IN
PARTNERSHIP
WITH
PLYMOUTH
UNIVERSITY

Name: **Ashen Pulle**
Student Reference Number: **10899670**

Module Code: **PUSL3123**	Module Name: **AI and Machine Learning**
Coursework Title: **AI and Machine Learning - Referral Coursework**	
Deadline Date: **26th Aug 2025**	Member of staff responsible for coursework: **Dr. Neamah Al-Naffakh**
Programme: **BSc (Hons) Software Engineering**	

Please note that University Academic Regulations are available under Rules and Regulations on the University website www.plymouth.ac.uk/studenthandbook.

Group work: please list all names of all participants formally associated with this work and state whether the work was undertaken alone or as part of a team. Please note you may be required to identify individual responsibility for component parts.

We confirm that we have read and understood the Plymouth University regulations relating to Assessment Offences and that we are aware of the possible penalties for any breach of these regulations. We confirm that this is the independent work of the group.

Signed on behalf of the group:

Individual assignment: ***I confirm that I have read and understood the Plymouth University regulations relating to Assessment Offences and that I am aware of the possible penalties for any breach of these regulations. I confirm that this is my own independent work.***

Signed:

Use of translation software: failure to declare that translation software or a similar writing aid has been used will be treated as an assessment offence.

I *have used/not used translation software.

If used, please state name of software……………………………………………………………………

Overall mark _____% **Assessors Initials ______** **Date- 10th Aug 2025**

Table of Contents

Introduction

Smartphones, smartwatches, and other connected devices are now part of daily life, used for online banking, digital payments, and private communication. As these devices handle sensitive data, ensuring only the authorised person can access them is crucial. Traditional security methods such as passwords, PIN codes, and pattern locks have well-known weaknesses - they can be forgotten, guessed, stolen, or shared. They also interrupt the user, requiring manual input each time access is needed.

Biometric authentication offers a stronger, more user-friendly option. Physiological methods, such as fingerprint or facial recognition, are common but typically applied only at login and can sometimes be bypassed. Behavioural biometrics work differently by verifying a user based on how they naturally interact with a device. This enables continuous authentication in the background without repeated manual checks.

A promising behavioural method is acceleration-based authentication, which uses motion data from sensors like accelerometers to recognise unique movement patterns. These patterns are hard to copy and can be measured during normal device use, offering a secure and non-intrusive way to verify identity.

This project uses acceleration-based features to train a Feedforward Multi-Layer Perceptron (FFMLP) neural network to identify the authorised user and reject impostors. Three feature sets are compared: Time Domain, Frequency Domain, and Combined Time–Frequency Domain (TDFD) to determine the most effective.

The aims are:

- Feature Analysis – Identify features that are stable for the same user (low intra-user variance) and distinctive between users (high inter-user variance).
- Model Evaluation – Test the FFMLP across all domains using cross-day data to measure accuracy, False Acceptance Rate (FAR), False Rejection Rate (FRR), and Equal Error Rate (EER).
- Optimisation – Improve the best-performing domain through network design changes, feature scaling, and threshold tuning.

By combining acceleration data with a neural network, this study aims to show that behavioural biometrics can be a secure, reliable, and convenient way to protect devices and data.

Literature Review

The need for secure and convenient authentication methods has increased with the widespread use of smartphones, smartwatches, and other connected devices for activities such as online banking, e-commerce, and communication. Traditional methods like passwords and PINs are criticised for weaknesses in both security and usability. Biometric authentication, particularly behavioural biometrics, offers a more seamless alternative by verifying identity through unique behavioural patterns during normal device use (Siddiqui et al., 2022).

Acceleration-based authentication is a form of behavioural biometrics that uses motion data from sensors such as accelerometers to distinguish between users. This review examines recent work in this area, focusing on feature extraction, classification techniques, performance, and limitations.

