IEEE Access
The journal for rapid open access publishing

Received March 23, 2016, accepted April 5, 2016, date of publication April 22, 2016, date of current version May 9, 2016.

Digital Object Identifier 10.1109/ACCESS.2016.2557846

Learning Human Identity From Motion Patterns

NATALIA NEVEROVA[1], CHRISTIAN WOLF[1], GRIFFIN LACEY[2], LEX FRIDMAN[3], DEEPAK CHANDRA[4], BRANDON BARBELLO[4], AND GRAHAM TAYLOR[2]

[1]Laboratoire d'InfoRmatique en Image et Systèmes, Centre National de la Recherche Scientifique, Institut National des Sciences Appliquées de Lyon, Université de Lyon, Lyon 69621, France
[2]School of Engineering, University of Guelph, Guelph, ON N1G 2W1, Canada
[3]Massachusetts Institute of Technology, Cambridge, MA 02139, USA
[4]Google, Mountain View, CA 94043, USA

Corresponding author: N. Neverova (natalia.neverova@liris.cnrs.fr)

ABSTRACT We present a large-scale study exploring the capability of temporal deep neural networks to interpret natural human kinematics and introduce the first method for active biometric authentication with mobile inertial sensors. At Google, we have created a first-of-its-kind data set of human movements, passively collected by 1500 volunteers using their smartphones daily over several months. We compare several neural architectures for efficient learning of temporal multi-modal data representations, propose an optimized shift-invariant dense convolutional mechanism, and incorporate the discriminatively trained dynamic features in a probabilistic generative framework taking into account temporal characteristics. Our results demonstrate that human kinematics convey important information about user identity and can serve as a valuable component of multi-modal authentication systems. Finally, we demonstrate that the proposed model can also be successfully applied in a visual context.

INDEX TERMS Authentication, biometrics (access control), learning, mobile computing, recurrent neural networks.

I. INTRODUCTION

For the billions of smartphone users worldwide, remembering dozens of passwords for all services we need to use and spending precious seconds on entering pins or drawing sophisticated swipe patterns on touchscreens becomes a source of frustration. In recent years, researchers in different fields have been working on creating fast and secure authentication alternatives that would make it possible to remove this burden from the user [6], [29].

Historically, biometrics research has been hindered by the difficulty of collecting data, both from a practical and legal perspective. Previous studies have been limited to tightly constrained lab-scale data collection, poorly representing real world scenarios: not only due to the limited amount and variety of data, but also due to essential *self consciousness* of participants performing the tasks. In response, we created an unprecedented dataset of *natural* prehensile movements (i.e. those in which an object is seized and held, partly or wholly, by the hand [20]) collected by 1,500 volunteers over several months of daily use (Fig. 1).

Apart from data collection, the main challenges in developing a *continuous* authentication system for smartphones are (1) efficiently learning task-relevant representations of noisy inertial data, and (2) incorporating them into a biometrics setting, characterized by limited resources. Limitations include low computational power for model adaptation to a new user and for real-time inference, as well as the absence (or very limited amount) of "negative" samples.

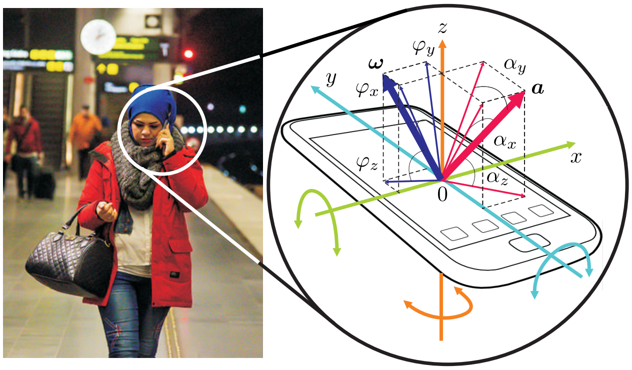

FIGURE 1. The accelerometer captures linear acceleration, the gyroscope provides angular velocity (photo taken from [24]).

In response to the above challenges, we propose a non-cooperative and non-intrusive method for on-device authentication based on two key components: temporal feature extraction by deep neural networks, and classification via a probabilistic generative model. We assess several popular deep architectures including one-dimensional convolutional nets and recurrent neural networks for feature extraction. However, apart from the application itself, the main contribution of this work is in developing a new shift-invariant temporal model which fixes a deficiency of the recently proposed Clockwork recurrent neural networks [18] yet retains their ability to explicitly model multiple temporal scales.

II. RELATED WORK

Exploiting wearable or mobile inertial sensors for authentication, action recognition or estimating parameters of a particular activity has been explored in different contexts. Gait analysis has attracted significant attention from the biometrics community as a non-contact, non-obtrusive authentication method resistant to spoofing attacks. A detailed overview and benchmarking of existing state-of-the-art is provided in [23]. Derawi et al. [9], for example, used a smartphone attached to the human body to extract information about walking cycles, achieving 20.1% equal error rate.

There exist a number of works which explore the problem of activity and gesture recognition with motion sensors, including methods based on deep learning. In [7] and [12], exhaustive overviews of preprocessing techniques and manual feature extraction from accelerometer data for activity recognition are given. Perhaps most relevant to this study is [25], the first to report the effectiveness of RBM-based feature learning from accelerometer data, and [5], which proposed a data-adaptive sparse coding framework. Convolutional networks have been explored in the context of gesture and activity recognition [11], [34]. Lefebvre et al. [19] applied a bidirectional LSTM network to a problem of 14-class gesture classification, while Berlemont et al. [4] proposed a fully-connected Siamese network for the same task.

We believe that multi-modal frameworks are more likely to provide meaningful security guarantees. A combination of face recognition and speech [17], and of gait and voice [32] have been proposed in this context. Deep learning techniques, which achieved early success modeling sequential data such as motion capture [31] and video [16] have shown promise in multi-modal feature learning [15], [21], [22], [30].

III. A GENERATIVE BIOMETRIC FRAMEWORK

Our goal is to separate a user from an impostor based on a time series of inertial measurements (Fig. 1). Our method is based on two components: a feature extraction pipeline which associates each user's motion sequence with a collection of discriminative features, and a biometric model, which accepts those features as inputs and performs verification. While the feature extraction component is the most interesting and novel aspect of our technique, we delay its discussion to Section IV. We begin by discussing the data format and the biometric model.

A. MOVEMENT DATA

Each reading (frame) in a synchronized raw input stream of accelerometer and gyroscope data has the form $\{a_x, a_y, a_z, \omega_x, \omega_y, \omega_z\} \in \mathbb{R}^6$, where a represents linear acceleration, ω angular velocity and x, y, z denote projections on corresponding axes, aligned with the phone. There are two important steps we take prior to feature extraction.

1) OBFUSCATION-BASED REGULARIZATION

It is important to differentiate between the notion of "device" and "user". In the dataset we collected (Section VI), each device is assigned to a single user, thus all data is considered to be authentic. However, in real-world scenarios such as theft, authentic and imposter data may originate from the same device.

