. 34, no. 2, pp. 57–63, 2020.

[18] M. Abuhamad, T. Abuhmed, D. Mohaisen, and D. Nyang, "Autosen: Deep-learning-based implicit continuous authentication using smartphone sensors," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5008–5020, 2020.

[19] X. Wang, F. Li, Y. Xie, S. Yang, and Y. Wang, "Gait and respiration-based user identification using wi-fi signal," *IEEE Internet of Things Journal*, vol. 9, no. 5, pp. 3509–3521, 2021.

[20] T. Zhao, Y. Wang, J. Liu, J. Cheng, Y. Chen, and J. Yu, "Robust continuous authentication using cardiac biometrics from wrist-worn wearables," *IEEE Internet of Things Journal*, vol. 9, no. 12, pp. 9542–9556, 2021.

[21] S. Vhaduri and C. Poellabauer, "Multi-modal biometric-based implicit authentication of wearable device users," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 12, pp. 3116–3125, 2019.

[22] A. I. Middya, S. Roy, and S. Mandal, "User recognition in participatory sensing systems using deep learning based on spectro-temporal representation of accelerometer signals," *Knowledge-Based Systems*, vol. 258, p. 110046, 2022.

[23] H. Alobaidi, N. Clarke, F. Li, and A. Alruban, "Real-world smartphone-based gait recognition," *Computers & Security*, vol. 113, p. 102557, 2022.

[24] G. M. Weiss, "Wisdm smartphone and smartwatch activity and biometrics dataset," *UCI Machine Learning Repository: WISDM Smartphone and Smartwatch Activity and Biometrics Dataset Data Set*, vol. 7, pp. 133 190–133 202, 2019.

[25] S. Bianco, P. Napoletano, A. Raimondi, and M. Rima, "U-wear: User recognition on wearable devices through arm gesture," *IEEE transactions on human-machine systems*, vol. 52, no. 4, pp. 713–724, 2022.

[26] L. Kuang, F. Zeng, D. Liu, H. Cao, H. Jiang, and J. Liu, "Lipauth: Securing smartphone user authentication with lip motion patterns," *IEEE Internet of Things Journal*, 2023.

[27] H. Cao, H. Jiang, K. Yang, S. Chen, W. Wu, J. Liu, and S. Dustdar, "Data augmentation-enabled continuous user authentication via passive vibration response," *IEEE Internet of Things Journal*, 2023.

[28] G. Batchuluun, R. A. Naqvi, W. Kim, and K. R. Park, "Body-movement-based human identification using convolutional neural network," *Expert Systems with Applications*, vol. 101, pp. 56–77, 2018.

[29] M. Gadaleta and M. Rossi, "Idnet: Smartphone-based gait recognition with convolutional neural networks," *Pattern Recognition*, vol. 74, pp. 25–37, 2018.

[30] S. Sprager and M. B. Juric, "An efficient hos-based gait authentication of accelerometer data," *IEEE transactions on information forensics and security*, vol. 10, no. 7, pp. 1486–1498, 2015.

[31] A. Kececi, A. Yildirak, K. Ozyazici, G. Ayluctarhan, O. Agbulut, and I. Zincir, "Implementation of machine learning algorithms for gait recognition," *Engineering Science and Technology, an International Journal*, vol. 23, no. 4, pp. 931–937, 2020.

[32] Y. Li, L. Liu, H. Qin, S. Deng, M. A. El-Yacoubi, and G. Zhou, "Adaptive deep feature fusion for continuous authentication with data augmentation," *IEEE Transactions on Mobile Computing*, 2022.

[33] F. Sun, C. Mao, X. Fan, and Y. Li, "Accelerometer-based speed-adaptive gait authentication method for wearable iot devices," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 820–830, 2018.

[34] J. Moon, J. Jung, E. Kang, and S.-I. Choi, "Open set user identification using gait pattern analysis based on ensemble deep neural network," *IEEE Sensors Journal*, vol. 22, no. 17, pp. 16 975–16 984, 2022.

[35] J. Gawlikowski, C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. Kruspe, R. Triebel, P. Jung, R. Roscher *et al.*, "A survey of uncertainty in deep neural networks," *Artificial Intelligence Review*, vol. 56, no. Suppl 1, pp. 1513–1589, 2023.

[36] E. Hüllermeier and W. Waegeman, "Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods," *Machine Learning*, vol. 110, pp. 457–506, 2021.

[37] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*. PMLR, 2016, pp. 1050–1059.

[38] R. Ahmadian, M. Ghatee, and J. Wahlstrom, "Superior scoring rules for probabilistic evaluation of single-label multi-class classification tasks," *Under Review*.

[39] B. Charpentier, D. Zügner, and S. Günnemann, "Posterior network: Uncertainty estimation without ood samples via density-based pseudo-counts," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1356–1367, 2020.