Evaluating keystroke dynamics as a biomarker for mental fatigue detection

Teresa Arroyo-Gallego (gallego@nq-medical.com)
nQ Medical Inc.
Alejandro Acien Ayala
nQ Medical
Aythami Morales
Autonomous University of Madrid
Ruben Vera-Rodriguez
Autonomous University of Madrid
Julian Fierrez
Autonomous University of Madrid
Ijah Mondesire-Crump
nQ Medical Inc.

Article

Keywords:

Posted Date: May 5th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1580509/v1

Evaluating keystroke dynamics as a biomarker for mental fatigue detection

Alejandro Acien[1,*], Aythami Morales[2], Ruben Vera-Rodriguez[2], Julian Fierrez[2], Ijah Mondesire-Crump[1], and Teresa Arroyo-Gallego[1,*]

[1]nQ Medical Inc., Cambridge, Massachusetts, USA
[2]Autonomous University of Madrid, BiDA-LAB, School of Engineering, Madrid, Spain
[*]alejandro.acien@uam.es
[*]gallego@nq-medical.com

ABSTRACT

In this paper we study the feasibility of employing keystroke biometrics for mental fatigue detection during natural typing. For this task, we employ TypeNet, a state-of-the-art deep neuronal network, originally intended for user authentication at large scale using keystroke dynamics. We adapted TypeNet for fatigue detection by leveraging the information embedded in TypeNet for person recognition, and applying that information to a different but related task as it is fatigue detection by employing domain adaptation techniques. All experiments were conducted using three keystroke databases that comprise different contexts and data collection protocols. Our preliminary results showed performances ranging between 72.2% and 80.0% for fatigue versus rested sample classification, which is aligned with previously published models on daily alertness and circadian cycles. This demonstrates the potential of our proposed system to characterize mental fatigue fluctuations via natural typing patterns. Finally, we studied the feasibility of an active detection approach that utilizes the continuous monitoring of keystroke biometric patterns for the real-time assessment of subject fatigue.

Mental fatigue is a state of brain exhaustion caused by long periods of cognitive activity, lack of sleep, or stress. According to Tanaka *et al.*[1], mental fatigue may lead to over-activation of the visual cortex in the occipital lobe, which has been linked to cognitive impairment and low psychomotor performance. Patients experiencing this condition usually report, among other symptoms, a reduction of their concentration capacity, headaches, dizziness and slowed reflexes and responses[2]. From a clinical point of view, these psychomotor impairments induced by mental fatigue could be a sign of other emerging diseases, including neurodegenerative or cardiovascular conditions[3,4]. As an example, Parkinson's disease patients have been reported to show higher level of physical and mental fatigue in early stages of the disease than healthy subjects[5]. Fatigue has been reported to be one of the major causes of disability for up to half of Parkinson's disease patients[6], limiting their ability to participate in daily routines or social activities[7,8].

Although multiple tools exist for the assessment of fatigue, there is no clinical standard that enables an objective and complete evaluation people's state in this domain. The most accepted method is the Fatigue Assessment Scale (FAS), a Patient Reported Outcome (PRO) comprised of 10 items that evaluate physical and physiological aspects of fatigue[9]. The subjective and episodic nature of these tools makes it difficult to detect and evaluate fatigue in daily practice and in the context of clinical trials. There is a clinical and research need to develop more accessible, accurate and specific biomarkers to monitor fatigue and its clinical causes[10,11].

Keystroke dynamics is a biometric trait commonly used to authenticate users based on their typing patterns[12,13]. The speed of pressing and releasing keys[14] or the pressure exerted when pressing a key[15] are some of these typing features used by keystroke biometric algorithms for user authentication. Finger kinematics during typing are fine motor skills ruled by the neuromotor cortex and have also been presented as a powerful biomarker in the diagnosis and monitoring of different neurodegenerative disease, including Parkinson's[16,17] and Alzheimer's disease[18]. Additionally, continuous keystroke data is easy to gather via commodity hardware (e.g., phones, laptops) without requiring the use of proprietary devices. Furthermore, remote data collection can avoid intrusive visits to the clinic, which enhances the patient's quality of life.

In this Article we study the applicability of keystroke dynamics as a potential biomarker of mental fatigue, going a step forward in the state-of-the-art characterization of this psychomotor condition by proposing a new active fatigue detection framework based on Deep Neuronal Networks (DNN). To develop this, we will employ TypeNet[19], a state-of-the-art DNN originally designed to model identity via typing patterns at large scale ($\sim$ 100,000 users). The main idea behind this work is to leverage the keystroke dynamics patterns learnt by TypeNet for user recognition, and re-optimize this network for the fatigue detection task. A schema of the proposed system is showed in Fig. 1. The system is comprised of three main elements: the

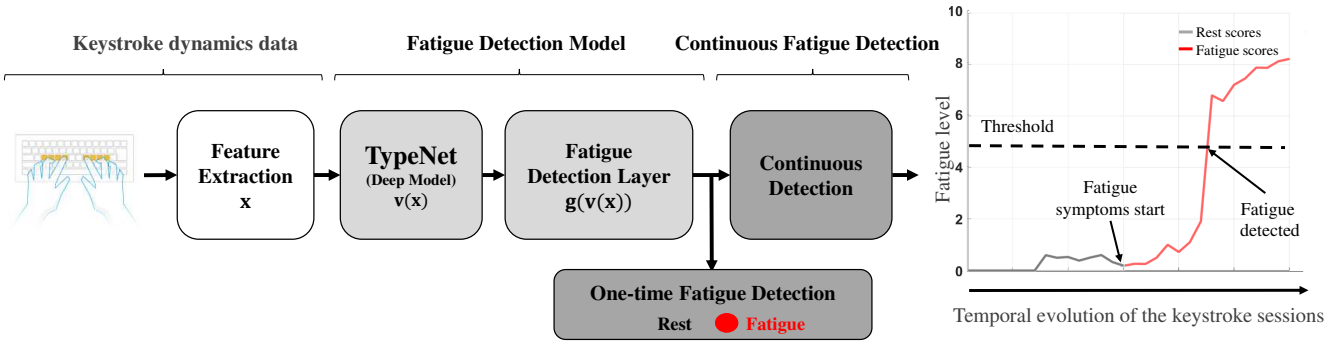

Figure 1. Block diagram of the entire system proposed. The fatigue detection layer adapts the TypeNet´s capacity to model user behavior through keystroke patterns for the fatigue detection task. This information is taken by the active detection algorithm to detect changes in users' fatigue level over consecutive keystroke sessions.

Database	#Subjects	#Sessions	Session Size	Supervised	Context
Aalto[20]	168K	15	∼70 keys	No	TypeNet development for general user typing model
neuroQWERTY Sleep Inertia (nQSI)[21]	14	4	15 min	Yes	Fatigue detection system development and evaluation
neuroQWERTY Crowdsource (nQCS)	251	∼1,000	∼3 min	No	Evaluation of fatigue detection in a real world environment

Table 1. List of study keystroke datasets

input layer, the fatigue detection model, and the post-processing module for active detection. The input layer ingests keystroke session data and generates a predefined feature vector that is then fed to the fatigue detection model. The fatigue detection model is created by connecting the output of the TypeNet network to a fatigue detection layer, which optimizes the original authentication model for fatigue identification. Finally, the post-processing module for active detection ingests the temporal sequences of fatigue detection scores to produce users' calibrated fatigue level on the basis of their baseline or previous fatigue states. This block enables real time monitoring of on-off fatigue fluctuations over consecutive keystroke sessions. In this work, we evaluate the proposed system in a controlled data context to test the performance of the fatigue detection model to discriminate between labeled rest and mental fatigue sessions. In addition to this, we present a real-world application of the system applied to natural typing data to evaluate its suitability to identify daily fatigue cycles in a healthy population.

The main contributions of this work are fourfold: *i)* we develop a deep neural network able to identify mental fatigue symptoms through keystroke patterns, *ii)* we analyze the ability of the proposed model to detect small variations in fatigue levels between different keystroke sessions, *iii)* we propose an active fatigue detection algorithm that continuously monitors users' keystroke session sequences to detect longitudinal variations in their fatigue state, and *iv)* we evaluate the applicability of the proposed system to detect fatigue trends in real-world user data.