In a recent study [8], it was shown that under lab conditions a particular *device* could be identified by a response of its motion sensors to a given signal. This happens due to imperfection in calibration of a sensor resulting in constant offsets and scaling coefficients (gains) of the output, that can be estimated by calculating integral statistics from the data. Formally, the measured output of both the accelerometer and gyroscope can be expressed as follows [8]:

$$\mathbf{a} = \mathbf{b}_a + \mathrm{diag}(\boldsymbol{\gamma}_a)\tilde{\mathbf{a}}, \quad \boldsymbol{\omega} = \mathbf{b}_\omega + \mathrm{diag}(\boldsymbol{\gamma}_\omega)\tilde{\boldsymbol{\omega}}, \tag{1}$$

where $\tilde{\mathbf{a}}$ and $\tilde{\boldsymbol{\omega}}$ are real acceleration and angular velocity vectors, $\mathbf{b}_a$ and $\mathbf{b}_\omega$ are offset vectors and $\boldsymbol{\gamma}_a$ and $\boldsymbol{\gamma}_\omega$ represent gain errors along each coordinate axes.

To partially obfuscate the inter-device variations and ensure decorrelation of user identity from device signature in the learned data representation, we introduce low-level additive (offset) and multiplicative (gain) noise per training example. Following [8], the noise vector is obtained by drawing a 12-dimensional (3 offset and 3 gain coefficients per sensor) obfuscation vector from a uniform distribution $\boldsymbol{\mu} \sim \mathcal{U}_{12}[0.98, 1.02]$.

2) DATA PREPROCESSING

In addition, we extract a set of angles $\alpha_{\{x,y,z\}}$ and $\varphi_{\{x,y,z\}}$ describing the orientation of vectors $\mathbf{a}$ and $\boldsymbol{\omega}$ in the phone's coordinate system (see Fig. 1), compute their magnitudes $|\mathbf{a}|$ and $|\boldsymbol{\omega}|$ and normalize each of the x, y, z components. Finally, the normalized coordinates, angles and magnitudes are combined in a 14-dimensional vector $\mathbf{x}^{(t)}$ with t indexing the frames.

B. BIOMETRIC MODEL

Relying on cloud computing to authenticate a mobile user is unfeasible due to privacy and latency. Although this technology is well established for many mobile services, our application is essentially different from others such as voice search, as it involves constant background collection of particularly sensitive user data. Streaming this information to the cloud would create an impermissible threat from a privacy perspective for users and from a legal perspective for service providers. Therefore, authentication must be performed on the device and is constrained by available storage, memory and processing power. Furthermore, adapting to a new user should be quick, resulting in a limited amount of training data for the "positive" class. This data may not be completely representative of typical usage. For these reasons, a purely discriminative setting involving learning a separate model per user, or even fine-tuning a model for each new user would hardly be feasible.

Therefore, we adapt a generative model, namely a Gaussian Mixture Model (GMM), to estimate a general data distribution in the dynamic motion feature space and create a *universal background model* (UBM). The UBM is learned offline, i.e. prior to deployment on the phones, using a large amount of pre-collected training data. For each new user we use a very small amount of enrollment samples to perform online (i.e. on-device) adaptation of the UBM to create a *client model*. The two models are then used for real time inference of trust scores allowing continuous authentication.

1) UNIVERSAL BACKGROUND MODEL

Let $\mathbf{y} = f(\{\mathbf{x}^{(t)}\}) \in \mathbb{R}^N$ be a vector of features extracted from a raw sequence of prehensile movements by one of the deep neural networks described in Section IV. Probability densities are defined over these feature vectors as a weighted sum of M multi-dimensional Gaussian distributions parameterized by $\Theta = \{\mu_i, \Sigma_i, \pi_i\}$, where μ_i is a mean vector, Σ_i a covariance matrix and π_i a mixture coefficient:

$$p(\mathbf{y}|\Theta) = \sum_{i=1}^{M} \pi_i \mathcal{N}(\mathbf{y}; \mu_i, \Sigma_i), \tag{2}$$

$$\mathcal{N}_i(\mathbf{y}) = \frac{1}{\sqrt{(2\pi)^N |\Sigma_i|}} e^{-\frac{(\mathbf{y}-\mu_i)'\Sigma_i^{-1}(\mathbf{y}-\mu_i)}{2}}. \tag{3}$$

The UBM $p(\mathbf{y}|\Theta_{\text{UBM}})$ is learned by maximising the likelihood of feature vectors extracted from the large training set using the expectation-maximisation (EM) algorithm.

The client model $p(\mathbf{y}|\Theta_{\text{client}})$ is adapted from the UBM. Both models share the same weights and covariance matrices to avoid overfitting from a limited amount of enrollment data. Along the lines of [26], maximum a posteriori (MAP) adaptation of mean vectors for a given user is performed. This has an immediate advantage over creating an independent GMM for each user, ensuring proper alignment between the well-trained background model and the client model by updating only a subset of parameters that are specific to the given user. In particular, given a set of Q enrollment samples $\{\mathbf{y}_q\}$ from the new device, we create a client-specific update to the mean of each mixture component i as follows:

$$E_i(\{\mathbf{y}_q\}) = \frac{1}{n_i} \sum_{q=1}^{Q} Pr(i|\mathbf{y}_q)\mathbf{y}_q, \tag{4}$$

where

$$n_i = \sum_{q=1}^{Q} Pr(i|\mathbf{y}_q), \quad Pr(i|\mathbf{y}_q) = \frac{\pi_i p_i(\mathbf{y}_q)}{\sum_{j=1}^{M} \pi_j p_j(\mathbf{y}_q)}. \tag{5}$$

Finally, the means of all Gaussian components are updated according to the following rule:

$$\hat{\mu}_i = \alpha_i E_i(\{\mathbf{y}_q\}) + (1 - \alpha_i)\mu_i,$$

where

$$\alpha_i = \frac{n_i}{n_i + r}, \tag{6}$$

where r is a relevance factor balancing the background and client models. In our experiments, r is held fixed.

2) SCORING

Given a set of samples $Y = \{\mathbf{y}_s\}$ from a given device, authenticity is estimated by scoring the feature vectors against the UBM and the client model, thresholding the log-likelihood ratio:

$$\Lambda(Y) = \log p(Y|\Theta_{\text{client}}) - \log p(Y|\Theta_{\text{UBM}}). \tag{7}$$

As a final step, zt-score normalization [2] is performed to compensate for inter-session and inter-person variations and reduce the overlap between the distribution of scores from authentic users and impostors.

IV. LEARNING EFFECTIVE AND EFFICIENT REPRESENTATIONS

Learning effective and efficient data representations is key to our entire framework since its ability to perform in the real-world is defined by such criteria as latency, representational power of extracted features and inference speed of the feature extractor. The first two conditions are known to contradict each other as performance of a standalone feature typically grows with integration time [21].

Two paradigms which strike a balance between representational power and speed have dominated the feature learning landscape in recent years. These are multi-scale temporal aggregation via 1-dimensional convolutional networks Fig. 2a, and explicit modeling of temporal dependencies via recurrent neural networks Fig. 2b.

