Biometric Authentication Based on Digital Signature Dynamics: An Empirical Analysis of Pen Stroke Patterns, Pressure, and Speed for Secure Cross-device Identification

Thompson Rivers University
Faculty of Science

Roods Bensly Pierre

February 2025

OverLeaf Link: `https://www.overleaf.com/6569282924bdjzjpswqgpb#598ac3`

Abstract

1 Introduction

In today's digital era, ensuring robust and reliable authentication methods is essential, yet traditional handwritten signatures continue to be widely used despite inherent vulnerabilities [1]. Consider a common scenario in schools: students are often required to submit parental signatures on permission slips for school trips. Given the ease with which these static signatures can be forged, it is not uncommon for students to replicate their parents' signatures, thereby compromising the intended security and trust in the system.

Digital signature biometrics address these challenges by capturing dynamic features such as pen pressure, stroke order, and speed—that are far more difficult to replicate [1]. This dynamic approach not only provides a stronger defense against forgery but also offers practical advantages in various real-world applications [2]. For example, banks can integrate digital signature biometrics into their transaction authentication processes, adding an extra layer of security that helps prevent fraudulent activities. Similarly, as more services migrate online, secure digital authentication methods become increasingly critical to protecting sensitive data and ensuring that only legitimate users gain access [1, 3].

In addition to enhanced security and user convenience, digital signature biometrics offer the unique advantage of being cancellable [4]. Unlike static biometric identifiers such as fingerprints or iris scans, which are permanent and immutable, digital signature biometrics can be revoked and replaced if compromised. This cancellability ensures that, in the event of a security breach, users can update their biometric profiles, thereby maintaining the integrity of the authentication system over time [5].

Despite its promise, several research questions remain open. In particular, it is unclear how consistent digital signature patterns remain when captured across different devices. Variations in hardware, such as digital tablets with varying sensitivity or sampling rates, could affect the reliability of the captured dynamics [6].

This paper aims to investigate the feasibility and reliability of biometric authentication via dynamic signature verification. By focusing on a systematic approach to data collection, feature extraction, machine learning model selection, and evaluation metrics, this work attempts to address ongoing gaps in cross-device adaptability and skilled forgery resistance. Ultimately, these findings could lay the groundwork for safer and more user-friendly authentication practices, applicable to diverse domains from finance to education.

2 Literature Review

A broad consensus in the research community acknowledges that biometric authentication enhances security by leveraging physiological or behavioral features that are difficult to replicate, thereby providing stronger safeguards compared to passwords or PINs [1]. This shift toward biometrics is driven by widespread vulnerabilities in conventional authentication systems, including social engineering and credential theft.

This section provides a review of the foundational and contemporary work in dynamic signature authentication, drawing on studies of fundamental biometric concepts, forgery detection, dynamic feature extraction, system robustness on mobile devices, and machine learning–based classification.

2.1 Signature as a Biometric Identifier

From a legal and practical standpoint, the signature has historically served as a cornerstone for document authentication and commercial transactions [4]. Traditional signature verification often hinges on comparing the static shape of a signature, leaving it vulnerable to skilled forgers who can carefully replicate shapes [2].

The reliability of digital signatures depends on their dynamic characteristics, such as stroke speed, pressure, and pen angle. Studies have demonstrated that integrating these dynamic traits into authentication systems significantly improves their ability to differentiate genuine users from impostors [3].However, these authors also emphasize that signatures are inherently behavioral, meaning they rely heavily on a person's neuromuscular coordination and can exhibit fluctuations under stress, fatigue, or other influences.

2.2 Forgery Challenges and Cancellable Biometrics

Forgery remains a critical challenge in traditional signature verification. Pal et al. categorizes forgeries into three types: random, simple, and skilled, where each level requires a correspondingly robust detection strategy [4]. Skilled forgeries, in which an impostor has access to a sample signature, pose the greatest risk. Even with dynamic features, well-trained forgers can imitate various aspects of signing behavior.

For this reason, a major area of research focuses on optimizing the trade-off between strict detection thresholds and practical usability. Systems that are overly sensitive may incorrectly reject legitimate users, whereas lenient systems run the risk of false acceptances.

