République Algérienne Démocratique et Populaire

Ministère de La Formation et de L'enseignement Professionnels

Ecole Supérieure en Sciences et Technologies de l'Informatique et du Numérique

(ESTIN) – Bejaïa

Rapport de Projet

Reconnaissance biométrique comportementale

Biométrie comportementale (frappe au clavier) en utilisant l'algorithme DTW

Encadré par : **Dr.** Benzenati Rahima

Réalisé par :

- **Mlle.** Hammou nour el houda
- **Mlle.** Mihoub Rahma

Année universitaire 2024-2025

Table des matières

Liste des figures

Liste des tableaux

Introduction générale

Dans un environnement de plus en plus digitalisé, la problématique de l'identité et de la sûreté se révèle cruciale. Que ce soit pour avoir accès à un service, sécuriser des informations délicates ou assurer la fiabilité d'un système, il est devenu indispensable de vérifier que l'individu avec qui l'on interagit est véritablement celui qu'il prétend être. Avec l'augmentation des menaces, les techniques traditionnelles d'authentification montrent leurs insuffisances .

C'est dans cette configuration que la biométrie, en tant que système d'identification basé sur des traits distinctifs à chaque individu, occupe une position de plus en plus dominante.Elle propose une solution innovante face aux défis de la sécurité et de l'authentification, en privilégiant non pas ce que nous détenons ou ce que nous savons, mais plutôt qui nous sommes et comment nous agissons.

Chapitre 1

La biométrie

La biométrie est une technologie d'identification et de vérification basée sur l'étude des traits distinctifs propres à chaque individu. Au lieu de faire confiance à des mots de passe ou des cartes, la biométrie s'appuie sur l'identité de l'utilisateur (physique ou biologique) ou comportemental ce qui donne aux systèmes une plus grande fiabilité et rend leur imitation plus complexe.

On distingue principalement trois grandes catégories de biométrie, chacune reposant sur des caractéristiques différentes :

Type de biométrie	**Définition**	**Exemples**
Physique (morphologique)	Analyse des caractéristiques corporelles stables et visibles	Empreintes digitales, forme du visage, iris, rétine, forme de la main
Comportementale	Analyse des habitudes et mouvements dynamiques propres à chaque individu	Frappe au clavier, dynamique de la souris, signature, voix, démarche
Biologique	Analyse d'éléments biologiques internes ou organiques	ADN, sang, salive, urine, odeur corporelle

TABLE 1.1 – Catégories principales de biométrie

Le fonctionnement d'un système biométrique repose sur deux étapes principales :

— **L'enrôlement** : consiste à enregistrer la représentation numérique des caractéristiques ou mesures biométriques dans une banque de données ou sur tout autre support à partir desquels d'autres données biométriques sont ensuite individuellement comparées.

— **La reconnaissance** : la banque de données saisie à l'étape d'enrôlement est comparée à une seconde donnée afin de créer une association et ainsi identifier ou authentifier une personne

Grâce à leurs propriétés uniques, difficiles à reproduire, les systèmes biométriques sont devenus des outils essentiels dans de nombreux domaines tels que la cybersécurité.

Chapitre 2

La biométrie comportementale

2.1 Définition et principes

La biometrie comportemental est une technologie d'identification et d'authentification qui focalise sur ce que vous faites au lieu de ce que vous etes ce qui veut dire que la biometrie comportemental identifie les comportement d'un individus sur un appareil Contrairement à la biométrie physique, qui s'appuie sur des traits anatomiques fixes .Ces comportements sont le reflet de schémas neurologiques, musculaires et cognitifs uniques, qui sont complexes à reproduire. La reconnaissance comportementale est habituellement continue et passive ce qui donne une confirmation permanente de l'identité de l'utilisateur sans perturber son activité en cours.

2.2 Avantages et limites de la biométrie comportementale

La biométrie comportementale présente plusieurs avantages significatifs. Elle permet une **authentification continue** sans intervention explicite de l'utilisateur, ce qui la rend moins intrusive. Elle est également **économique** car elle ne nécessite pas de grands matériel. Toutefois, certaines limitations existent. Les comportements peuvent varier selon le contexte : fatigue, stress, changement d'appareil ce qui peut peut affecter la précision de la reconnaissance.