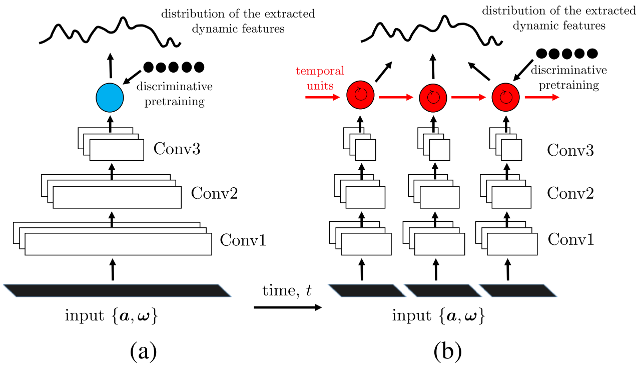
distribution of the extracted dynamic features
discriminative pretraining
Conv3
Conv2
Conv1
input $\{a, \omega\}$
(a)
time, t
distribution of the extracted dynamic features
temporal units
discriminative pretraining
Conv3
Conv2
Conv1
input $\{a, \omega\}$
(b)

FIGURE 2. Learning data representations: (a) static convnet directly operating on sequences, aggregating temporal statistics by temporal pooling; (b) explicitly modeling temporal transitions with recurrent connections.

The former model, popular in speech recognition [13], involves convolutional learning of integrated temporal statistics from short and long sequences of data (referred to as "short-term" and "long-term" convnets). Short-term architectures produce outputs at relatively high rate (1 Hz in our implementation) but fail to model context. Long-term networks can learn meaningful representations at different scales, but suffer from a high degree of temporal inertia and do not generalize to sequences of arbitrary length.

Recurrent models which explicitly model temporal evolutions can generate low-latency feature vectors built in the context of previously observed user behavior. The dynamic nature of their representations allow for modeling richer temporal structure and better discrimination among users acting under different conditions. There have been a sufficiently large number of neural architectures proposed for modeling temporal dependencies in different contexts: the baseline methods compared in this work are summarized in Fig. 3c. The rest of this section provides a brief description of these models. Then, Section V introduces a new shift-invariant model based on modified Clockwork RNNs [18].

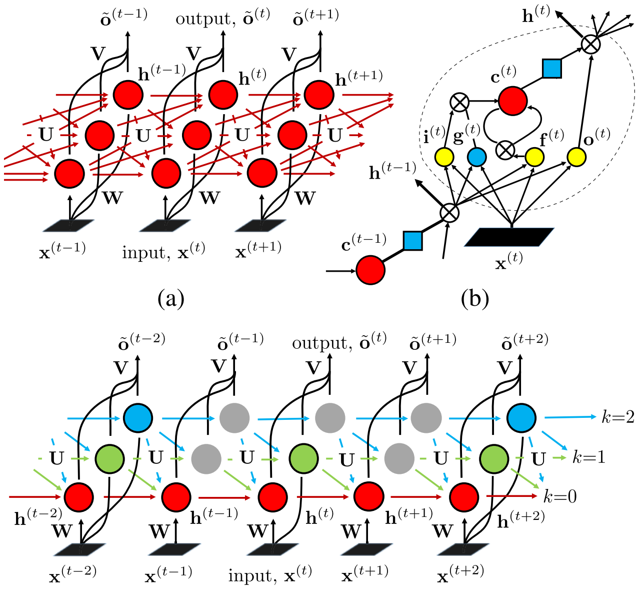

FIGURE 3. Temporal models: (a) a basic recurrent unit; (b) an LSTM unit [14]; (c) Clockwork RNN [18] with 3 bands and a base of 2; Increasing *k* indicates lower operating frequency. Grey color indicates inactivity of a unit..

All feature extractors are first pretrained discriminatively for a multi-device classification task, then, following removal of the output layer the activations of the penultimate hidden layer are provided as input (which we denote, for conciseness,[1] by $\mathbf{y}$) to the generative model described in Section III. The final outputs of the background and client models are integrated over a 30 sec window. Accordingly, after 30 sec the user is either authenticated or rejected.

A. CLASSICAL RNN AND CLOCKWORK RNN

The vanilla recurrent neural network (RNN) is governed by the update equation

$$\mathbf{h}^{(t)} = \psi(\mathbf{U}\mathbf{h}^{(t-1)} + \mathbf{W}\mathbf{x}^{(t)}), \tag{8}$$

where $\mathbf{x}$ is the input, $\mathbf{h}^{(t)}$ denotes the network's hidden state at time t, $\mathbf{W}$ and $\mathbf{U}$ are feed-forward and recurrent weight matrices, respectively, and ψ is a nonlinear activation function, typically tanh. The output is produced combining the hidden state in a similar way, $\mathbf{o}^{(t)} = \phi(\mathbf{V}\mathbf{h}^{(t)})$, where $\mathbf{V}$ is a weight matrix.

One of the main drawbacks of this model is that it operates at a predefined temporal scale. In the context of free motion which involves large variability in speed and changing intervals between typical gestures, this may be a serious limitation. The recently proposed Clockwork RNN (CWRNN) [18] operates at several temporal scales which are incorporated in a single network and trained jointly. It decomposes a recurrent layer into several bands of high frequency ("fast") and low frequency ("slow") units (see Fig. 3c). Each band is updated at its own pace. The size of the step from band to band typically increases exponentially (which we call *exponential update rule*) and is defined as n^k, where n is a base and k is the number of the band.

In the CWRNN, fast units (shown in red) are connected to all bands, benefitting from the context provided by the slow bands, while the low frequency units ignore noisy high frequency oscillations. Equation (8) from classical RNNs is modified, leading to a new update rule for the k-th band of output $\mathbf{h}$ at iteration t as follows:

$$\mathbf{h}_k^{(t)} = \begin{cases} \psi\left(\mathbf{U}(k)\mathbf{h}_k^{(t-1)} + \mathbf{W}(k)\mathbf{x}^{(t)}\right) & \text{if } (t \bmod n^k) = 0, \\ \mathbf{h}_k^{(t-1)} & \text{otherwise} \end{cases}$$

where $\mathbf{U}(k)$ and $\mathbf{W}(k)$ denote rows k from matrices $\mathbf{U}$ and $\mathbf{W}$. Matrix $\mathbf{U}$ has an upper triangular structure, which corresponds to the connectivity between frequency bands. This equation is intuitively explained in the top part of Fig. 4, inspired from [18]. Each line corresponds to a band. At time step $t = 6$ for instance, the first two bands $k = 0$ and $k = 1$ get updated. The triangular structure of the matrix results in each band getting updated from bands of lower (or equal) frequency only. In Fig. 4, not active rows are also shown as zero (black) in $\mathbf{U}$ and $\mathbf{W}$. In addition to multi-scale dynamics, creating sparse connections (high-to-low frequency connections are missing) reduces the number of free parameters and inference complexity.

B. LONG SHORT-TERM MEMORY

Long Short-Term Memory (LSTM) networks [14], another variant of RNNs, and their recent convolutional

[1] This decision was made to avoid introducing notation to index hidden layers, as well as simplify and generalize the presentation in the previous section, where the $\mathbf{y}$ are taken as generic temporal features.