Results

Datasets overview. Table 1 summarizes the main characteristics of the three keystroke databases that were employed in this work (see section Methods for more details):

- The Aalto database[20], comprises keystroke data from 168,000 subjects. This database was used in previous work to train and test the TypeNet DNN architecture for user authentication at large scale[19,22]. This database is used to model the typing patterns of a general population.
- The neuroQWERTY Sleep Inertia (nQSI) database[21], includes controlled typing data from a group of 14 healthy volunteers, each user provided two keystroke streams generated during fatigue and rested states. This database is used to adapt the TypeNet model to the fatigue detection task.
- The neuroQWERTY Crowdsource (nQCS) database is employed to test our system in a real world environment. For this analysis, we used data from 251 healthy volunteers included in the nQCS database that contributed the keystroke data from their daily interaction with their personal computer.

System design. The fatigue detection model is trained and tested with the labeled keystroke data from the nQSI database. As depicted in Fig. 2.a, the input of the TypeNet network is a keystroke feature vector **x** extracted from the raw keystroke data in

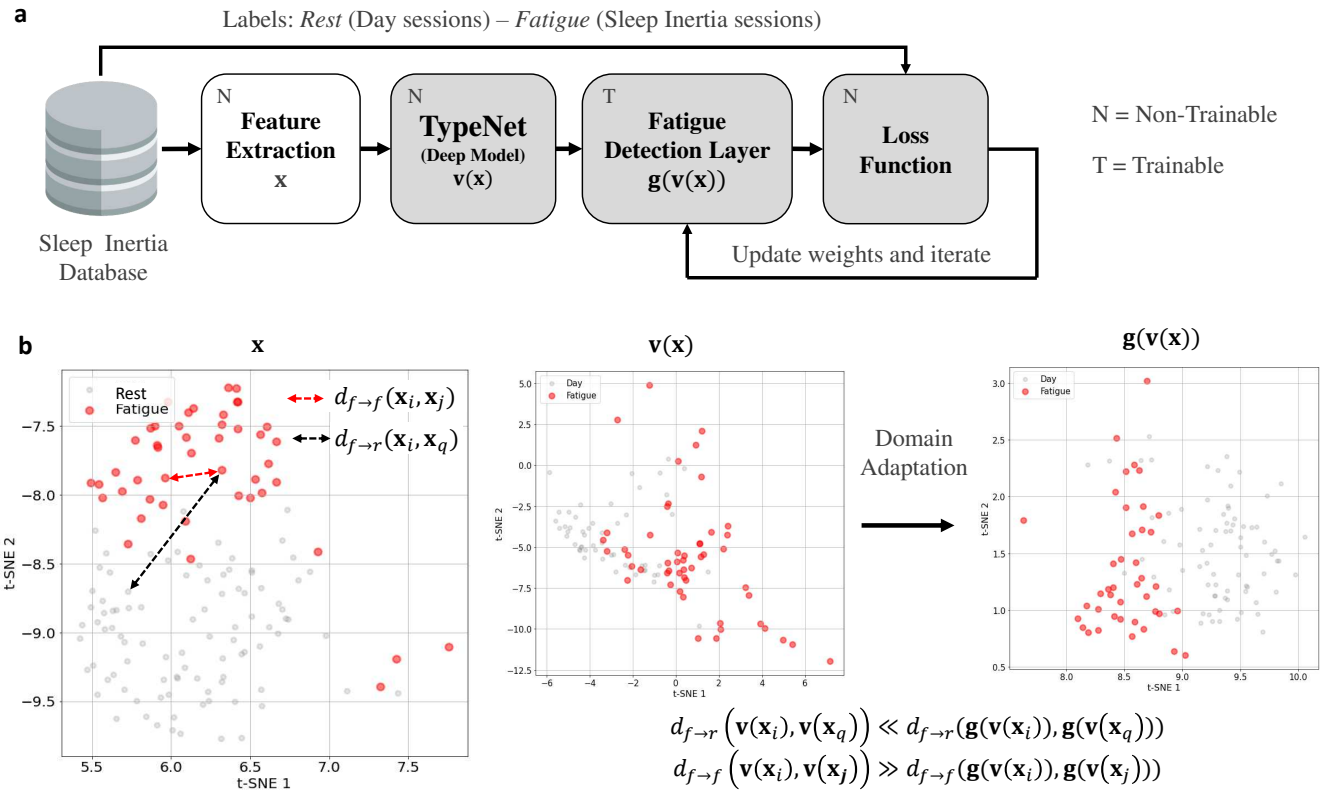

Figure 2. Overview of the fatigue detection model design. The fatigue detection model is trained with the labeled keystroke data from the nQSI database. At the output, the model separates the fatigue embedding vectors $\mathbf{g}(\mathbf{v}(\mathbf{x}))$ that correspond to each of the two user's states under study (i.e., fatigue or rest) while favouring proximity between the embedding vectors that belong to the same class. In **b** we show an example of the transformation from the embedding vectors generated by TypeNet $\mathbf{v}(\mathbf{x})$ to the fatigue embedding vectors generated at the output of the proposed model $\mathbf{g}(\mathbf{v}(\mathbf{x}))$. The sample output shown in this figure applies t-distributed Stochastic Neighbor Embedding (t-SNE) to generate a 2D projection of the 1×128 output.

the nQSI database. This vector includes 4 features corresponding to: (i) Hold Latency (HL), the elapsed time between key press and release events; (ii) Inter-key Latency (IL), the elapsed time between releasing a key and pressing the next key; (iii) Press Latency (PL), the elapsed time between two consecutive press events; and (iv) Release Latency (RL), the elapsed time between two consecutive release events. The output of TypeNet is a 1×128 dimensional embedding feature vector $\mathbf{v}(\mathbf{x})$ that authenticates users by applying a Distance Metric Learning method (DML)[23]. TypeNet was originally trained to model the typing patterns of $100,000$ users. The training process of TypeNet was aimed to generate a 128 dimensional feature space where keystroke events generated by the same user tend to cluster in a closer region of the feature space, while events from different users are projected in different areas of the same feature space. In this work, we use the nQSI dataset to adapt the transformed authentication feature space to the fatigue detection task. We apply domain adaptation techniques[24] based on the addition of a fatigue detection layer that is trained to transform the authentication-based feature vectors, $\mathbf{v}(\mathbf{x})$, into fatigue detection feature vectors with the same dimension, $\mathbf{g}(\mathbf{v}(\mathbf{x}))$ (as shown in Fig. 2.b). The fatigue detection layer is optimized using a Distance Metric Learning (DML) approach and a leave-one-out cross validation protocol (LOO-CV). Fig. 3 presents some examples showing the results of the transformation in the nQSI dataset. The hypothesis underlying the method is that the features learned to model the typing patterns $\mathbf{x}$ of 100,000 users contain useful information to characterize users' fatigue patterns. The fatigue detection layer serves as a non-linear transformation $\mathbf{g}()$ to reveal such patterns in the learned space $\mathbf{v}(\mathbf{x})$. The fatigue score is computed at the output of the fatigue detection model as the Euclidean distance between pairs of fatigue detection feature vectors,

$$d(\mathbf{x}_i,\mathbf{x}_j) = \left\|\mathbf{g}(\mathbf{v}(\mathbf{x}_i)) - \mathbf{g}(\mathbf{v}(\mathbf{x}_j))\right\| \tag{1}$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ are two keystroke samples generated by the same user. In this work, a sample integrates all keystrokes captured