Acceleration-Based Authentication Using Wearable Devices

Al-Naffakh et al. (2020) developed a continuous authentication system using smartwatch accelerometer data collected over multiple days. Features were extracted from the time domain (e.g., mean, variance) and frequency domain (e.g., spectral energy). Classifiers including Neural Networks, k-Nearest Neighbours (k-NN), and Support Vector Machines (SVM) were tested, with Neural Networks achieving the best balance between accuracy and error rates. However, results varied between individuals, highlighting the challenge of intra-user variability.

Combining Time and Frequency Features

Dos Santos et al. (2022) compared behavioural and physiological biometrics, finding that combining time and frequency domain features improved accuracy by up to 6% compared to single-domain models. This approach captures both short- and long-term movement characteristics, though high feature counts can increase overfitting risk with limited data.

Deep Learning for Behavioural Biometrics

Siddiqui et al. (2022) applied deep learning to behavioural biometrics using Artificial Neural Networks (ANNs) and 1D Convolutional Neural Networks (CNNs). ANN-based models achieved over 95% accuracy and handled noisy inputs well, but the dataset was collected in controlled conditions, which may not reflect real-world usage.

Classifier Choice in Mobile Authentication

Pelto et al. (2023) investigated touch dynamics on smartphones, testing Neural Networks, XGBoost, and SVM. Neural Networks performed best, exceeding 97% accuracy. The study also highlighted adaptive learning, where models are retrained to handle changes in user behaviour. While our project did not use adaptive retraining, it applies an FFMLP trained on first-day (FDay) data and tested on multi-day (MDay) data to assess stability across sessions.

Feature Selection and Stability

Kang et al. (2019) used wavelet-based multi-resolution analysis to select features with low intra-user variance and high inter-user variance, improving classification stability. Although the dataset was small, the method demonstrates the value of identifying stable and distinctive features, a concept applied in our optimisation stage.

Gap Analysis

From these studies, three trends stand out:

- Combining feature domains improves performance by capturing multiple aspects of user behaviour.
- Neural Networks outperform traditional classifiers when tuned to behavioural data.
- Stable features are essential for consistent performance across sessions.

However, many studies rely on controlled environments, limiting real-world applicability. Structured optimisation is underexplored, and while combined domains can improve results, they add complexity and risk of overfitting.

This project addresses these gaps by:

- Using cross-day datasets to evaluate stability in realistic conditions.
- Comparing time, frequency, and combined domains under identical setups.
- Applying targeted optimisation through architecture adjustments, feature scaling, and threshold tuning to improve accuracy and reduce errors.

Testing Methodology

This project develops a user authentication system using motion data from accelerometers, aiming to distinguish between the authorised user (target) and impostors. The problem is approached as a binary classification task, where the model outputs 1 for the target user and 0 for impostors. The system is evaluated using three types of features: Time Domain, Frequency Domain, and Combined Time–Frequency Domain (TDFD).

Data Analysis

Before model training, the dataset was analysed to understand feature behaviour through intra-user and inter-user variance:

- Intra-user variance measures how consistent a user's motion patterns are across sessions - specifically comparing first-day (FDay) and multi-day (MDay) data. Features with low intra-user variance are considered stable and more reliable for authentication.
- Inter-user variance measures how different each user's features are compared to others. Features with high inter-user variance are better for distinguishing between individuals.

This analysis ensured that selected features could support accurate classification, even with natural variations in user behaviour over time.

Data Preparation

The dataset contains accelerometer readings from 10 users. For each user and feature domain, two datasets were prepared:

- FDay – used for training.
- MDay – used for testing.

For each target user:

- Training set: 36 samples from the target user's FDay data (labelled 1) and 20 samples from each of the other 9 users (180 in total, labelled 0).
- Testing set: Prepared in the same way, using MDay data.

This approach ensures that the model is trained and tested on different sessions to evaluate performance under cross-day conditions.

Model Training

A Feedforward Multi-Layer Perceptron (FFMLP) neural network was implemented using MATLAB’s patternnet() function due to its ability to model complex, non-linear relationships.