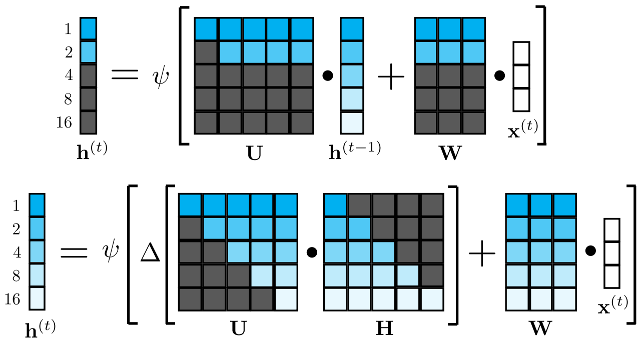

FIGURE 4. Updates made by the Clockwork RNN [18] (top) and our proposed Dense CWRNN (bottom). Units and weights colored in blue are the ones updated or read at the example time step $t = 6$.

extensions [10], [27] have proven to be, so far, the best performing models for learning long-term temporal dependencies. They handle information from the past through additional gates, which regulate how a memory cell is affected by the input signal. In particular, an input gate allows to add new memory to the cell's state, a forget gate resets the memory and an output gate regulates how gates at the next step will be affected by the current cell's state.

The basic unit is composed of input i, output o, forget f, and input modulation g gates, and a memory cell c (see Fig. 3b). Each element is parameterized by corresponding feed-forward ($\mathbf{W}$) and recurrent ($\mathbf{U}$) weights and bias vectors.

Despite its effectiveness, the high complexity of this architecture may appear computationally wasteful in the mobile setting. Furthermore, the significance of learning long-term dependencies in the context of continuous mobile authentication is compromised by the necessity of early detection of switching between users. Due to absence of annotated ground truth data for these events, efficient training of forgetting mechanisms would be problematic.

C. CONVOLUTIONAL LEARNING OF RNNs

Given the low correlation of individual frames with user identity, we found it strongly beneficial to make the input layer convolutional regardless of model type, thereby forcing earlier fusion of temporal information. To simplify the presentation, we have not made convolution explicit in the description of the methods above, however, it can be absorbed into the input-to-hidden matrix $\mathbf{W}$.

V. DENSE CONVOLUTIONAL CLOCKWORK RNNs

Among the existing temporal models we considered, the clockwork mechanisms appear to be the most attractive due to low computational burden associated with them in combination with their high modeling capacity. However, in practice, due to inactivity of "slow" units for long periods of time, they cannot respond to high frequency changes in the input and produce outputs which, in a sense, are stale. Additionally, in our setting, where the goal is to learn dynamic data representations serving as an input to a probabilistic framework, this architecture has one more weakness which stems from the fact that different bands are active at any given time step. The network will respond differently to the same input stimuli applied at different moments in time. This "shift-variance" convolutes the feature space by introducing a shift-associated dimension.

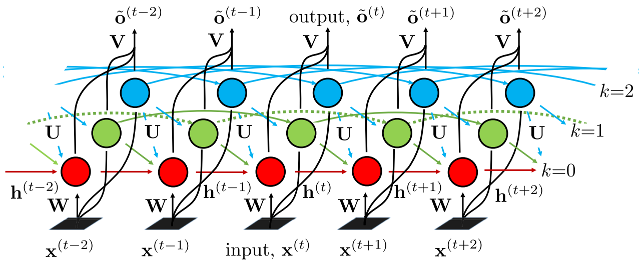

FIGURE 5. Proposed dense clockwork RNN with the same parameters as the original clockwork RNN shown in Fig. 3a.

In this work, we propose a solution to both issues, namely "twined" or "dense" clockwork mechanisms (DCWRNN, see Fig. 5), where during *inference* at each scale k there exist n^k parallel threads shifted with respect to each other, such that at each time a unit belonging to one of the threads fires, updating its own state and providing input to the higher frequency units. All weights between the threads belonging to the same band are shared, keeping the overall number of parameters in the network the same as in the original clockwork architecture. Without loss of generality, and to keep the notation uncluttered of unnecessary indices, in the following we will describe a network with a single hidden unit h_k per band k. The generalization to multiple units per band is straightforward, and the experiments were of course performed with the more general case.

The feedforward pass for the whole dense clockwork layer (i.e. all bands) can be given as follows:

$$\mathbf{h}^{(t)} = \psi\left(\mathbf{W}\mathbf{x}^{(t)} + \Delta(\mathbf{U}\mathbf{H})\right) \qquad (9)$$

where $H = [\mathbf{h}^{(t-1)} \ldots \mathbf{h}^{(t-n^k)} \ldots \mathbf{h}^{(t-n^K)}]$ is a matrix concatenating the history of hidden units and we define $\Delta(\cdot)$ as an operator on matrices returning its diagonal elements in a column vector. The intuition for this equation is given in Fig. 4, where we compare the update rules of the original CWRNN and the proposed DCWRNN using an example of a network with 5 hidden units each associated with one of $K = 5$ base $n = 2$ bands. To be consistent, we employ the same matrix form as in the original CWRNN paper [18]) and show components, which are inactive at time t, in dark gray. As mentioned in Section IV-A, in the original CWRNN, at time instant $t = 6$, for instance, only unit h_1 and h_2 are updated, i.e. the first two lines in Fig. 4. In the dense network, all hidden units h_k are updated at each moment in time.

In addition, what was vector of previous hidden states $\mathbf{h}^{(t-1)}$ is replaced with a lower triangular "history" matrix $\mathbf{H}$ of size $K \times K$ which is obtained by concatenating several columns from the history of activations **h**. Here, K is the number of bands. Time instances are not sampled consecutively, but strided in an exponential range, i.e. $n, n^2, \ldots n^K$. Finally, the diagonal elements of the dot product of two triangular matrices form the recurrent contribution to the vector $\mathbf{h}^{(t)}$. The feedforward contribution is calculated in the same way as in a standard RNN.

The practical implementation of the lower-triangular matrix containing the history of previous hidden activations in the DCWRNN requires usage of an additional memory buffer whose size can be given as $m = \sum_{k=1}^{K} |\mathbf{h}_k|(n^{k-1}-1)$, whereas here we have stated the general case of $|\mathbf{h}_k| \geq 1$ hidden units belonging to band k.

During training, updating all bands at a constant rate is important for preventing simultaneous overfitting of high-frequency and underfitting of low-frequency bands. In practice it leads to a speedup of the training process and improved performance. Finally, due to the constant update rate of all bands in the dense network, the learned representations are invariant to local shifts in the input signal, which is crucial in unconstrained settings when the input is unsegmented. This is demonstrated in Section VII.

VI. DATA COLLECTION

The dataset introduced in this work is a part of a more general multi-modal data collection effort performed by Google ATAP, known as Project Abacus. To facilitate the research, we worked with a third party vendor's panel to recruit and obtain consent from volunteers and provide them with LG Nexus 5 research phones which had a specialized read only memory (ROM) for data collection. Volunteers had complete control of their data throughout its collection, as well as the ability to review and delete it before sharing for research. Further, volunteers could opt out after the fact and request that all of their data be deleted. The third party vendor acted as a privacy buffer between Google ATAP and the volunteers.