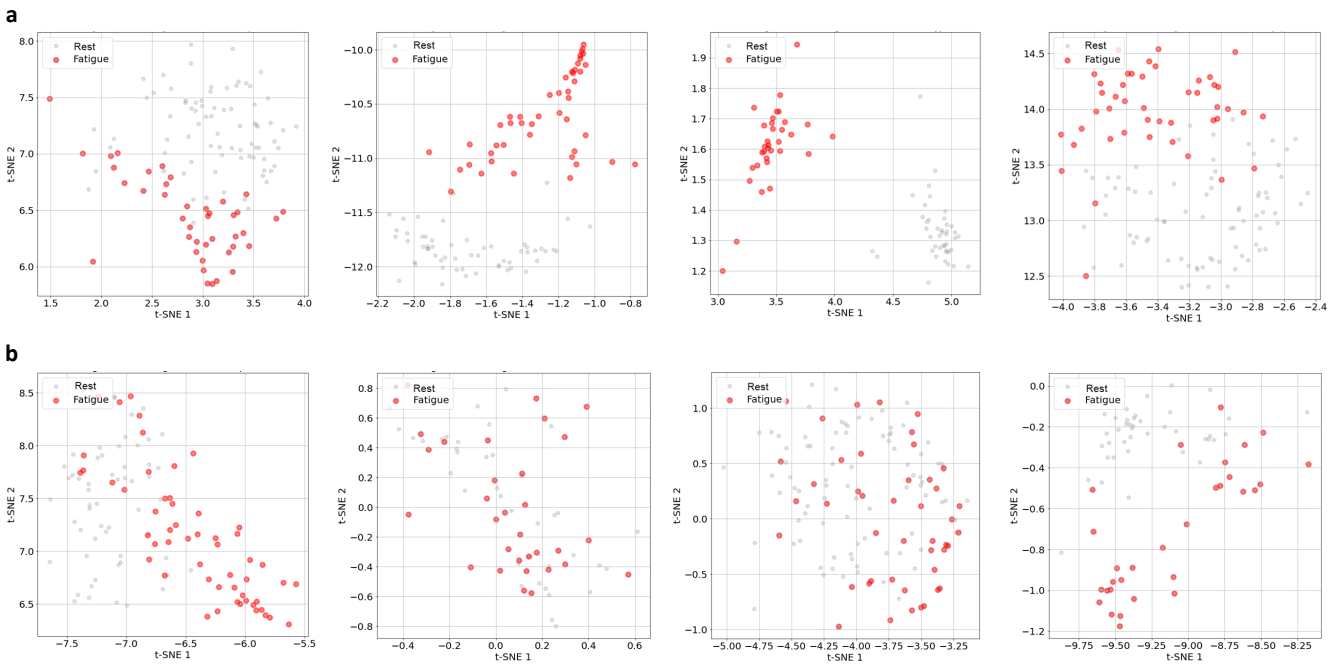

Figure 3. Intra-user variation of the embedding fatigue vectors $\mathbf{g}(\mathbf{v}(\mathbf{x}))$. We observe how the fatigue detection model presents varying performance depending on the user. Row **a** shows examples of fatigue embedding vectors for those subjects where we observe a good separation between fatigue and rest embedding vectors, while for subjects in the row **b** the separation is not as clear. This user-dependent performance could be a result of the varying levels of intra-user fluctuations observed during natural typing[25].

over a given typing session.

One-time Fatigue Detection Approach.To evaluate the performance of the fatigue detection model, we use the nQSI dataset to generate a pool of intra-user keystroke sample pairs. We contemplate a binary classification framework based on two scenarios: 1) no change: when the two samples belong to the same class (fatigue→fatigue or rest→rest); and 2) change: when the two samples belong two different classes (fatigue→rest or rest→fatigue). In Fig. 2 we can observe the distances for two examples: $d_{f\to f}(\mathbf{x}_i,\mathbf{x}_j)$ is the distance between two fatigue samples (no change, distance between two red dots) and $d_{f\to r}(\mathbf{x}_i,\mathbf{x}_q)$ is the distance between the fatigue and rest sample (change, distance between a red and a grey dots). The distance between samples is directly compared to a pre-defined threshold. A fatigue score superior to the threshold (note that we use distances therefore dissimilarity scores) reveals a change in the keystroke patterns, while a value below the threshold implies no change. We compare the performance of the fatigue detection model based on DML with different statistical classification algorithms trained with the feature vectors $\mathbf{x}$: Random Forest (RF), Support Vector Machine with Gaussian Kernel (SVM), k-Nearest Neighbours (k-NN), and the proposed fatigue detection model architecture replacing the DML by a softmax activation layer. Fig. 4 presents the Receiver Operating Characteristic (ROC) analysis comparison in two different set-ups. First, limiting the input size to 150 keystrokes per sample. This input format was defined in accordance to the design of the pre-trained TypeNet architecture. In this scenario, the best performance is achieved by the proposed fatigue detection model with that achieves an Area Under the Curve (AUC) of 72.1%, followed by the RF classifier with AUC of 68.4%. The worst performance is observed in the softmax-based variation of the proposed fatigue detection model.

We also propose a second set-up where we increase the input size to 5-minute long keystroke sessions (i.e., an average of $\sim$ 1,100 keys per sample) for the RF, SVM and k-NN classifiers, while keeping the original 150-keystroke long inputs for the proposed fatigue detection methods and its softmax variation (due to the limitation of 150 keys as input size of the TypeNet model). In this case the DML approach is slightly outperformed by the RF and SVM classifiers that present AUCs of 77.8% and 74.4%, respectively, in exchange of larger input data.

Continuous Fatigue Detection Approach. In this experiment we consider the Quick Change Detection (QCD) algorithm[26] that dynamically updates a confidence fatigue score by performing a cumulative sum from previously measured fatigue states. The purpose of this algorithm is to adapt the fatigue detection method to the needs posed by real time evaluation of fatigue in a

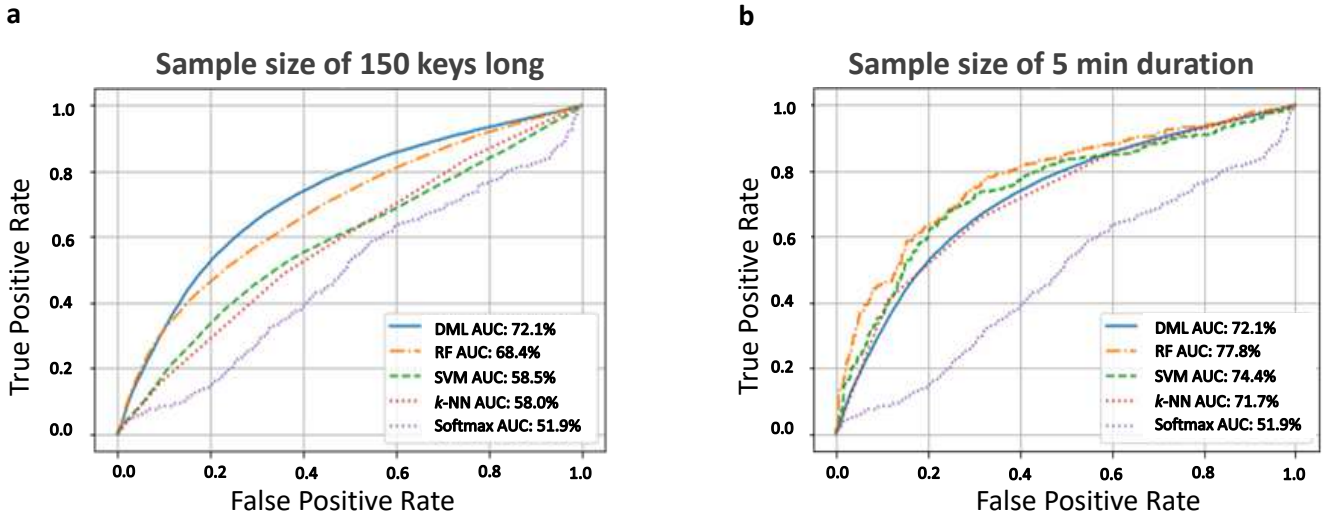

Figure 4. ROC analysis for the fatigue detection. AUC scores computed with keystroke sample pairs of length 150 keys (**a**) and 5 min duration (**b**). The ROC curves were calculated independently for each subject and the ROCs showed are the average of all of them.

real world environment.

In Fig. 5.a we show an example of the application of this algorithm at the output of the fatigue detection model. The example uses a simulated sequence of keystroke sessions generated by concatenating 15 rest and 15 fatigue keystroke samples from a user in the nQSI database. As the simulated sequence starts in a rest state, the initial fatigue scores are lower and close to zero during the first 15 evaluation intervals (i.e., the 15 user keystroke sessions labeled as rest in the nQSI database). As the simulated sequence starts introducing fatigue samples (from the remaining 15 user keystroke sessions labeled as fatigue), the active fatigue detection score tends to increase until reaches a certain threshold that would indicate there has been a fatigue state change. The number of keystroke sessions elapsed since the models starts getting fatigue samples until the active fatigue detection algorithm reaches the fatigue threshold is called Average Detection Delay (ADD). This parameter measures the number of keystroke sessions required to detect fatigue since the symptoms start.

The configuration of the threshold in the active fatigue detection score is crucial for the performance of the algorithm. As shown in Fig. 5.b, as we lower the threshold we reduce the ADD from 7 (Fig. 5.a) to 3 keystroke sessions, in exchange of a higher risk of false positives. This value is called Probability of False Detection (PFD) and measures the probability of false fatigue detection (similar to the False Match Rate). On the other hand, increasing the threshold controls the PFD at the cost of increasing the ADD as well as the Probability of Non Detection (PND). PND measures the probability of the active fatigue score never reaching the threshold over a sequence of keystroke sessions in a fatigued interval (See Fig. 5.c).