Chapitre 3

L'analyse des frappes au clavier

3.1 Définition de la dynamique de frappe

La dynamique de frappe, aussi connue sous le nom de *k*eystroke *d*ynamics, est une méthode d'identification qui se base sur le mode d'utilisation du clavier par un individu. Elle évalue des indicateurs comme le temps d'appui sur une touche (dwell time), l'intervalle de temps entre deux frappes (flight time), ainsi que les enchaînements de touches pressées simultanément, tels que les digraphes (paires de lettres) ou les trigraphes (triplets de lettres). Ces informations sont recueillies de façon passive durant la saisie et sont spécifiques à chaque individu.

3.2 Origine de l'unicité

L'unicité de la manière de taper résulte d'un ensemble de facteurs neurophysiologiques et comportementaux. La façon dont on tape peut être influencée par plusieurs facteurs tels que la morphologie des mains et des doigts, la mémoire musculaire, la posture adoptée, la coordination entre les yeux et les mains, ainsi que les habitudes développées au fil des années. Cette complexité rend compliqué de reproduire la dynamique de frappe,meme en cas de possession du bon mot de passe. D'après diverses recherches, y compris celle de Monrose et Rubin (2000)[3], la dynamique de frappe peut constituer un indice comportemental sûr.

3.3 Paramètres mesurés

Les principaux paramètres utilisés pour l'analyse des frappes sont :

— **Dwell time** : le temps pendant lequel une touche reste enfoncée.

— **Flight time** : le temps entre la libération d'une touche et l'appui sur la suivante.

— **Latency** : le temps entre deux pressions (press-to-press) ou entre deux relâchements (release-to-release).

— **Digraphs et trigraphs** : combinaisons fréquentes de lettres analysées pour détecter des séquences comportementales caractéristiques.

Ces paramètres sont enregistrés sous forme de séries temporelles qui peuvent ensuite être comparées à un modèle d'utilisateur pour identifier ou authentifier celui-ci.

3.4 Pourquoi cette modalité est pertinente

L'analyse des frappes au clavier présente plusieurs avantages :

— **Pas de matériel supplémentaire** : tout ordinateur ou terminal disposant d'un clavier peut collecter les données nécessaires.

— **Utilisation continue** : elle peut s'intégrer à des systèmes d'authentification dynamique pour surveiller en temps réel l'identité de l'utilisateur.

— **Difficulté d'imitation** : contrairement aux mots de passe, le style de frappe est difficile à reproduire, même si l'attaquant connaît le texte à saisir.

— **Complémentarité** : elle peut renforcer d'autres formes d'authentification (2FA, biométrie physique, etc.).

En résumé, cette modalité repose sur des caractéristiques comportementales discrètes mais puissantes, ce qui en fait une solution prometteuse dans les contextes de sécurité numérique.

Chapitre 4

Algorithme Dynamic Time Warping (DTW)

Introduction

L'algorithme Dynamic Time Warping (DTW) est une technique d'alignement utilisée pour comparer des séquences qui peuvent différer dans le temps ou la vitesse. Ce chapitre explore ses fondements, son fonctionnement interne et ses applications.

4.1 Présentation de DTW

Le **Dynamic Time Warping (DTW)** est un algorithme conçu pour comparer deux séquences temporelles en les alignant de façon **non linéaire**. Contrairement à la **distance euclidienne**, qui compare les éléments correspondants (même index), DTW permet d'adapter dynamiquement l'alignement entre les séquences en étirant ou compressant localement l'axe temporel.

Cette flexibilité rend DTW bien plus robuste que la distance euclidienne lorsqu'il s'agit de comparer des séquences de longueurs différentes ou présentant des variations de vitesse. Par exemple, deux frappes de clavier similaires effectuées à des rythmes différents apparaîtront éloignées selon la distance euclidienne, alors que DTW reconnaîtra leur similarité en ajustant l'alignement.[2]

DTW est donc largement utilisé dans des domaines où la variation temporelle est fréquente, notamment :

- Reconnaissance vocale,
- Reconnaissance d'écriture manuscrite,
- Authentification par frappe clavier.