The data corpus consisted of 27.62 TB of smartphone sensor signals, including images from a front-facing camera, touchscreen, GPS, bluetooth, wifi, cell antennae, etc. The motion data was acquired from three sensors: accelerometer, gyroscope and magnetometer. This study included approximately 1,500 volunteers using the research phones as their primary devices on a daily basis. The data collection was completely passive and did not require any action from the volunteers in order to ensure that the data collected was representative of their regular usage.

Motion data was recorded from the moment after the phone was unlocked until the end of a session (i.e., until it is locked again). For this study, we set the sampling rate for the accelerometer and gyroscope sensors to 200 Hz and for the magnetometer to 5 Hz. However, to prevent the drain of a battery, the accelerometer and gyro data were not recorded when the device was at rest. This was done by defining two separate thresholds for signal magnitude in each channel. Finally, accelerometer and gyroscope streams were synchronized on hardware timestamps.

Even though the sampling rate of the accelerometer and the gyroscope was set to 200 Hz for the study, we noticed that intervals between readings coming from different devices varied slightly. To eliminate these differences and decrease power consumption, for our research we resampled all data to 50 Hz. For the following experiments, data from 587 devices were used for discriminative feature extraction and training of the universal background models, 150 devices formed the validation set for tuning hyperparameters, and another 150 devices represented ''clients'' for testing.

VII. EXPERIMENTAL RESULTS

In this section, we use an existing but relatively small inertial dataset to demonstrate the ability of the proposed DCWRNN to learn shift-invariant representations. We then describe our study involving a large-scale dataset which was collected ''in the wild''.

A. VISUALIZATION: HMOG DATASET

To explore the nature of inertial sensor signals, we performed a preliminary analysis on the HMOG dataset [33] containing similar data, but collected in constrained settings as a part of a lab study. This data collection was performed with the help of 100 volunteers, each performing 24 sessions of predefined tasks, such as reading, typing and navigation, while sitting or walking.

Unfortunately, direct application of the whole pipeline to this corpus is not so relevant due to 1) absence of task-to-task transitions in a single session and 2) insufficient data to form separate subsets for feature learning, the background model, client-specific subsets for enrollment, and still reserve a separate subset of ''impostors'' for testing that haven't been seen during training.

However, a detailed visual analysis of accelerometer and gyroscope streams has proven that the inertial data can be seen as a combination of periodic and quasi-periodic signals (from walking, typing, natural body rhythms and noise), as well non-periodic movements. This observation additionally motivates the clockwork-like architectures allowing for explicit modelling of periodic components.

In this subsection, we describe the use of HMOG data to explore the shift-invariance of temporal models that do not have explicit reset gates (i.e. RNN, CWRNN and DCWRNN). For our experiment, we randomly selected 200 sequences of normalized accelerometer magnitudes and applied three different networks each having 8 hidden units and a single output neuron. All weights of all networks were initialized randomly from a normal distribution with a fixed seed. For both clockwork architectures we used a base 2 exponential setting rule and 8 bands.

Finally, for each network we performed 128 runs (i.e. 2^{K-1}) on a shifted input: for each run x the beginning of the sequence was padded with $x-1$ zeros. The resulting hidden activations were then shifted back to the initial

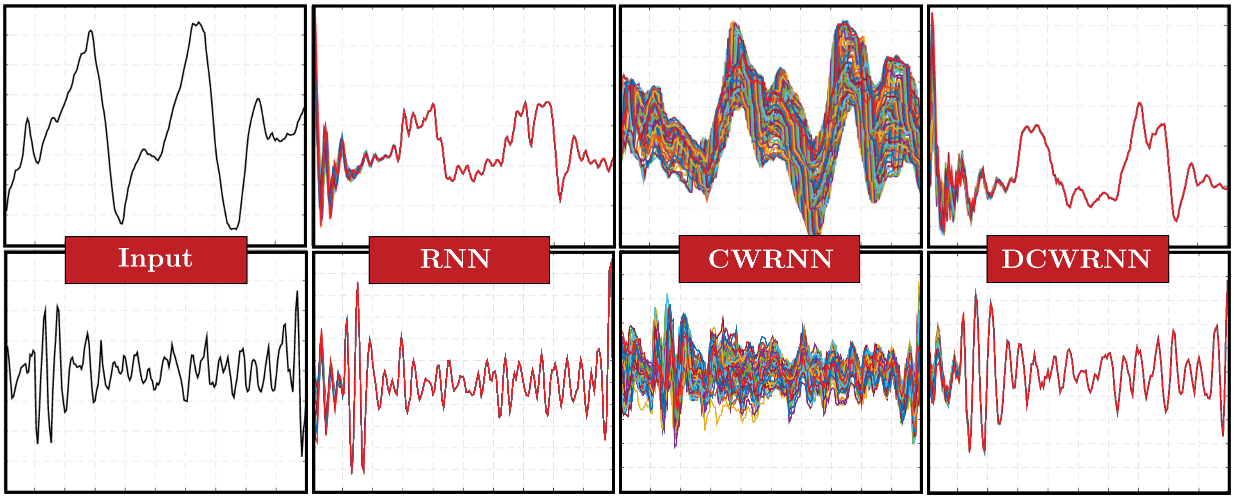

FIGURE 6. On spatial invariance. From left to right: original sequence and traces of RNN, CWRNN and DCWRNN units. The first row: reading while walking, the second row: typing while sitting.

position and superimposed. Fig. 6 visualizes the hidden unit traces for two sample sequences from the HMOG dataset, corresponding to two different activities: reading while walking and writing while sitting. The figure shows that the RNN and the dense version of the clockwork network can be considered shift-invariant (all curves overlap almost everywhere, except for minor perturbance at the beginning of the sequence and around narrow peaks), while output of the CWRNN is highly shift-dependent.

For this reason, in spite of their atractiveness in the context of multi-scale periodic and non-periodic signals, the usage of the CWRNN for the purpose of feature learning from unsegmented data may be suboptimal due to high shift-associated distortion of learned distributions, which is not the case for DCWRNNs.

B. LARGE-SCALE STUDY: GOOGLE ABACUS DATASET

We now evaluate our proposed authentication framework on the real-world dataset described in Section VI. Table 4 in the Appendix provides architectural hyper-parameters chosen for two 1-d Convnets (abbreviated ST and LT for short and log-term) as well as the recurrent models. To make a fair comparison, we set the number of parameters to be approximately the same for all of the RNNs. The ST Convnet is trained on sequences of 50 samples (corresponding to a 1s data stream), LT Convnets take as input 500 samples (i.e. 10s). All RNN architectures are trained on sequences of 20 blocks of 50 samples each with 50% of inter-block overlap to ensure smooth transitions between blocks (therefore, also a 10s duration). For the dense and sparse clockwork architectures we set the number of bands to 3 with a base of 2. All layers in all architectures use tanh activations. During training, the networks produce a softmax output per block in the sequence, rather than only for the last one. The mean per-block negative log likelihood loss taken over all blocks is minimized.