According to this, there is always a trade-off between the PND and PFD values as we move the threshold. Fig. 5.d shows the PND (left y-axis) versus PFD and ADD (right y-axis) versus PFD. Optimizing both specificity and sensitivity metrics at the same time we have the point Equal Error Rate (EER). The EER value is the point where the blue curve (i.e., PND vs PFD) crosses the diagonal (the dotted black line) and is equal to 20.0%. This would be equivalent to an AUC of $\text{AUC} = 100 - EER = 80.0\%$. Based on the configuration of the threshold, we can infer the number of fatigue keystroke sessions required, according to our results, to reach the threshold (i.e., the ADD value). Represented by the red curve (PFD vs ADD) in Fig. 5.d we see the number of sessions required for a scenario that uses the EER to define the threshold is slightly above 3.

Independent evaluation in real world environment. As mentioned above, the nQSI database employed to evaluate our system was acquired under supervised conditions with labeled keystroke sessions. To evaluate the behavior of the proposed method in the context of its intended use we applied the resulting model to the nQCS database. As a reminder, this database includes keystroke data from a group of healthy volunteers that was captured during their daily use of the device, without any supervision or prompt to stimulate typing activity. We compute the fatigue scores measured on each pair of consecutive keystroke sessions for each user typing stream. Each user typing stream is comprised of multiple keystroke sessions generated over varying observation periods and activity levels. We only take into account the fatigue scores obtained between keystroke sessions with elapsed time of less than 2 hours within the same day, in order to avoid long pauses between sessions that may introduce artifacts in the resulting fatigue signal.

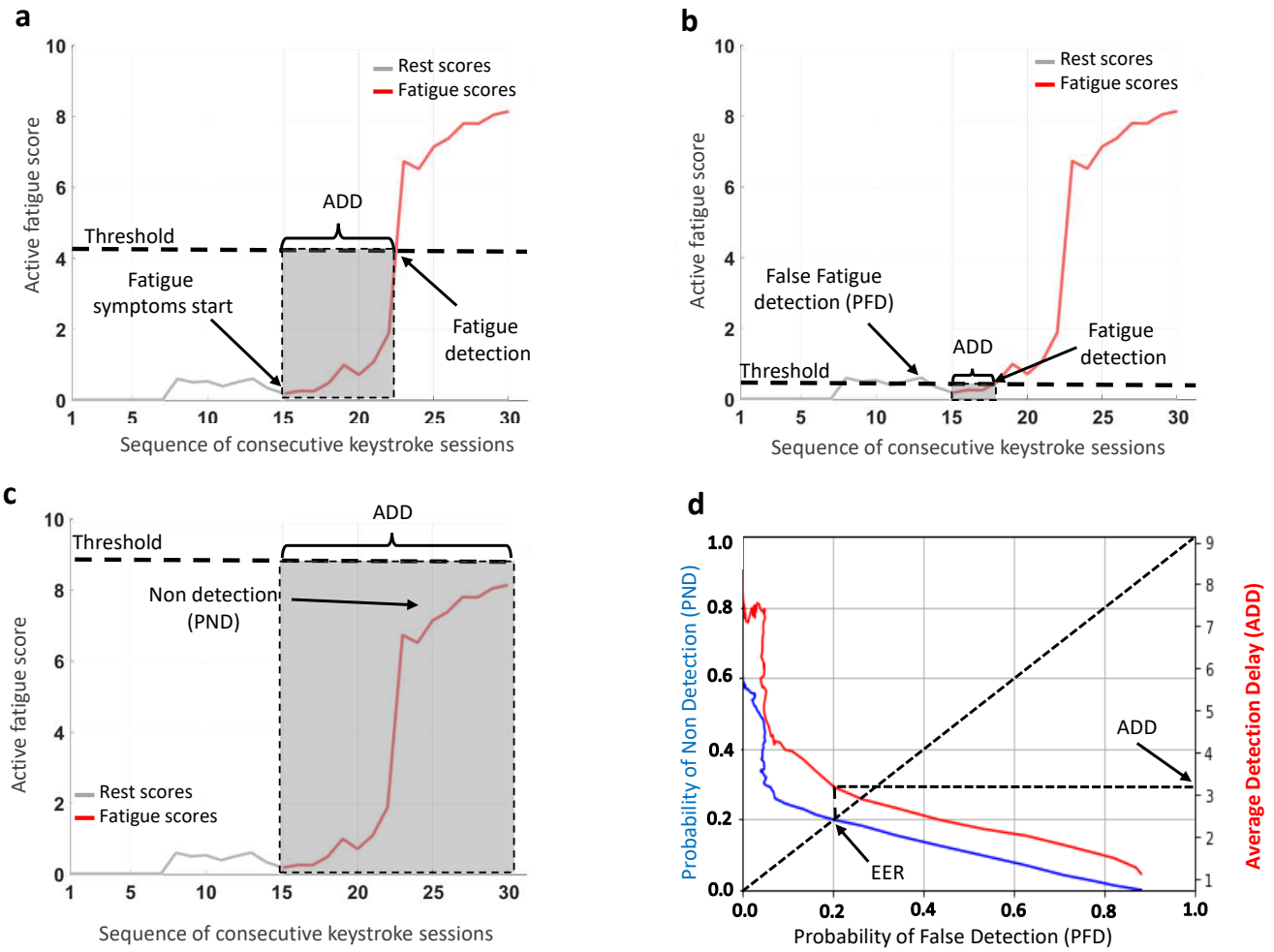

Figure 5. Active fatigue detection curves. a,**b**, and **c** are three different use cases of the active fatigue detection algorithm where the threshold chosen affects the performance. **d** shows the PFD vs PND and PFD vs ADD curves as a result of moving the threshold, the value chosen for the threshold is the point where both PFD and PND values are equal, called EER.

In Fig. 6 we present the aggregate trends of the fatigue score levels versus the time of the day. The results suggest lower fatigue levels during the morning and midday hours. Higher fatigue scores are observed during the afternoon hours and overnight. Note that this figure was obtained averaging the scores from all 251 volunteers in nQCS database and therefore, there is an equalization effect caused by the different user's habits.

Discussion

Using domain adaptation techniques, we leveraged an algorithm built for user authentication to detect signs of fatigue via natural typing. The resulting classifier was then adapted for real time fatigue monitoring by appending an active detection algorithm that compares successive user states. This allows for background evaluation of users' fatigue state in an objective and real-world environment. The proposed classifier was able to differentiate intra-patient fatigue vs rested states with an AUC of 72.1% in one-time detection setup or 80% detection accuracy in a continuous detection setup. A preliminary application of the fatigue classifier combined with active detection showcased its applicability to real-world data in a crowdsource dataset. Given that this method relies on data collected passively from a user's daily interactions with their computer, the proposed pipeline operates unobtrusively with low burden and allows for a background, objective evaluation of a user's fatigue state in the real-world environment.

Relying on machine learning techniques, we were able to liaise a large dataset created to study typing behaviors in the general population with the information gathered in a limited size dataset built specifically to characterize fatigue through the analysis of keystroke dynamics. This approach allowed us to apply a deep learning architecture in the absence of a high-dimensional dataset specifically characterized for the phenomenon under study, quantification of daily fatigue levels

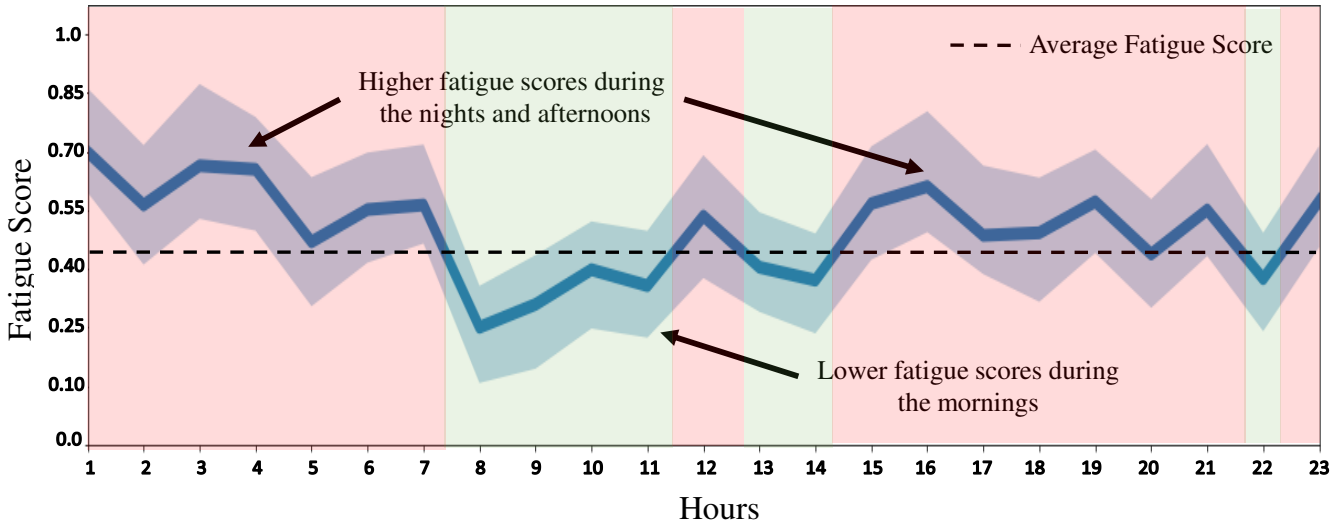

Figure 6. Fatigue score analysis in the nQCS database. The fatigue scores are calculated, at user level, between consecutive sessions over their daily typing activity. The graph presents the nQCS population aggregate average and confidence intervals of the resulting fatigue score daily sequences.