Since signature biometrics can be cancellable, they hold an advantage over certain physiological biometrics in contexts where compromised credentials must be swiftly "reset"[4]. This adaptability is especially relevant in large-scale authentication scenarios where user credentials can be compromised by phishing attacks or data breaches.

2.3 Behavioral Biometrics and Fraud Detection

Behavioral biometrics, including signature verification, offer a novel approach to detecting fraudulent activity [5]. MasterCard demonstrates how behavior-based profiling, combining insights on typing speed, touchscreen interactions, and device-specific cues can help flag anomalies in real-time [5]. These insights can be integrated into fraud detection mechanisms, alerting users or blocking transactions when behavioral deviations are detected [5].

Emerging trends indicate that behavioral biometrics combined with artificial intelligence (AI) can improve fraud detection rates. However, these AI-driven approaches raise concerns regarding privacy and security, as continuous monitoring of user behavior can lead to ethical dilemmas in biometric surveillance.

2.4 Machine Learning in Signature Authentication

Machine learning has proven effective in digital signature verification. Leghari et al. compare classifiers like Support Vector Machines (SVM), Random Forest (RF), and Convolutional Neural Networks (CNNs), showing that deep learning models outperform traditional methods [7].

These studies demonstrate the feasibility of real-time authentication, with CNN models achieving high accuracy in distinguishing genuine and forged signatures [3]. However, these advanced methods are computationally intensive and may require specialized hardware for real-time use.

Despite the growth of machine learning solutions, cross-device consistency remains a challenge. Martinez-Diaz et al. note that signatures collected on mobile devices often suffer from degraded performance due to smaller input areas, inconsistent screen sensitivity, and varying ergonomics [6]. Nevertheless, mobile-based verification is promising for mass adoption, driving the need for algorithms that can normalize data across different devices.

2.5 Gaps and Trends in Research

Collectively, the existing literature paints a picture of a rapidly advancing field with several open questions:

1. **Cross-Device Consistency:** While dynamic signatures are robust, the ability to maintain accuracy across devices is insufficiently explored [6].

2. **Privacy Concerns:** Continuous user behavior monitoring raises ethical issues about privacy and consent, requiring transparent data policies [5].

3. **Forgery Resistance:** Deep learning methods enhance authentication but still lack robust models to counter highly skilled forgeries without compromising usability [4].

Across these studies, feature engineering emerges as a critical success factor. Whether using conventional statistical descriptors or deep learning–extracted features, capturing the temporal and spatial dynamics of handwriting remains the linchpin of accurate signature verification.

3 Model Building and Evaluation

3.1 Methodology

A machine learning model will be developed using Python and TensorFlow. The proposed methodology involves:

1. **Data Collection**: Signature samples will be obtained from digital tablets and mobile devices to analyze intra-user variations and cross-device consistency. Each participant will be required to sign multiple times to account for natural variations in handwriting.

2. **Preprocessing**: Raw signature data will be cleaned, normalized, and augmented to improve model generalization.

3. **Feature Extraction**: Dynamic features such as stroke velocity, acceleration, pressure, and stroke order will be extracted from the signatures.

4. **Model Training**: Various machine learning algorithms will be tested, including CNNs, SVM, and Random Forest.

5. **Evaluation**: Models will be assessed using key performance metrics such as accuracy, Equal Error Rate (EER), False Acceptance Rate (FAR), and False Rejection Rate (FRR).