The dimensionality of the feature space produced by each of the networks is PCA-reduced to 100. GMMs with 256 mixture components are trained for 100 iterations after initialization with k-means (100 iterations). MAP adaptation for each device is performed in 5 iterations with a relevance factor of 4 (set empirically). For zt-score normalization, we exploit data from the same training set and create 200 t-models and 200 z-sequences from non-overlapping subsets. Each t-model is trained based on UBM and MAP adaptation. All hyper-parameters were optimized on the validation set.

The networks are trained using stochastic gradient descent, dropout in fully connected layers, and negative log likelihood loss. In the temporal architectures we add a mean pooling layer before applying the softmax. Each element of the input is normalized to zero mean and unit variance. All deep nets were implemented with Theano [3] and trained on 8 Nvidia Tesla K80 GPUs. UBM-GMMs were trained with the Bob toolbox [1] and did not employ GPUs.

1) FEATURE EXTRACTION

We first performed a quantitative evaluation of the effectiveness of feature extractors alone as a multi-class classification problem, where one class corresponds to one of 587 devices from the training set. This way, one class is meant to correspond to one "user", which is equal to "device" in the training data (assuming devices do not change hands). To justify this assumption, we manually annotated periods of non-authentic usage based on input from the smartphone camera and excluded those sessions from the test and training sets. Experiments showed that the percentage of such sessions is insignificant and their presence in the training data has almost no effect on the classification performance.

Note that for this test, the generative model was not considered and the feature extractor was simply evaluated in terms of classification accuracy. To define accuracy, we must consider that human kinematics sensed by a mobile device can be considered as a weak biometric and used to perform a soft clustering of users in behavioral groups. To evaluate the quality of each feature extractor in the classification scenario, for each session we obtained aggregated probabilities over target classes and selected the 5% of classes with highest probability (in the case of 587 classes, the top 5% corresponds to the 29 classes). After that, the user behavior was considered to be interpreted correctly if the ground truth label was among them.

TABLE 1. Performance and model complexity of the feature extractors. These results assume one user per device and accuracy is defined based on whether or not the user is in the top 5% of classes according to the output distribution.

Feature learning: evaluation		
Model	Accuracy, %	# parameters
ST Convnet	37.13	6 102 137
LT Convnet	56.46	6 102 137
Conv-RNN	64.57	1 960 295
Conv-CWRNN	68.83	1 964 254
Conv-LSTM	68.92	1 965 403
Conv-DCWRNN	**69.41**	**1 964 254**

The accuracy obtained with each type of deep network with its corresponding number of parameters is reported in Table 1. These results show that the feed forward convolutional architectures generally perform poorly, while among the temporal models the proposed dense clockwork mechanism Conv-DCWRNN appeared to be the most effective, while the original clockwork network (Conv-CWRNN) was slightly outperformed by the LSTM.

2) AUTHENTICATION EVALUATION

when moving to the binary authentication problem, an optimal balance of false rejection and false acceptance rates, which is not captured by classification accuracy, becomes particularly important. We use a validation subset to optimize the generative model for the minimal equal error rate (EER). The obtained threshold value θ_{EER} is then used to evaluate performance on the test set using the half total error rate (HTER) as a criterion: HTER $= 1/2[\text{FAR}(\theta_{EER}) + \text{FRR}(\theta_{EER})]$, where FAR and FRR are false acceptance and false rejection rates, respectively. For the validation set, we also provide an average of per-device and per-session EERs (obtained by optimizing the threshold for each device/session separately) to indicate the upper bound of performance in the case of perfect score normalization (see italicized rows in Table 2).

An EER of 20% means that 80% of the time the correct user is using the device, s/he is authenticated, only by the way s/he moves and holds the phone, not necessarily interacting with it. It also means that 80% of the time the system identifies the user, it was the correct one. These results align well with the estimated quality of feature extraction in each case and show that the context-aware features can be efficiently incorporated in a generative setting.

TABLE 2. Performance of the GMM-based biometric model using different types of deep neural architectures. EER is given on the validation set, while HTER is estimated on the final test set using the same threshold.

User authentication: evaluation		
Model	EER, %	HTER, %
Raw features	36.21	42.17
ST Convnet	32.44	34.89
LT Convnet	28.15	29.01
Conv-RNN	22.32	22.49
Conv-CWRNN	21.52	21.92
Conv-LSTM	21.13	21.41
Conv-DCWRNN	20.01	20.52
Conv-DCWRNN, zt-norm	**18.17**	**19.29**
Conv-DCWRNN (per device)	15.84	16.13
Conv-DCWRNN (per session)	8.82	9.37

To compare the GMM performance with a traditional approach of retraining, or finetuning a separate deep model for each device (even if not applicable in a mobile setting), we randomly drew 10 devices from the validation set and replaced the output layer of the pretrained LSTM feature extractor with a binary logistic regression. The average performance on this small subset was 2% inferior with respect to the GMM, due to overfitting of the enrollment data and poor generalization to unobserved activities. This is efficiently handled by mean-only MAP adaptation of a general distribution in the probabilistic setting.

Another natural question is whether the proposed model learns something specific to the user "style" of performing tasks rather than a typical sequence of tasks itself. To explore this, we performed additional tests by extracting parts of each session where all users interacted with the same application (a popular mail client, a messenger and a social network application). We observed that the results were almost identical to the ones previously obtained on the whole dataset, indicating low correlation with a particular activity.

VIII. MODEL ADAPTATION FOR A VISUAL CONTEXT

Finally, we would like to stress that the proposed DCWRNN framework can also be applied to other sequence modeling tasks, including the visual context. The described model is not specific to the data type and there is no particular reason why it cannot be applied to the general human kinematic problem (such as, for example, action or gesture recognition from motion capture).

To support this claim, we have conducted additional tests of the proposed method within a task of *visual gesture recognition*. Namely, we provide results on the *motion capture (mocap)* modality of the *ChaLearn 2014 Looking at People* gesture dataset [28]. This dataset contains about 14000 instances of Italian conversational gestures with the aim to detect, recognize and localize gestures in continuous noisy recordings. Generally, this corpus

TABLE 3. **Performance of the proposed DCWRNN architecture on the *Chalearn 2014 Looking at People dataset* (mocap modality). Network parameters: input 183×9, conv. layer 25×3×1, 2 fully connected layers with 700 units, the recurrent layer (RNN-280, CWRNN-300, DCWRNN-300, Small LSTM-88, Large LSTM-300), 21 output class.**

ChaLearn 2014: sequential learning			
Model	Jaccard Index	Accuracy	N parameters
Single network [21]	0.827	91.62	1 384 621
Ensemble [21]	0.831	91.96	4 153 863
Conv-RNN	0.826	91.79	3 974 581
Small Conv-LSTM	0.815	91.50	3 976 863
Large Conv-LSTM	0.825	91.89	4 900 621
Conv-CWRNN	0.834	92.38	3 972 496
Conv-DCWRNN	**0.841**	**93.02**	3 972 496

comprises multimodal data captured with the Kinect and therefore includes RGB video, depth stream and mocap data. However, only the last channel is used in this round of experiments. The model evaluation is performed using the Jaccard index, penalizing for errors in classification as well as imprecise localization.