Initial configuration (before optimisation):

- Hidden layers: Single hidden layer with 10 neurons.
- Activation function: Sigmoid (logsig).
- Training algorithm: Levenberg–Marquardt (trainlm).
- Epochs: 100 iterations.
- Data division: All data used for training; no internal validation/testing split.
- Training strategy: Each user and domain trained independently.

Testing and Evaluation

After training, the model outputs prediction scores between 0 and 1. A fixed threshold of 0.5 was applied:

- Scores ≥ 0.5 - classified as target user.
- Scores < 0.5 - classified as impostor.

Performance metrics:

- Accuracy – proportion of correctly classified samples.
- False Acceptance Rate (FAR) – proportion of impostor samples incorrectly accepted.
- False Rejection Rate (FRR) – proportion of genuine samples incorrectly rejected.
- Equal Error Rate (EER) – rate at which FAR and FRR are equal, used to assess the balance between security and usability.

The above process was repeated for all 10 users across the Time, Frequency, and TDFD domains, both before and after optimisation, to determine the most effective configuration.

Evaluation

Data Analysis

Intra-user variance measures the stability of a user's motion features across different days - in this case, between FDay (training data) and MDay (testing data). Low intra-user variance indicates that a user's behaviour is consistent, making it easier for the model to recognise them correctly in future sessions. High intra-user variance suggests that their movement patterns change more across days, which can lead to a higher risk of false rejections.

User	Feature Number		
	Frequency Domain	Time Domain	Time-Frequency Domain
User 01	2,10,18,20,26,32	62,63,64,72,73,84, 85,86	105,106,107,115,116,127,128,129
User 02	4,10,18,20,28,32	62,63,64,74,75,76,81,82,83,84,85,86 ,88	103,105,106,107,117,118,119,124,125,126,127,128
User 03	2,4,10,14,18,28,32,34,40	61,62,63,64,65,74,75,84,85,86	103,104,105,106,107,117,118,119,127,128,129 ,130
User 04	14,18,20,26,28,32	62,63,64,65,71,72,73,83,84,85,86	105,106,107,108,114,115,116,126,127,128,129
User 05	2,4,10,13,32,34,40	57,60,62,63,64,65,72,74,84,85,86,87,88	103,105,106,107,108,115,116,117,127,128,129,130,131
User 06	2,18,32,34,40	54,62,64,84,85,86	105,107,116,127,128,129
User 07	No Variance	No Variance	No Variance
User 08	2,14,18,28,32	61,62,63,64,65,72,73,82,83,84,85	104,105,106,107,108,115,116,125,126,127,128
User 09	2,14,18,26,28,34,40	62,63,64,66,71,72,73,74,83,84,85	97,105,106,107,109,114,115,116,117,126,127,128
User 10	2,10,13,18,28	52,60,61,63,64,65,70,71,72,73,74,75,84,85,86,87	95,103,104,106,107,108,113,114,115,116,117,118,127,128,129

Table 1. Intra Variance

From Table 1, most users had at least some features showing variance between sessions. Interestingly, User 07 displayed no measurable intra-user variance in any domain. While this might appear ideal, in practice it may mean the model lacks enough diverse information to differentiate between genuine and impostor cases - a factor later reflected in their performance results.

Inter-user variance measures how different one user's feature values are from those of others. A higher inter-user variance means the features are more distinctive, making it easier for the model to separate the target user from impostors.

Domain	Feature Number
Frequency Domain	2,4,10,13,14,18,20,26,28,32,34,40
Time Domain	52,54,56,57,59,60,61,62,63,64,65,66,70,71,72,73,74,75,76,81,82,83,84,85,86,87,88
Time-Frequency Domain	94,95,103,104,105,106,107,108,109,113,114,115,116,117,118,119,124,125,126,127,128,129,130,131,

Table 2. Inter Variance

As shown in Table 2, the frequency domain includes distinctive spectral features such as 2, 4, 10, 13, 14, 18, 20, 26, 28, 32, 34, and 40. These have likely contributed to strong performance for high-accuracy users like User 03 and User 10, who consistently ranked among the top in this domain. The time domain offers a much broader set of discriminative features (52–88), providing richer information for separation between users - a factor reflected in its generally higher accuracy range. The combined time–frequency domain (TDFD) incorporates distinctive elements from both sets and adds additional features (94, 95, 103-109, 113-119, 124-131) that capture both temporal and spectral characteristics. This fusion helps explain why TDFD delivered the most balanced results overall, as it leverages features that are both distinctive between users and stable within the same user across days.