in users' keystroke patterns. Our work exhibits the potential of domain adaptation techniques to minimize the complexity of gathering large and curated data repositories required to train deep learning models by taking advantage of open-source unsupervised datasets in combination with much smaller supervised datasets. In our case, the Aalto database supplies the high volume of data required to build a network optimized for user authentication that is then fine-tuned using the Sleep Inertia dataset to solve the fatigue detection task. Another novel technical contribution of this work is the addition of an active detection algorithm that adapts the classifier for its application in real-time fatigue detection. This dynamic adaptation of the fatigue score threshold turns users into their own controls over time by carrying information from previous estimates to generate the present score. It is one of the main differentiators of this work from prior state-of-the-art approaches to this problem, which generally use a cross-sectional design to evaluate fatigue at a given time point[27–29].

The classification results in the controlled experiment (i.e. accuracy in the separation of rested vs fatigue samples in the Sleep Inertia dataset) are worse than the ones presented in previous work completed using the same dataset[21]. However, this approach reduces significantly the size of the input sample, 150 keystroke sequences (less than one minute at average speed) in comparison to the 15-minute long typing samples used in Giancardo et al. The value of this parameter is critical for the applicability of fatigue detection via keystroke monitoring in a real-world setting as users are unlikely to generate continuous 15-minute long typing samples on their daily use of computer keyboards. When applied on an independent dataset comprised by natural typing data collected in a real-world environment, the population-level results align with the results presented in previous studies on daily sleep and alertness cycles[30].

In general, our results suggest users are usually more awake and active during the mornings. Fatigue appears generally during the afternoon and increases as the day gets closer to regular sleep times. The daily averaged scores suggest a subtle fatigue peak after midday that could be associated with what has been referred in the literature as post-lunch dip in performance[31]. This consensus with sleep and performance studies supports our hypothesis that keystroke dynamics can be used to quantify in an objective and unobtrusive manner daily fatigue in computer users. However, it is important to note these daily cycle results have been analyzed at a population level and are not taking into account the variability in participants' personal routines and schedules. Future studies pairing keystroke with other high frequency fatigue related data (e.g. sleep, activity, etc.) could help us better assess the performance of the proposed method at user level.

As for its clinical application, fatigue is a common symptom that can precede or reflect the presence of a more serious mental or physical condition. The current standard to clinically assess fatigue relies on patient reported outcomes through standardized questionnaires, such as the Fatigue Severity Scale[32]. To identify fatigue as a symptom, patients must first identify unusually excessive fatigue patterns and then alert their physician before it can be further investigated. This leaves fatigue as a commonly overlooked or unrecognized predictor of other emerging disorders[3,4]. Fatigue has also been reported as a frequent side effect of disease treatment[33] and long-term sequel of conditions such as COVID-19[34].

The proposed methodology is designed to validate an approach for objective and passive fatigue monitoring. Leveraging the widespread use of personal computers, this framework presents an opportunity to provide more visibility and accurate tracking

of fatigue and its clinical implications. As it runs in the background of users' computers, this approach could potentially be used to alert patients and healthcare professionals of early signs of abnormal fatigue to uncover progressive disease or the presence of underlying conditions. In the context of clinical trials or during disease management, this method could also be used to enable objective and real-world evaluation of the impact of newly developed or existing treatment regimens on a patient's fatigue state.

As a major limitation of this work, the fatigue detection model performs better for some subjects than others due to the intra-user variations when typing[25]. In users who show little variation between resting and fatigue states, the model does not effectively classify performance. An example of this was showed in Fig. 3, where we can observe a clear separation between the rest keystroke sessions and the fatigue ones for the subjects (Fig. 3 **a**), meanwhile the fatigue detection model struggles trying to separate the keystroke sessions for the subjects in Fig. 3 **b** with poor results.

This work presents a step towards the development of a real-world fatigue monitoring tool that operates passively by leveraging users' natural interaction with their personal computers. It is important to note that the dataset used in these analyses is comprised solely by healthy controls, future work should evaluate the performance of the proposed method in a cohort that includes participants suffering from conditions impacting psychomotor health that may mask or be confounded by fatigue symptoms. Another limitation and potential line for future research is that this work has been tested using mechanical keyboard data, thus future applications of this specific methodology require users who type frequently on mechanical keyboard devices. Adapting this framework to include touchscreen devices would expand the population that could benefit from this method. Given that typing kinematics vary significantly between mechanical and touchscreen devices, this adaptation would require additional studies. The limited dimension of the Sleep Inertia database is another aspect to take into account in future studies. Although the use of domain adaptation techniques reduces the need for larger supervised datasets, increasing the size of the controlled cohort would allow for optimization of the target task layer and independent validation of the fatigue detection classifier. Finally, while the crowdsource results are similar to previously published studies on daily alertness, full validation would require a labeled real-world dataset to test the generalizability of the proposed framework for its application in the real work setting. Additional validation in specific use case scenarios would pave the way use of this method as an objective, high-resolution and quasi-continuous way to monitor users' fatigue with minimal burden on their daily routine.

Methods

Keystroke datasets. In this section we analyze in more detail the 3 keystroke databases employed in this work to train and evaluate our proposed system:

- The Aalto database[20] was employed to train the TypeNet model that we used as keystroke embedding feature extractor in our fatigue detection model. This database is comprised of 168,000 subjects with 15 keystroke sessions per subject. The database was acquired using an online questionnaire under an uncontrolled environment where each user employed their own physical keyboard. All user were initially informed of the acquisition of their press (keydown) and release (keyup) event timings during the completion of the questionnaire. The questionnaire required users: 1) To memorize a English sentence randomly chosen from a pool of 1,525 (Enron mobile email and Gigaword Newswire corpus). These sentences contained a minimum of 3 words and a maximum of 70 characters. And 2) Type them as quickly and accurately as they could the memorized sentence. All participants in the database completed 15 sessions (i.e. one sentence for each session) on either a desktop or a laptop physical keyboard. The authors of the database reported demographic statistics of the users: 72% of the participants took a typing course, 218 countries were involved, and 85% of them have English as native language. The richness of the Aalto database resides not only in the huge amount of participants acquired, but also in the diversity of ethnicities, countries and different typing skill levels of the participants enrolled (more details in `https://userinterfaces.aalto.fi/136Mkeystrokes/`), allowing TypeNet DNN to authenticate users thought keystroke dynamics at Internet scale with a high performance[22].

- The neuroQWERTY Sleep Inertia database[21] (nQSI) was designed to detect psychomotor impairment by waking up the participants during the night, thus inducing a sleep inertia status (a mental fatigue condition produced by lack of sleep). The database comprises 14 healthy subjects with 4 keystroke sessions per subject of 15 min duration collected in mechanical keyboards. Two of the keystroke sessions were captured during the day, whenever the subject felt well rested, labeling them as rest state (no fatigue). The other two keystroke sessions labeled as the fatigue ones were captured at midnight, when the subjects were woke up during the phase III/IV of the sleep cycle[35] to capture the keystroke sessions, thereby inducing the sleep inertia state. The acquisition process was monitored by the owners of the database to ensure the quality of the keystroke data captured in both rest and fatigue states (supervised scenario). We used this database to train and test our proposed system for the mental fatigue detection task trough keystroke dynamics. Subjects gave informed consent prior to experiments and experimental procedures were approved by COUHES (Committee On the Use of Humans as Experimental Subjects) at the Massachusetts Institute of Technology, protocol no. 1311.