Direct application of the GMM to gesture recognition is suboptimal (as the vocabulary is rather small and defined in advance), therefore, in this task we perform end-to-end discriminative training of each model to evaluate the effectiveness of *feature extraction* with the Dense CWRNN model.

In the spirit of [21] (the method ranked firstst in the ECCV 2014 ChaLearn competition), we use the same skeleton descriptor as input. However, as in the described authentication framework, the input is fed into a convolutional temporal architecture instead of directly concatenating frames in a spatio-temporal volume. The final aggregation and localization step correspond to [21]. Table 3 reports both the Jaccard index and per-sequence classification accuracy and shows that in this application, the proposed DCWRNN also outperforms the alternative solutions.

IX. CONCLUSION

From a modeling perspective, this work has demonstrated that temporal architectures are particularly efficient for learning of dynamic features from a large corpus of noisy temporal signals, and that the learned representations can be further incorporated in a generative setting. With respect to the particular application, we have confirmed that natural human kinematics convey necessary information about person identity and therefore can be useful for user authentication on mobile devices. The obtained results look particularly promising, given the fact that the system is completely non-intrusive and non-cooperative, i.e. does not require any effort from the user's side.

Non-standard weak biometrics are particularly interesting for providing the context in, for example, face recognition or speaker verification scenarios. Further augmentation with data extracted from keystroke and touch patterns, user location, connectivity and application statistics (ongoing work) may be a key to creating the first secure non-obtrusive mobile authentication framework.

Finally, in the additional round of experiments, we have demonstrated that the proposed Dense Clockwork RNN can be successfully applied to other tasks based on analysis of sequential data, such as gesture recognition from visual input.

APPENDIX

In this section, we provide additional detail for reproducability that was not provided in the main text.

A. HYPER-PARAMETER SELECTION

Table 4 provides the complete set of hyper-parameters that were chosen based on a held-out validation set. For convolutional nets, we distinguish between convolutional layers (Conv, which include pooling) and fully-connected layers (FCL). For recurrent models, we report the total number of units (in the case of CWRNN and DCWRNN, over all bands).

B. DETAILS ON zt-NORMALIZATION

Here we provide details on the zt-normalization that were not given in Section 3. Recall that we estimate authenticity, given a set of motion features by scoring the features against a universal background model (UBM) and client model. Specifically, we threshold the log-likelihood ratio in Eq. 7. An offline z-step (zero normalization) compensates for inter-model variation by normalizing the scores produced by each client model to have zero mean and unit variance in order

TABLE 4. **Hyper-parameters: values in parentheses are for short-term (ST) convnets when different from long-term (LT).**

Layer	Filter size / # of units	Pooling	Filter size / # of units	Pooling
	Convolutional feature learning		Sequential feature learning	
Input	500×14 (50 × 14)	-	10×50×14	-
Conv1	25×9×1	8×1 (2×1)	25×7×1	2×1
Conv2	25×9×1	4×1 (1×1)	25×7×1	2×1
Conv3	25×9×1	1×1	25×7×1	1×1
FCL1	2000	-	-	-
FCL2	1000	-	-	-
Recurrent	-	-	RNN 894, LSTM 394, CWRNN and DCWRNN 1000	- -

to use a single global threshold:

$$\Lambda_z(Y|\Theta_{\text{client}}) = \frac{\Lambda(Y) - \mu(Z|\Theta_{\text{client}})}{\sigma(Z|\Theta_{\text{client}})}, \quad (10)$$

where Y is a test session and Z is a set of impostor sessions. Parameters are defined for a given user once model enrollment is completed. Then, the $\mathcal{T}$-norm (test normalization) compensates for inter-session differences by scoring a session against a set of background $\mathcal{T}$-models.

$$\Lambda_{zt}(Y) = \frac{\Lambda_z(Y|\Theta_{\text{client}}) - \mu_z(Y|\Theta_{\mathcal{T}})}{\sigma_z(Y|\Theta_{\mathcal{T}})}. \quad (11)$$

The T-models are typically obtained through MAP-adaptation from the universal background model in the same way as all client models, but using different subsets of the training corpus. The Z-sequences are taken from a part of the training data which is not used by the T-models.

ACKNOWLEDGMENT

The authors would like to thank their colleagues Elie Khoury, Laurent El Shafey, Sébastien Marcel and Anupam Das for valuable discussions.

REFERENCES

[1] A. Anjos, L. El Shafey, R. Wallace, and M. Günther, C. McCool, and S. Marcel, “Bob: A free signal processing and machine learning toolbox for researchers,” in *Proc. 20th ACM MM*, 2012, pp. 1449–1452.

[2] R. Auckenthaler, M. Carey, and H. Lloyd-Thomas, “Score normalization for text-independent speaker verification systems,” *Digit. Signal Process.*, vol. 10, no. 1, pp. 42–54, 2000.

[3] J. Bergstra *et al.*, “Theano: A CPU and GPU math compiler in Python,” in *Proc. Python Sci. Comput. Conf. (SciPy)*, 2010, pp. 1–7.

[4] S. Berlemont, G. Lefebvre, S. Duffner, and C. Garcia, “Siamese neural network based similarity metric for inertial gesture classification and rejection,” in *Proc. FG*, 2015, pp. 1–6.

[5] S. Bhattacharya, P. Nurmi, N. Hammerla, and T. Plotz, “Using unlabeled data in a sparse-coding framework for human activity recognition,” *Pervasive Mobile Comput.*, vol. 15, pp. 242–262, Dec. 2014.

[6] C. Bo, L. Zhang, X.-Y. Li, Q. Huang, and Y. Wang, “SilentSense: Silent user identification via touch and movement behavioral biometrics,” in *Proc. MobiCom*, 2013, pp. 187–190.

[7] A. Bulling, U. Blanke, and B. Schiele, “A tutorial on human activity recognition using body-worn inertial sensors,” *ACM Comput. Surv.*, vol. 46, no. 3, 2014, Art. no. 33.

[8] A. Das, N. Borisov, and M. Caesar. (2015). “Exploring ways to mitigate sensor-based smartphone fingerprinting.” [Online]. Available: http://arxiv.org/abs/1503.01874

[9] M. O. Derawi, C. Nickel, P. Bours, and C. Busch, “Unobtrusive user-authentication on mobile phones using biometric gait recognition,” in *Proc. IIH-MSP*, 2010, pp. 306–311.

[10] J. Donahue *et al.* (2015). “Long-term recurrent convolutional networks for visual recognition and description.” [Online]. Available: http://arxiv.org/abs/1411.4389

[11] S. Duffner, S. Berlemont, G. Lefebre, and C. Garcia, “3D gesture classification with convolutional neural networks,” in *Proc. ICASSP*, 2014, pp. 5432–5436.