Results Before Optimisation

The performance of the FFMLP classifier was evaluated across three domains: Time domain (88 features), Frequency domain (43 features), and the Combined Time-Frequency domain (TDFD – 131 features). For each user, models were trained on FDay data and tested on MDay data to assess cross-session performance. Evaluation metrics include Accuracy, False Acceptance Rate (FAR), False Rejection Rate (FRR), and Equal Error Rate (EER). The following sections summarize the results obtained for each domain before optimization.

Frequency Domain

Users	Accuracy	FAR	FRR	EER
User 01	93.06%-95.37%	3.89%-4.44%	8.33%-19.44%	6.11%-11.94%
User 02	94.91%-99.07%	1.11%-6.11%	0.00%	0.56%-3.06%
User 03	99.07%-99.54%	0.00%-0.56%	2.78%-5.56%	1.39%-2.78%
User 04	87.96%-89.81%	12.22%-13.33%	0.00%-5.56%	6.67%-9.44%
User 05	98.15%-99.07%	1.11%-2.22%	0.00%-2.78%	0.56%-1.11%
User 06	93.52%-96.30%	0.56%-7.22%	2.78%-19.44%	5.00%-10.00%
User 07	78.70%-85.19%	17.78%-26.11%	0.00%	8.89%-13.06%
User 08	93.98%-94.44%	1.11%-2.22%	16.67%25.00%	9.44%-12.22%
User 09	89.81%-95.37%	0.00%-2.78%	11.11%-19.44%	5.56%-13.89%
User 10	98.15%-99.07%	1.11%-2.22%	0.00%-2.78%	0.56%-2.22%
Overall	**92.27%-94.84%**	**4.89%-7.72%**	**6.94%-8.33%**	**5.78%-7.75%**

Table 3 Frequency Domain Before Optimising

In the frequency domain, average accuracy ranged between **92.27% and 94.84%** across users, indicating that spectral features can provide solid performance in many cases. However, there were significant differences between individuals:

- High performers – Users 03, 05, and 10 consistently exceeded 98% accuracy with low EER values (0.56%–2.78%). These users also had several features in Table 2 with strong inter-user variance, allowing the model to reliably distinguish them from impostors.
- Low performers – User 07 achieved the lowest accuracy (78.70%–85.19%), with the highest FAR (up to 26.11%). Despite having no intra-user variance, their feature set may lack discriminatory power against impostors in the frequency domain. User 08 also showed elevated FRR (up to 25.00%), likely due to multiple high-variance time-domain features that are not well captured in the frequency domain alone.

The EER values were generally low for most users, but the gap between top and bottom performers shows that frequency-only features may not generalise well to all behaviour patterns, especially when impostor similarities exist.

Time Domain

Users	Accuracy	FAR	FRR	EER
User 01	87.96%-98.61%	1.67%-14.44%	0.00%	0.83%-7.22%
User 02	91.20%-99.54%	0.56%-10.56%	0.00%-2.78%	0.28%-5.28%
User 03	97.22%-100.00%	0.00%-3.33%	0.00%	0.00%-1.67%
User 04	97.69%-100.00%	0.00%-11.11%	0.00%-2.78%	0.00%-5.56%
User 05	90.28%-99.07%	1.11%-11.67%	0.00%-2.78%	0.83%-5.83%
User 06	97.69%-100.00%	0.00%-3.89%	0.00%-2.78%	0.56%-1.94%
User 07	87.04%-95.83%	5.00%-15.56%	0.00%	2.78%-7.78%
User 08	98.61%-99.54%	0.56%-2.22%	0.00%	0.28%-1.11%
User 09	81.02%-89.35%	18.33%-21.67%	2.78%-5.56%	5.56%-13.61%
User 10	99.07%-100.00%	1.11%-2.22%	0.00%-2.78%	0.28%-1.67%
Overall	**94.58%-96.57%**	**4.89%-7.72%**	**0.00%-2.78%**	**5.78%-7.75%**

Table 4 Time Domain Before Optimising

The time domain model achieved the highest overall accuracy before optimisation, ranging from **94.58% to 96.57%.** This suggests that statistical features such as mean, variance, and range capture distinctive and stable movement patterns for most users.