La figure suivante compare visuellement l'alignement produit par la distance euclidienne et celui produit par DTW. On remarque que DTW permet un meilleur couplage des points similaires en dépit de différences de tempo.

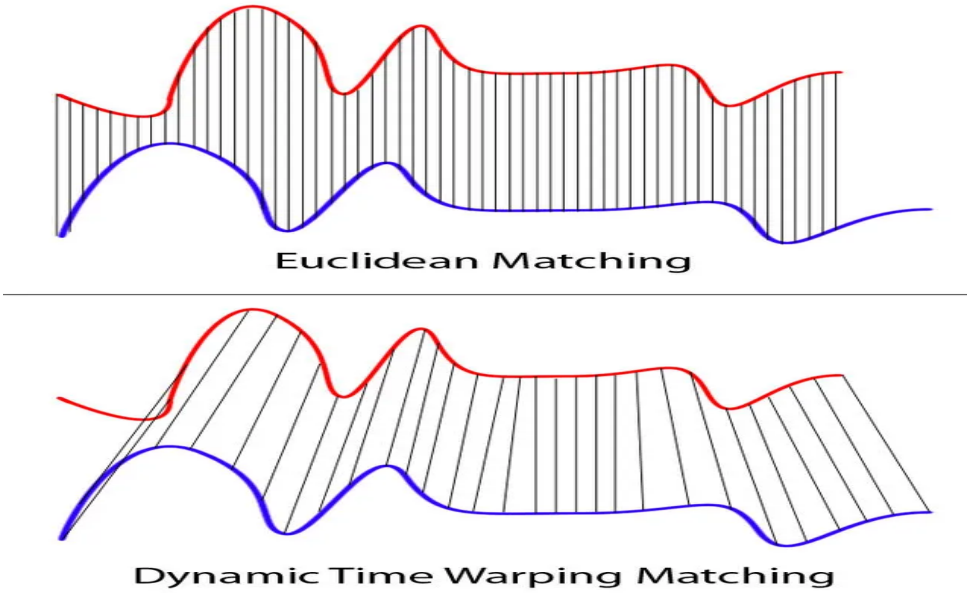

FIGURE 4.1 – Comparaison entre un alignement Euclidien et un alignement DTW.

[1]

L'objectif principal de DTW est de minimiser un coût global de distorsion en trouvant un alignement optimal entre les points de deux séquences.

4.2 Détails de l'algorithme DTW

L'algorithme DTW repose sur une matrice de coût où chaque cellule représente une distance entre deux éléments des séquences. Voici les étapes principales du processus, accompagnées de leurs illustrations :

1. **Construction de la matrice de distance locale** : Chaque cellule (i, j) contient la distance euclidienne entre x_i et y_j :

$$d(i, j) = \sqrt{(x_i - y_j)^2}$$

La figure suivante illustre cette étape fondamentale.

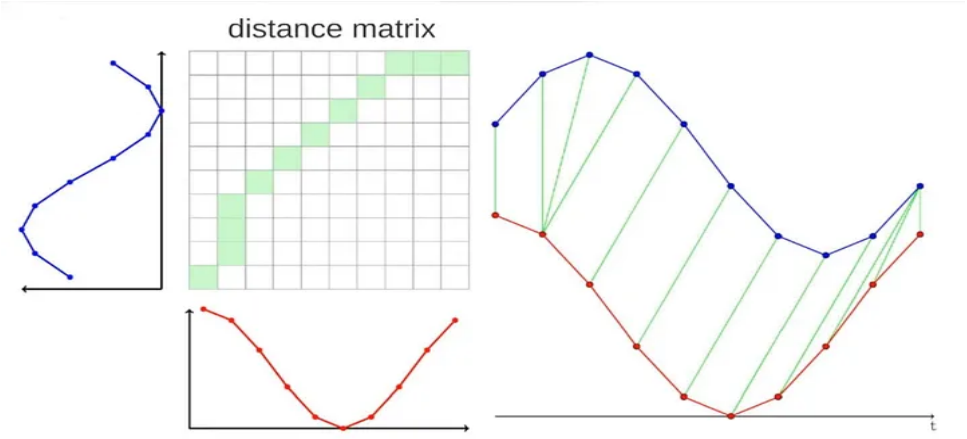

FIGURE 4.2 – Matrice de distance locale entre deux séquences temporelles.