[12] D. Figo, P. C. Diniz, D. R. Ferreira, and J. M. P. Cardoso, “Preprocessing techniques for context recognition from accelerometer data,” *Pers. Ubiquitous Comput.*, vol. 14, no. 7, pp. 645–662, 2010.

[13] A. Hannun *et al.* (2014). “Deep speech: Scaling up end-to-end speech recognition.” [Online]. Available: http://arxiv.org/abs/1412.5567

[14] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[15] S. E. Kahou *et al.*, “Combining modality specific deep neural networks for emotion recognition in video,” in *Proc. ICMI*, 2013, pp. 543–550.

[16] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, “Large-scale video classification with convolutional neural networks,” in *Proc. CVPR*, 2014, pp. 1725–1732.

[17] E. Khoury, L. El Shafey, C. McCool, M. Gunther, and S. Marcel, “Bi-modal biometric authentication on mobile phones in challenging conditions,” *Image Vis. Comput.*, vol. 32, no. 12, pp. 1147–1160, 2014.

[18] J. Koutnik, K. Greff, F. Gomez, and J. Schmidhuber, “A clockwork RNN,” in *Proc. 31st ICML*, 2014, pp. 1863–1871.

[19] G. Lefebvre, S. Berlemont, F. Mamalet, and C. Garcia, “BLSTM-RNN based 3D gesture classification,” in *Proc. 22nd ICANN*, 2013, pp. 381–388.

[20] J. R. Napier, “The prehensile movements of the human hand,” *J. Bone Joint Surgery*, vol. 38-B, no. 4, pp. 902–913, 1956.

[21] N. Neverova, C. Wolf, G. W. Taylor, and F. Nebout, “ModDrop: Adaptive multi-modal gesture recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&arnumber=7169562

[22] J. Ngiam, A. Khosla, M. Kin, J. Nam, H. Lee, and A. Y. Ng, “Multimodal deep learning,” in *Proc. 28th ICML*, 2011, pp. 1–8.

[23] C. Nickel, H. Brandt, and C. Busch, “Benchmarking the performance of svms and hmms for accelerometer-based biometric gait recognition,” in *Proc. ISSPIT*, 2011, pp. 281–286.

[24] *People With Phones at the Station—Walking and Talking*, accessed on Nov. 15, 2015. [Online]. Available: https://www.flickr.com/photos/infomastern/16275617512/in/photolist-qNdN43-qvR6ht-qvSsqr-quRXrC-qS9eNN-vRtLdL-rrK5jM-qv5KHv-quS3Ko-rrP3iA-krP8vt-rrTJwr-Ew8J2C-C77CYL-uxdvJ1

[25] T. Plötz, N. Y. Hammerla, and P. Olivier, “Feature learning for activity recognition in ubiquitous computing,” in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, 2011, pp. 1729–1734.

[26] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, “Speaker verification using adapted Gaussian mixture models,” *Digit. Signal Process.*, vol. 10, nos. 1–3, pp. 19–41, 2000.

[27] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, “Convolutional, long short-term memory, fully connected deep neural networks,” in *Proc. ICASSP*, 2015, pp. 4580–4584.

[28] S. Escalera *et al.*, “ChaLearn looking at people challenge 2014: Dataset and results,” in *Proc. ECCVW*, 2014, pp. 459–473.

[29] Z. Sitova *et al.* (2015). “HMOG: New behavioral biometric features for continuous authentication of smartphone users.” [Online]. Available: http://arxiv.org/abs/1501.01199

[30] N. Srivastava and R. Salakhutdinov, “Multimodal learning with deep Boltzmann machines,” in *Proc. NIPS*, 2013, pp. 2222–2230.

[31] G. W. Taylor, G. E. Hinton, and S. T. Roweis, “Two distributed-state models for generating high-dimensional time series,” *J. Mach. Learn. Res.*, vol. 12, pp. 1025–1068, Mar. 2011.

[32] E. Vildjiounaite *et al.*, “Unobtrusive multimodal biometrics for ensuring privacy and information security with personal devices,” in *Proc. Pervasive Comput.*, 2006, pp. 187–201.

[33] Q. Yang *et al.*, “A multimodal data set for evaluating continuous authentication performance in smartphones,” in *Proc. 12th ACM SenSys*, 2014, pp. 358–359.

[34] M. Zeng *et al.*, “Convolutional neural networks for human activity recognition using mobile sensors,” in *Proc. 6th Int. Conf. MobiCASE*, 2014, pp. 197–205.

NATALIA NEVEROVA is currently pursuing the Ph.D. candidate degree with the Institut National des Sciences Appliquées de Lyon and LIRIS (CNRS, France) working in the area of gesture and action recognition with an emphasis on multimodal aspects and deep learning methods. She is advised by Christian Wolf and Graham Taylor. Her research is part of Interabot project in partnership with Awabot SAS. She was a Visiting Researcher with the University of Guelph in 2014 and at Google in 2015.

CHRISTIAN WOLF received the M.Sc. degree in computer science from the Vienna University of Technology, in 2000, the Ph.D. degree from the Institut National des Sciences Appliquées de Lyon (INSA de Lyon), in 2003, and the Habilitation Diploma degree in 2012. He has been an Assistant Professor with the INSA de Lyon and LIRIS, CNRS, since 2005. He is interested in computer vision and machine learning, especially the visual analysis of complex scenes in motion, structured models, graphical models, and deep learning.

GRIFFIN LACEY is currently pursuing the M.A.S. degree with the University of Guelph. His primary research interests are in developing tools and techniques to support deep learning on FPGAs. Recently, he acted as a Visiting Researcher with Google. He is a BENG in Engineering Systems and Computing with the University of Guelph.

LEX FRIDMAN received the B.S., M.S., and Ph.D. degrees from Drexel University. He is a Post-Doctoral Associate with the Massachusetts Institute of Technology. His research interests include machine learning, decision fusion, and computer vision applied especially to detection and analysis of human behavior in semi-autonomous vehicles.

DEEPAK CHANDRA heads authentication with the Machine Intelligence and Research group at Google. The project aims at completely redefining authentication for digital and physical world. Prior to this, he was the Program Lead in Google's Advanced Technology and Projects organization, where he heads all product, engineering, and design for mobile authentication projects. He defined company wide authentication strategy for Motorola, prior to leading the efforts at Google. He has developed multiple wearable authentication products, including Motorola Skip and Digital Tattoo.

BRANDON BARBELLO is a Product Manager with Google, where he works on privacy-sensitive on-device machine learning. He was with Google Advanced Technology and Projects on the Project Abacus Team, where he managed efforts to develop a multimodal continuous authentication system for smartphones. Prior to Google, he co-founded four companies across electronics, fintech, and private equity.

GRAHAM TAYLOR received the Ph.D. degree from the University of Toronto, in 2009, where his thesis co-advisors were G. Hinton and S. Roweis. He is an Assistant Professor with the University of Guelph. He is interested in statistical machine learning and biologically-inspired computer vision, with an emphasis on unsupervised learning and time series analysis. He did a post-doctoral research with NYU with C. Bregler, R. Fergus, and Y. LeCun.