- High performers – Users 03, 04, 06, 08, and 10 recorded near-perfect accuracy, often with 0.00% FRR. Many of their features (e.g., 61–66, 72–76, 84–88) showed strong inter-user variance and low intra-user variance, making them both unique and consistent.
- Lower performers – User 09 had the lowest accuracy (81.02%–89.35%) and the highest FAR (up to 21.67%). Several of their features (e.g., 62–74, 83–85) displayed high intra-user variance, meaning their behaviour changed more between training and testing sessions. User 07 also showed higher FAR (up to 15.56%), suggesting less distinctiveness compared to others despite having consistent features.

The large set of distinctive time-domain features (52–88) explains the generally strong results, but cases like Users 07 and 09 show that both stability and uniqueness are needed for consistent high accuracy.

Time-Frequency Domain

Users	Accuracy	FAR	FRR	EER
User 01	93.52%–95.37%	3.89%–6.11%	8.33%–11.11%	6.11%–8.33%
User 02	91.20%–91.67%	10.00%–10.56%	0.00%	5.00%–5.28%
User 03	100.00%	0.00%	0.00%	0.00%
User 04	90.28%–91.20%	10.56%–11.67%	0.00%	5.28%–5.83%
User 05	97.69%–99.54%	0.56%–2.78%	0.00%	0.28%–1.39%
User 06	94.91%–98.15%	2.22%–6.11%	0.00%	1.11%–3.06%
User 07	84.26%–87.96%	14.44%–18.89%	0.00%	7.22%–9.44%
User 08	97.22%–99.07%	1.11%–3.33%	0.00%	0.56%–1.67%
User 09	84.26%–87.50%	15.00%–18.89%	0.00%	7.50%–9.44%
User 10	99.07%–100.00%	0.00%–1.11%	0.00%	0.00%–0.56%
Overall	**94.03%–94.26%**	**6.67%–7.00%**	**0.83%–1.11%**	**3.78%–3.92%**

Table 5 Time- Frequency Domain Before Optimising

The combined time–frequency domain (TDFD) achieved an average accuracy of **94.03%–94.26%**, making it the most balanced feature set before optimisation. It merges distinctive spectral features from the frequency set (2, 4, 10, 13, 14, 18, 20, 26, 28, 32, 34, 40) with rich statistical features from the time set (52–88) and unique TDFD-only features (94, 95, 103–109, 113–119, 124–131).

- High performers – Users 03, 05, 08, and 10 achieved perfect or near-perfect accuracy, with 0.00% EER in some cases. These users had multiple low intra-user variance and high inter-user variance features across both domains, providing strong identifiers.
- Lower performers – Users 07 and 09 recorded lower accuracy (84.26%–87.96%) and higher FAR (up to 18.89%). This suggests that, despite the added robustness from combined features, they still lacked enough distinctive traits across sessions to separate them reliably from impostors.

Compared to single-domain results, TDFD reduced FRR to 0.83%–1.11% for most users, indicating better recognition of genuine samples. This supports the idea that combining domains offsets the weaknesses of each, delivering a better balance between accuracy, FAR, and FRR.

Comparative Summary

TDFD delivered the most balanced results before optimisation, with accuracy of 94.03%–94.26%, low FRR (0.83%–1.11%), and the lowest EER (3.78%–3.92%). Its strength came from blending stable time-domain features (52–88) with distinctive frequency-domain features (2, 4, 10, 13, 14, 18, 20, 26, 28, 32, 34, 40) and unique TDFD-specific features (94, 95, 103–109, 113–119, 124–131).