- Finally, the neuroQWERTY Crowdsource database (nQCS) is composed by more than 800 subjects from a healthy control group and group of patients with self-reported neurodegenerative diseases or other conditions (e.g., Parkinson's disease, Alzheimer's disease, Multiple Sclerosis or Rheumatoid Arthritis) typing in mobile keyboards during 9 months time span. An enormous challenge for exploiting this dataset is that the keystroke data captured was acquired passively, in a total transparent way for the participant, without any kind of supervision or labeled data. In the context of this work, this database was used to study whether our proposed system was able to detect trends in the fatigue levels during the daily typing habits of the healthy participant subset (a total of 251 healthy participants). Subjects gave informed consent prior to experiments, and experimental procedures were approved by the Committee On the Use of Humans as Experimental Subjects (COUHES) at the Massachusetts Institute of Technology, protocol no. 1504007090.

Data pre-processing and feature extraction. The raw data captured in all three keystroke databases are time series of three dimensions: press times, release times, and the keycode of each key. Owing to privacy concerns, the keycode was discarded and the keystroke features computed for each keystroke session are based only on the press and release key time events. These timestamps were in UTC format but with different time resolution depending on the acquisition protocol and device employed in each keystroke database. In order to normalize the keystroke data of the three databases all timestamps were converted to seconds, and also ensuring that all keystroke features computed later are close to 1. This normalization step is necessary to avoid saturation of the neurons in the recurrent layers of our system.

The keystroke features set is extracted at key level and is composed by: *i)* hold times (i.e., the elapsed time between press and release a key), *ii)* flight times (i.e., the elapsed time between two consecutive press events), *iii)* inter-key latency (i.e., the elapsed time between release a key and press the next key), and *iv)* inter-release latency (i.e., the elapsed time between two consecutive release events). According to this, the keystroke feature vector employed as input of our model $\mathbf{x}$ has a dimension of 150×4 (150 keystrokes by 4 features). If the keystroke sequence is lower than 150 keys we compute zero padding to fill with zeros up to reach that length, otherwise we truncate the keystroke sequence taking the first 150 keys.

The reason why we chosen this keystroke features is because we ensure to keep the same feature set as the one employed to evaluate TypeNet DNN model in previous works[19,22,36] with the Aalto Database. Remember that the TypeNet model is part of our fatigue detection system that we adapt for the fatigue detection task with transfer learning techniques, and therefore, the keystroke features set employed to feed the TypeNet model (i.e., the input of our fatigue detection model) must be the same.

TypeNet Architecture and Domain Adaptation. The TypeNet architecture proposed in[22] is composed by two Long-Short Term Memory (LSTM) layers of 128 neurons. LSTM layers are a special kind of RNN layers specifically designed to be sensitive to temporal changes in the input sequences, that we think could be well suited to detect relevant changes in the typing behaviour of the subject when they are fatigued. Additionally, each recurrent layer has a recurrent dropout of 0.2 and a dropout layer of 0.5 between them to avoid overfitting during training. The input of the TypeNet architecture has a masking layer to avoid the computation of error gradients for those zeros (i.e., zeros generated when zero padding is needed for keystroke sequences lower than 150 keys) and do not contribute to the loss function during training (more details of TypeNet architecture and evaluation in Acien *et al.*[22]. Finally, the output of the TypeNet architecture is an embedding feature vector $\mathbf{f}(\mathbf{x})$ of size 128×1.

In this work we transform this embedding feature vector $\mathbf{v}(\mathbf{x})$ (originally employed for keystroke user authentication at large scale) into a new embedding vector $\mathbf{g}(\mathbf{v}(\mathbf{x}))$ of the same size that is better suited for the fatigue detection task. To do this we employ domain adaptation techniques[24], in which the model learn a new task (i.e., the keystroke fatigue detection task) through the transfer of knowledge from a related task (i.e., keystroke user authentication) previously learnt by the model. In Fig. 2 an overview of the entire transfer learning process is depicted. The output of the TypeNet model is connected to the fatigue detection layer, that is composed by a Multi-Layer Perceptron (MLP) layer of 128 neurons with '*relu*' activation. During the training process, the keystroke feature vector $\mathbf{x}$ extracted from keystroke sessions of the Sleep Inertia database is employed to feed the TypeNet network, that is frozen during the entire training process so the weights of this network are not altered. Then, TypeNet computes the embedding features vector $\mathbf{v}(\mathbf{x})$ that are optimized for keystroke user authentication, thanks to the previous training with the Aalto database in Acien *et al.*[22]). Finally, the fatigue detection layer is fed with this embedding feature vector and learns to transform these embedding features into a new feature embedding vector $\mathbf{g}(\mathbf{v}(\mathbf{x}))$ optimized for the fatigue detection task, thanks to the labeled data of the nQSI Sleep Inertia database.

This kind of domain adaptation process is also referred to as fine-tuning, where part of the architecture is frozen (the TypeNet model in our case) that has the knowledge of typing patterns from thousands of users of the Aalto database, and therefore, we only need to train the last layer (the fatigue detection layer) to adapt these typing patterns for the fatigue detection task with the Sleep Inertia database. The main reason why we employ transfer learning with fine-tuning techniques is because to train a DNN model from the scratch for the fatigue detection task we will need thousands of subjects with labeled keystroke data to make the model robust, generalizable, and accurate. This technique allow us to overcome this issue, taking advantage of other DNN model previously trained with thousands of subjects for a similar task like TypeNet, and adapt it for the fatigue

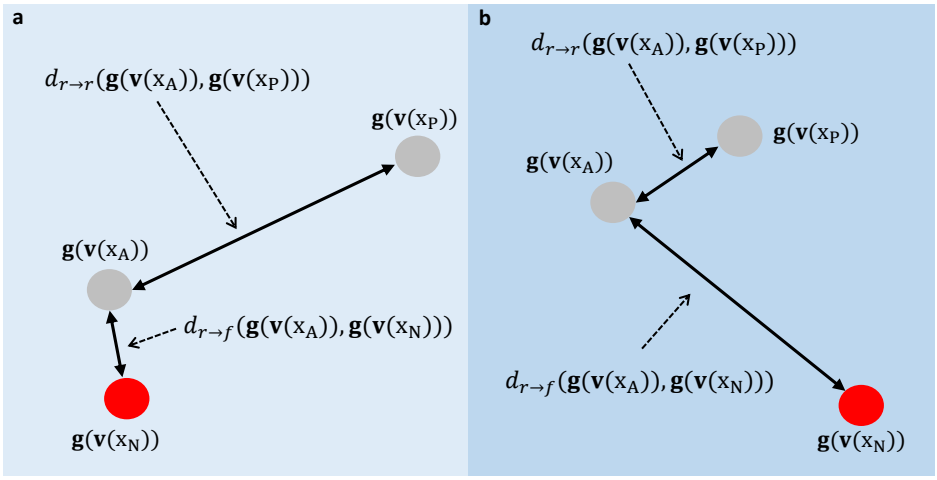

Figure 7. Example of how Triplet loss works. 2D representation of the embedding feature fatigue vectors $\mathbf{g}(\mathbf{v}(\mathbf{x}))$ before (**a**) and after (**b**) the triplet loss training. The embedding vectors that belong to the same class ($\mathbf{g}(\mathbf{v}(\mathbf{x}_\mathrm{P}))$ and $\mathbf{g}(\mathbf{v}(\mathbf{x}_\mathrm{A}))$) get closer meanwhile they get far from the embedding of the opposite class ($\mathbf{g}(\mathbf{v}(\mathbf{x}_\mathrm{P}))$ and $\mathbf{g}(\mathbf{v}(\mathbf{x}_\mathrm{N}))$).

detection task using only the 16 subjects of the Sleep Inertia database. Fine-tuning techniques have been broadly used in state-of-the-art works[37–39], where the databases employed are not large enough to train a DNN model from scratch.