3.2 Initial Code Implementation

```python
"""
Proposed Python code for training signature verification model using CNN based on lierature.
---------------------------------------------------------------
Features:
  - stroke_coords: NxTx2 (N samples, T time steps, x-y coords)
  - pressure: NxT
  - speed: NxT
Optional:
  - signature_images: NxH x W ( will soon decide if using images)
Labels: Nx1, where each entry is 0 (forged) or 1 (genuine)
"""

import numpy as np
import tensorflow as tf
from tensorflow.keras import layers
from sklearn.model_selection import train_test_split

# 1. Loading  dataset (assuming they will be npy)
# ------------------------------------------------
# stroke_coords.shape = (num_samples, time_steps, 2)
# pressure.shape = (num_samples, time_steps)
# speed.shape = (num_samples, time_steps)
# labels.shape = (num_samples,)
#

data = np.load("full_dataset.npy", allow_pickle=True).item()
stroke_coords = data["stroke_coords"]
pressure      = data["pressure"]
speed         = data["speed"]
labels        = data["labels"]

# If using image data as discussed :
# signature_images = data["signature_images"]

# 2. Combining features for 1D CNN input
# ----------------------------------------------------------
# Use x, y, pressure, and speed as separate channels.
# This yields a shape of (N, T, 4) -> T time steps, 4 channels.

num_samples = stroke_coords.shape[0]
time_steps  = stroke_coords.shape[1]  # T

# Expand pressure and speed to match dimensionality: (N, T, 1)
pressure_expanded = pressure[..., np.newaxis]
speed_expanded    = speed[..., np.newaxis]

# Concatenate along the channels axis = 2
# stroke_coords has shape (N, T, 2)
# combined_features => (N, T, 4)
combined_features = np.concatenate(
    [stroke_coords, pressure_expanded, speed_expanded], axis=2
)

# Split into training and test sets
X_train, X_test, y_train, y_test = train_test_split(
    combined_features, labels, test_size=0.2, random_state=42
)

# 3. Define a 1D CNN Model
# ------------------------------------------------------------
# I will probably use Conv1D layers to process the time dimension, while each feature is treated as a separate channel.

model = Sequential([
    # 1D convolutional layer with 32 filters, kernel size=3
    layers.Conv1D(filters=32, kernel_size=3, activation='relu',
    ↪  input_shape=(time_steps, 4)),
    layers.MaxPooling1D(pool_size=2),
    layers.Conv1D(filters=64, kernel_size=3, activation='relu'),
    layers.GlobalAveragePooling1D(),
    layers.Dropout(0.3),
    layers.Dense(128, activation='relu'),
    layers.Dropout(0.3),
    layers.Dense(1, activation='sigmoid')
])

# 4. Compile the model
model.compile(
    optimizer='adam',
    loss='binary_crossentropy',
    metrics=['accuracy']
)

model.summary()

# 5. Train the CNN
epochs = 10
model.fit(X_train, y_train, epochs=epochs,
↪  validation_data=(X_test, y_test))

# 6. Evaluate performance
loss, accuracy = model.evaluate(X_test, y_test)
print(f"Test Loss: {loss:.4f}")
print(f"Test Accuracy: {accuracy:.4f}")
```

References

[1] A. Jain, A. Ross, and S. Pankanti, "Biometrics: a tool for information security. ieee tran inform forensics secur," *Information Forensics and Security, IEEE Transactions on*, vol. 1, pp. 125 – 143, 07 2006.

[2] E. N. Ekwonwune, D. A. Ekekwe, C. I. Ubochi, and H. C. Oleribe, "Dynamic signature verification using pattern recognition," *Journal of Software Engineering and Applications*, vol. 17, no. 5, pp. 214–227, 2024.

[3] K. Roszczewska and E. Niewiadomska-Szynkiewicz, "Online signature biometrics for mobile devices," *Sensors*, vol. 24, no. 11, p. 3524, 2024.

[4] S. Pal, U. Pal, and M. Blumenstein, *Signature-Based Biometric Authentication*, pp. 285–314. Cham: Springer International Publishing, 2014.

[5] Mastercard, "Behavioral biometrics explained," *Mastercard News*, 2021.

[6] M. Martinez-Diaz, J. Fierrez, R. P. Krish, and J. Galbally, "Mobile signature verification: Feature robustness and performance comparison," *IET Biometrics*, vol. 3, no. 4, pp. 267–277, 2014.

[7] M. Leghari, A. A. Chandio, M. A. Soomro, S. Z. Nizamani, and M. H. Soomro, "A comparative analysis of machine learning algorithms for online signature recognition," *VFAST Transactions on Software Engineering*, vol. 12, no. 2, p. 231–240, 2024.