[1]

2. **Initialisation de la matrice de coût cumulé** : Le point de départ $(0, 0)$ est initialisé avec $d(0, 0)$. Les premières lignes et colonnes sont remplies avec des coûts cumulés.
3. **Remplissage de la matrice** : Chaque cellule est ensuite remplie dynamiquement en choisissant le minimum entre les trois cellules précédentes (haut, gauche, diagonale) :

$$D(i, j) = d(i, j) + \min\{D(i - 1, j), D(i, j - 1), D(i - 1, j - 1)\}$$

4. **Extraction du chemin optimal** : Une fois la matrice remplie, on détermine le chemin d'alignement optimal à partir de la cellule $(N - 1, M - 1)$. Ce chemin représente la meilleure correspondance entre les séquences, même si elles sont désynchronisées.

3	33	23	19	16	19	23	18	17	18	15
7	31	20	18	16	19	17	17	18	15	18
5	25	19	13	12	16	15	14	15	14	16
1	21	18	10	11	11	19	14	13	17	18
2	21	13	9	10	12	16	11	12	16	17
8	20	9	13	16	19	9	12	17	18	21
9	13	7	11	11	14	8	13	18	16	21
4	5	4	5	5	8	12	12	13	15	16
3	2	3	4	4	7	13	14	14	17	17
1	0	5	6	8	9	17	20	22	27	29
	1	6	2	3	0	9	4	3	6	3

FIGURE 4.3 – Visualisation du chemin optimal dans la matrice de coût cumulé.

[1]

5. **Complexité** :
   - Complexité temporelle : $O(NM)$,

- Complexité spatiale : $O(NM)$.

Pour des séquences longues, une version approximative nommée *FastDTW* permet de réduire la complexité à $O(N)$ avec une perte minimale de précision.

Conclusion

DTW offre une flexibilité essentielle pour comparer des séquences décalées ou non uniformes. Sa robustesse en fait un pilier de la reconnaissance temporelle, notamment dans les systèmes d'authentification biométrique.

Chapitre 5

Implémentation

Introduction

Cette partie explique comment nous avons intégré l'algorithme DTW à un système d'authentification basé sur la dynamique de frappe. Chaque composant logiciel, de l'interface à la vérification, est présenté avec son code.

5.1 Interface graphique - Tkinter

L'interface utilisateur est conçue avec la bibliothèque Tkinter. Elle permet aux utilisateurs d'interagir facilement avec le système à travers des boutons d'enrôlement, de vérification, et de visualisation.

```
def setup_ui(self):
    ...
    self.enroll_btn = ttk.Button(
        button_frame,
        text="Enroll Typing Profile",
        command=self.enroll,
        style='TButton'
    )
    ...
    self.verify_btn = ttk.Button(
        button_frame,
        text="Verify Identity",
        command=self.verify,
        style='TButton'
    )
    ...
```

```
    self.plot_btn = ttk.Button(
        button_frame,
        text="Show DTW Analysis",
        command=self.show_plot,
        style='TButton'
    )
```

5.2 Capture des frappes clavier

La saisie clavier est enregistrée en capturant les événements de type “down” et “up” pour chaque touche. Ces événements permettent d’analyser le comportement de frappe.

```
def record_keystrokes(self, prompt="Type the password: "):
    ...
    while True:
        event = keyboard.read_event()
        if event.event_type in ("down", "up"):
            timestamp = time.time() - start
            events.append((event.name, event.event_type, timestamp))
        ...
```

5.2.1 Extraction des caractéristiques temporelles

Les données capturées sont transformées en caractéristiques : *hold time* (temps de maintien) et *flight time* (temps entre les frappes consécutives).

```
def extract_features(self, events):
    ...
    for key, event_type, timestamp in events:
        if event_type == "down":
            ...
        elif event_type == "up" and key in key_down_times:
            hold_times[key] = timestamp - key_down_times[key]
    ...
    return np.array(features).reshape(-1)
```