Finally, to train the fatigue detection model successfully we employ the triplet loss function. This loss function is well suited for DML approaches where the output of the model to train is an embedding feature vector instead of a single score. A triplet is composed by three different samples from two different classes: Anchor (A) and Positive (P) are different keystroke sequences from the same class (Fatigue or Rest), and Negative (N) is a keystroke sequence from the other class. The Triplet loss function is defined as follows:

$$\mathscr{L}_{TL} = \max\{0, d(\mathbf{g}(\mathbf{v}(\mathbf{x}_\mathrm{A})),\mathbf{g}(\mathbf{v}(\mathbf{x}_\mathrm{P}))) - d(\mathbf{g}(\mathbf{v}(\mathbf{x}_\mathrm{A})),\mathbf{g}(\mathbf{v}(\mathbf{x}_\mathrm{N}))) + \alpha\} \tag{2}$$

where α is a margin between positive and negative pairs and d is the Euclidean distance calculated with Eq. 1. This learning process minimizes the distance between embedding vectors from the same class ($d(\mathbf{g}(\mathbf{v}(\mathbf{x}_\mathrm{A})),\mathbf{g}(\mathbf{v}(\mathbf{x}_\mathrm{P})))$), and maximizes it for embeddings from different classes ($d(\mathbf{g}(\mathbf{v}(\mathbf{x}_\mathrm{A})),\mathbf{g}(\mathbf{v}(\mathbf{x}_\mathrm{N})))$). Note that all three samples $\mathbf{x}_\mathrm{A},\mathbf{x}_\mathrm{P},\mathbf{x}_\mathrm{N}$ belong to the same subject in order to avoid intra-user variations as much as possible. An example of how the triplet loss function works is depicted in Fig. 7, where $\mathbf{g}(\mathbf{v}(\mathbf{x}_\mathrm{P}))$ and $\mathbf{g}(\mathbf{v}(\mathbf{x}_\mathrm{A}))$ are two feature embedding vectors (i.e., the output of the fatigue detection model when is fed with $\mathbf{x}_\mathrm{P}$ and $\mathbf{x}_\mathrm{A}$ samples respectively) that belong to the same class, while $\mathbf{g}(\mathbf{v}(\mathbf{x}_\mathrm{N}))$ belongs to the opposite class. During the training process (see Fig. 7.b), the triplet loss function will make $\mathbf{g}(\mathbf{v}(\mathbf{x}_\mathrm{P}))$ and $\mathbf{g}(\mathbf{v}(\mathbf{x}_\mathrm{A}))$ get closer at the same time they get far from $\mathbf{g}(\mathbf{v}(\mathbf{x}_\mathrm{N}))$. Remember that we only train the fatigue detection layer because the TypeNet network is frozen during training (fine-tuning), thereby this entire process is learnt by the fatigue detection layer. The unique purpose of this layer is to separate in the latent space the feature embedding vectors that belong to the rest state from those that belong to the fatigue state. Examples of the final results are showed in Fig. 3 applying dimensional reduction to the embedding feature vectors for 2D visualization.

Regarding experimental protocol details, we follow a Leave-One-Out (LOO) cross validation strategy by employing all subjects but one of the Sleep Inertia database to train the proposed system and testing with the remaining subject. This means that we have 16 different fatigue detection models (one for each test subject). The best results were achieved with a learning rate of 0.005, Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\varepsilon = 10^{-8}$. The models were trained for 30 epochs with 100 batches per epoch and 64 triplets in each batch.

QCD algorithm. The Active Fatigue Detection (AFD) algorithm is based on the QCD algorithm proposed in Perera *et al.*[26] for intrusion detection based on mobile behavior biometrics. In this work the algorithm is redesigned for the active fatigue detection task. The algorithm is based on calculating a new score from the cumulative sum of previous events (keystroke sessions). If the subject is in a rested state (grey lines in Fig. 5), the cumulative sum will be almost zero. At the moment the subject mental state changes into fatigue during typing, this score will tend to increase until reaching a certain threshold, in which we detect the fatigue symptoms. The cumulative sum is calculated as follow:

$$score_j^{AFD} = \max(score_{j-1}^{AFD} + L_j, 0) \tag{3}$$

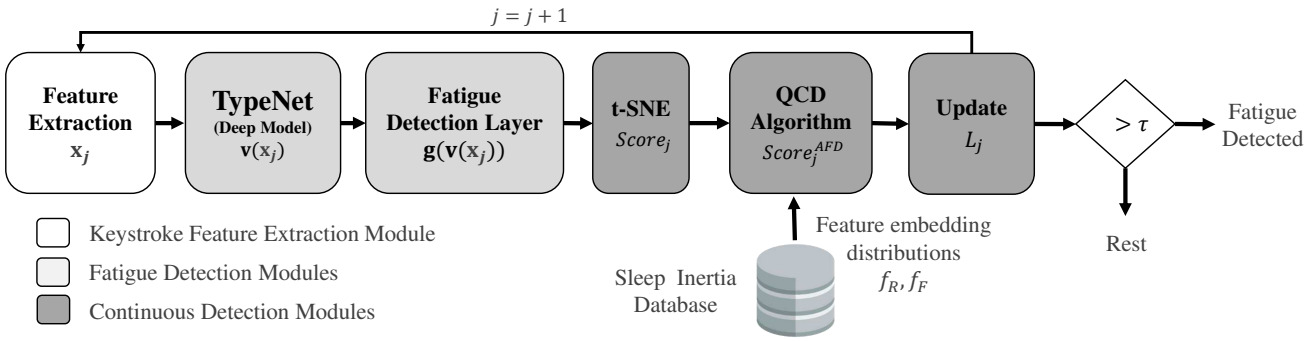

Figure 8. The entire pipeline of the Active Detection Algorithm. The $score_j$ is computed by performing t-SNE for dimensional reduction to the embedding fatigue vector $\mathbf{g}(\mathbf{v}(\mathbf{x}))$. Then, $score_j^{AFD}$ is obtained by comparing $\mathbf{g}(\mathbf{v}(\mathbf{x}))$ with the distributions obtained from the nQSI database. Finally, a threshold τ is used to detect the Fatigue states.

where j means the actual keystroke session and $score_{j-1}^{AFD}$ is the previous cumulative score. L_j is the contribution of the actual event calculated as the log-likelihood ratio between score distributions:

$$L_j = \log\left(\frac{f_F(score_j)}{f_R(score_j)}\right) \tag{4}$$

where f_R and f_F are the probability distributions of the rest and fatigue scores respectively, calculated previously with the fatigue detection model (as showed in Fig. 2) and $score_j$ is the fatigue score of the actual event. Note that the output of the fatigue detection model is an embedding feature vector $\mathbf{g}(\mathbf{f}(\mathbf{x}))$ of size 1×128, so we compute t-SNE for dimensional reduction to one dimension (i.e, we reduce the size of the embedding vector to one), in order to obtain a single fatigue score $score_j$. According to Eq. 4, the log-likelihood ratio L_j will be negative if $score_j$ belongs to rested keystroke session and positive in the opposite case, and therefore, multiple consecutive keystroke sessions of the fatigued subject will increase the cumulative sum $score_j^{AFD}$. Fig. 8 depicts an example of the entire AFD algorithm pipeline for a single subject. The fatigue detection model computes the embedding feature vector $\mathbf{g}(\mathbf{f}(\mathbf{x}))$ when is fed with a keystroke session. Then, we compute t-SNE for dimensional reduction to obtain the $score_j$. Finally, we upgrade $score_j^{AFD}$ by computing the L_j with the new score according to Eq. 3, which will increase up to reach the fatigue detection threshold in the case the subject is fatigued.

Data Availability

Anonymized data, not published in the article, will be shared on reasonable request from a qualified investigator.

References

1. Tanaka, M., Ishii, A. & Watanabe, Y. Effects of mental fatigue on brain activity and cognitive performance: a magnetoencephalography study. *Anat Physiol* **4**, 1–5 (2015).
2. Johansson, B., Starmark, A., Berglund, P., Rödholm, M. & Rönnbäck, L. A self-assessment questionnaire for mental fatigue and related symptoms after neurological disorders and injuries. *Brain Inj.* **24**, 2–12 (2010).
3. O'Keefe-McCarthy, S., McGillion, M. H., Victor, J. C., Jones, J. & McFetridge-Durdle, J. Prodromal symptoms associated with acute coronary syndrome acute symptom presentation. *Eur. J. Cardiovasc. Nurs.* **15**, e52–e59 (2015).
4. Goldman, J. G. & Postuma, R. Premotor and nonmotor features of Parkinson's disease. *Curr. Opin. Neurol.* **27** (2014).
5. Lou, J.-S., Kearns, G., Oken, B., Sexton, G. & Nutt, J. Exacerbated physical fatigue and mental fatigue in Parkinson's disease. *Mov. Disord. Off. J. Mov. Disord. Soc.* **16**, 190–196 (2001).
6. Kluger, B. M. *et al.* Parkinson's disease-related fatigue: a case definition and recommendations for clinical research. *Mov. Disord.* **31**, 625–631 (2016).
7. Friedman, J. H. *et al.* Fatigue in Parkinson's disease: a review. *Mov. Disord. Off. J. Mov. Disord. Soc.* **22**, 297–308 (2007).
8. Zesiewicz, T., Patel-Larson, A., Hauser, R. & Sullivan, K. Social Security Disability Insurance (SSDI) in Parkinson's disease. *Disabil. Rehabil.* **29**, 1934–1936 (2007).
9. De Vries, J., Michielsen, H. J. & Van Heck, G. L. Assessment of fatigue among working people: a comparison of six questionnaires. *Occup. Environ. Medicine* **60**, i10–i15 (2003).