5.2.2 Comparaison des profils avec DTW

Chaque séquence test est comparée à une référence à l’aide de FastDTW. Cela permet de mesurer la similarité entre les dynamiques de frappe.

```
def compare_features(self, ref, test):
    ...
    distance, _ = fastdtw(ref.reshape(-1, 1), test.reshape(-1, 1), dist=
        euclidean)
    return distance
```

5.3 Enrôlement (enregistrement)

Lors de l'enrôlement, l'utilisateur saisit le mot de passe plusieurs fois. Les vecteurs obtenus sont moyennés pour constituer un profil unique.

```
def enroll(self):
    ...
    for i in range(3):
        events = self.record_keystrokes(f"Type sample {i+1}: {password}")
        features = self.extract_features(events)
        reference_vectors.append(features)
    ...
```

5.4 Vérification de l'identité

Cette phase compare une nouvelle tentative de frappe au profil enregistré. Une distance DTW inférieure à un seuil valide l'authentification.

```
def verify(self):
    ...
    distance = self.compare_features(average_reference, test_features)
    if distance < threshold:
        ...
```

5.5 Visualisation de l'analyse DTW

Les distances obtenues lors de l'enrôlement sont affichées sur un graphique, ce qui aide à évaluer la cohérence et la précision du profil.

```
def show_plot(self):
    ...
    ax.plot(distances, marker='o', color=BUTTON_COLOR, linewidth=2)
    ax.axhline(y=threshold, color='r', linestyle='--', label='Threshold')
```

```
    ...
```

Conclusion

Le système développé s'appuie sur la dynamique de frappe pour authentifier un utilisateur de manière fiable. L'intégration de DTW assure une comparaison robuste, tandis que l'interface facilite l'expérience utilisateur.

Conclusion générale

La biométrie comportementale, notamment la reconnaissance de la frappe au clavier, constitue une innovation majeure dans le domaine des technologies d'authentification. Elle présente un moyen d'identification plus subtil et moins intrusif que les systèmes classiques reposant sur des mots de passe ou des empreintes digitales. Dans cette application, l'algorithme Dynamic Time Warping (DTW) a fait preuve d'une grande efficacité en offrant une comparaison robuste et adaptable des séquences de frappe.

DTW, en alignant de manière non linéaire deux séquences temporelles, se révèle supérieur aux techniques traditionnelles telles que la distance euclidienne, car il prend en considération les variations naturelles du tempo et de la vélocité de frappe propres à chaque individu. L'algorithme DTW est spécialement conçu pour gérer des données de saisie au clavier, où les utilisateurs peuvent être tentés de taper à des vitesses variées tout en préservant des modèles de comportement uniques.

L'emploi du DTW dans le domaine de la biométrie comportementale ne se limite pas à améliorer l'exactitude des systèmes d'authentification, il offre également une sécurité accrue en introduisant un niveau d'identification comportementale singulier et complexe à reproduire. Ce genre de système, qui s'appuie sur des modèles comportementaux, peut donc servir d'alternative ou même de substitut aux méthodes traditionnelles dans des applications où la sécurité et la confidentialité sont essentielles.

En somme, malgré les atouts indiscutables de DTW, particulièrement en matière de flexibilité et de robustesse, des recherches ultérieures pourraient viser à améliorer l'efficacité de l'algorithme, notamment sa rapidité de calcul, afin de le rendre plus apte aux usages en direct.

Bibliographie

[1] Sarthak Bhan. « What is Dynamic Time Warping ? » In : *Medium* (2023). Consulté en avril 2025. url : https://medium.com/@sarthakbhan/what-is-dynamic-time-warping-253a6880ad12.

[2] Jeff Heaton. *Dynamic Time Warping (DTW)*. https://youtu.be/_K1OsqCicBY. Consulté en avril 2025. 2014.

[3] Fabian Monrose et Aviel Rubin. « Keystroke dynamics as a biometric for authentication ». In : *Future Generation Computer Systems* 16.4 (2000), p. 351-359.