10. Rahimian Aghdam, S., Alizadeh, S. S., Rasoulzadeh, Y. & Safaiyan, A. a. Fatigue assessment scales: A comprehensive literature review. *Arch. Hyg. Sci.* **8** (2019).

11. Michielsen, H. J., De Vries, J., Van Heck, G. L., Van de Vijver, F. J. & Sijtsma, K. Examination of the dimensionality of fatigue. *Eur. J. Psychol. Assess.* **20**, 39–48 (2004).

12. Kim, J. & Kang, P. Freely typed keystroke dynamics-based user authentication for mobile devices based on heterogeneous features. *Pattern Recognit.* **108**, 107556 (2020).

13. Morales, A. *et al.* Keystroke Biometrics Ongoing Competition. *IEEE Access* **4**, 7736–7746 (2016).

14. Banerjee, S. & Woodard, D. Biometric authentication and identification using keystroke dynamics: A survey. *J. Pattern Recognit. Res.* **7**, 116–139 (2012).

15. Acien, A., Morales, A., Vera-Rodriguez, R. & Fierrez, J. Keystroke mobile authentication: Performance of Long-Term approaches and fusion with behavioral profiling. In *Proc. of the Iberian Conf. on Pattern Recognition and Image Analysis (IBPRIA)*, vol. 11868 of *LNCS*, 12–24 (Springer, 2019).

16. Giancardo, L. *et al.* Computer keyboard interaction as an indicator of early Parkinson's disease. *Sci. Reports* **6**, 1–10 (2016).

17. Arroyo-Gallego, T. *et al.* Detecting motor impairment in early Parkinson's disease via natural typing interaction with keyboards: Validation of the neuroQWERTY approach in an uncontrolled at-home setting. *J. Med. Internet Res.* **20**, e9462 (2018).

18. Van Waes, L., Leijten, M., Mariën, P. & Engelborghs, S. Typing competencies in Alzheimer's disease: An exploration of copy tasks. *Comput. Hum. Behav.* **73**, 311–319 (2017).

19. Acien, A., Morales, A., Vera-Rodriguez, R., Fierrez, J. & Monaco, J. V. TypeNet: Scaling up keystroke biometrics. In *IEEE International Joint Conference on Biometrics (IJCB)*, 1–7 (2020).

20. Dhakal, V., Feit, A. M., Kristensson, P. O. & Oulasvirta, A. Observations on typing from 136 million keystrokes. In *Proc. of the ACM CHI Conference on Human Factors in Computing Systems* (2018).

21. Giancardo, L., Sánchez-Ferro, A., Butterworth, I., Mendoza, C. & Hooker, J. M. Psychomotor impairment detection via finger interactions with a computer keyboard during natural typing. *Sci. Reports* **5**, 1–8 (2015).

22. Acien, A., Morales, A., Monaco, J. V., Vera-Rodriguez, R. & Fierrez, J. TypeNet: Deep learning keystroke biometrics. *IEEE Transactions on Biom. Behav. Identity Sci.* **4**, 57–70 (2022).

23. Hadsell, R., Chopra, S. & Lecun, Y. Dimensionality reduction by learning an invariant mapping. In *Proc. Computer Vision and Pattern Recognition Conference* (2006).

24. Singh, R., Vatsa, M., Patel, V. M. & Ratha, N. *Domain Adaptation for Visual Understanding* (Springer, 2020).

25. Acien, A. *et al.* On the analysis of keystroke recognition performance based on proprietary passwords. In *Proc. of the 8th International Conference on Pattern Recognition Systems (ICPRS-17)*, 1–6 (2017).

26. Perera, P., Fierrez, J. & Patel, V. Quickest intruder detection for multiple user active authentication. In *IEEE Intl. Conf. on Image Processing (ICIP)* (2020).

27. Ulinskas, M., Damasevicius, R., Maskeliunas, R. & Wozniak, M. Recognition of human daytime fatigue using keystroke data. In Shakshuki, E. M. & Yasar, A. (eds.) *The 9th International Conference on Ambient Systems, Networks and Technologies (ANT 2018) / The 8th International Conference on Sustainable Energy Information Technology (SEIT 2018) / Affiliated Workshops, May 8-11, 2018, Porto, Portugal*, vol. 130 of *Procedia Computer Science*, 947–952 (Elsevier, 2018).

28. Marlon de Jong, I., Bonvanie, A. M. & Lorist, M. M. Dynamics in typewriting performance reflect mental fatigue during real-life office work. *PLOS ONE* **15**, 1–15 (2020).

29. Al-Libawy, H., Al-Ataby, A., Al-Nuaimy, W., Al-Taee, M. A. & Al-Jubouri, Q. Fatigue detection method based on smartphone text entry performance metrics. In *International Conference on Developments in eSystems Engineering (DeSE)*, 40–44 (2016).

30. Kryger, M., Roth, T. & Dement, W. C. In *Principles and Practice of Sleep Medicine* (Elsevier, 2017), sixth edition edn.

31. Monk, T. H. The post-lunch dip in performance. *Clin. Sports Medicine* **24**, e15—23, xi—xii (2005).

32. Krupp, L., LaRocca, N., Muir-Nash, J. & Steinberg, A. The fatigue severity scale. application to patients with Multiple Sclerosis and Systemic Lupus Erythematosus. *Arch. Neurol.* **46**, 1121—1123 (1989).

33. Morrow, G. R., Andrews, P. L., Hickok, J. T., Roscoe, J. A. & Matteson, S. Fatigue associated with cancer and its treatment. *Support. Care Cancer* **10**, 389–398 (2002).

34. Wostyn, P. COVID-19 and chronic fatigue syndrome: Is the worst yet to come? *Med. Hypotheses* **146**, 110469 (2021).

35. Carskadon, M. A., Dement, W. C. *et al.* Normal human sleep: An overview. *Princ. Pract. Sleep Medicine* **4**, 13–23 (2005).

36. Morales, A., Fierrez, J., Acien, A., Tolosana, R. & Serna, I. SetMargin Loss applied to deep keystroke biometrics with circle packing interpretation. *Pattern Recognit.* **122**, 108283 (2022).

37. Creagh, A. P., Lipsmeier, F., Lindemann, M. & De Vos, M. Interpretable deep learning for the remote characterisation of ambulation in multiple sclerosis using smartphones. *Sci. Reports* **11** (2021).

38. Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L. & Muller, P.-A. Transfer Learning for time series classification. In *Proc. of the IEEE international Conference on Big Data (Big Data)*, 1367–1376 (IEEE, 2018).

39. Phan, H. *et al.* Towards more accurate automatic sleep staging via deep transfer learning. *IEEE Transactions on Biomed. Eng.* **68**, 1787–1798 (2020).

Acknowledgements

This work is a collaboration between nQ Medical Inc. and BiDA-LAB group and has been supported by projects: TRESPASS-ETN (MSCA-ITN-2019-860813), PRIMA (MSCA-ITN-2019-860315), BBforTAI (PID2021-127641OB-I00 MICINN/FEDER), IDEA-FAST (H2020-IMI2-2018-15-853981), edBB (UAM), and Instituto de Ingenieria del Conocimiento (IIC). A. Acien is supported by a FPI fellowship from the Spanish MINECO. A. Morales is supported by "Programa de Excelencia del Profesorado Universitario" from CAM.

We would like to thank the neuroQWERTY team for their work in creating the nQCS and nQSI datasets. We would like to thank the nQ Medical's team members for their feedback in the preparation of this manuscript.

Author contributions statement

A.A., A.M., and T.A.G. conceived the experiment(s), A.A. and T.A.G. conducted the experiment(s), T.A.G., R.V. and J.F. analysed the results. All authors reviewed the manuscript.

Competing Interests

The Authors declare no Competing Non-Financial Interests but the following Competing Financial Interests: A.A., I. M.-C., T. A.-G. are employees at nQ Medical Inc. and received a regular salary while contributing to this work.