The Time Domain achieved the highest peak accuracy (94.58%–96.57%) but was more affected by intra-user variance, leading to higher FAR and EER for users like User 09.

The Frequency Domain performed well for certain users but showed greater variability, with higher FAR in cases like User 07, where spectral features were less distinctive.

By combining statistical and spectral information, TDFD reduced errors across users and offered the best trade-off between accuracy and reliability, making it the most suitable domain for optimisation.

Optimization

Pre-optimisation results showed that the Time–Frequency Domain (TDFD) achieved the most balanced performance across all metrics, making it the strongest candidate for refinement. However, there was still scope for improvement: some users (notably User 07 and User 09) recorded higher error rates, and overall, FAR and FRR could be lowered further. The optimisation process aimed to:

- Improve the network's ability to separate genuine and impostor samples.
- Reduce overfitting and improve generalisation.
- Achieve a better balance between security (low FAR) and usability (low FRR).

Time-Frequency Domain Results (After Optimization)

User	Accuracy (%)	FAR (%)	FRR (%)	EER (%)
User 01	96.30 – 98.15	1.67 – 3.89	2.78	2.22 – 3.33
User 02	98.61 – 100.00	0.00 – 1.67	0.00	0.00 – 0.83
User 03	100.00	0.00	0.00	0.00
User 04	100.00	0.00	0.00	0.00
User 05	97.69 – 99.54	0.56 – 2.22	0.00 – 2.78	0.28 – 2.50
User 06	99.54 – 100.00	0.00 – 0.56	0.00	0.00 – 0.28
User 07	96.76 – 99.07	1.11 – 3.89	0.00	0.56 – 1.94
User 08	98.15 – 100.00	0.00 – 1.67	0.00 – 2.78	0.00 – 2.22
User 09	89.35 – 96.76	3.33 – 12.22	2.78 – 5.56	3.06 – 7.78
User 10	100.00	0.00	0.00	0.00
Overall	**98.19 – 99.03%**	**1.06 – 2.00%**	**0.56 – 1.11%**	**0.81 – 1.44%**

Table 6 Time- Frequency Domain After Optimising

After optimisation, average accuracy improved substantially compared to the pre-optimisation TDFD (94.03%–94.26%). Several users (02, 03, 04, 06, 08, and 10) achieved perfect accuracy with zero error rates. Previously weaker performers such as User 07 and User 09 recorded notable reductions in errors, although User 09 retained a slightly elevated FRR, indicating some feature instability across days.

Optimisation Steps and Justification

1. **Network Architecture Adjustment** - The original FFMLP with a single hidden layer (10 neurons) was replaced with a deeper configuration [30, 20, 15]. This hierarchy allowed broad feature capture in the first layer, refinement in the second, and fine-grained discrimination in the third - ideal for complex, high-dimensional TDFD features.
2. **Feature Scaling** – mapminmax normalisation ensured all features contributed equally, preventing large-valued features from biasing training and improving convergence stability.
3. **Threshold Adjustment** - Instead of a fixed 0.5 decision threshold, a sweep of 100 values between 0 and 1 identified the point where FAR and FRR were closest (Equal Error Rate), improving the security–usability trade-off.
4. **Consistent Pre-processing** - Standardised data handling across all users to remove variability between runs and ensure reproducibility.
5. **Feature Selection with ReliefF** - From 131 original TDFD features, the top 50 most discriminative were retained. This reduced dimensionality, removed redundant or noisy features, improved generalisation, and cut computational load - particularly benefiting users with high intra-user variance.

Why This Configuration Was Effective

- Deeper architecture enabled more detailed learning of complementary time and frequency patterns.
- Normalisation stabilised and balanced the training process.
- Threshold tuning optimised performance for both usability and security.
- ReliefF feature selection reduced overfitting and improved consistency across users.

This combination produced a high-performing, robust TDFD model capable of accurate cross-day authentication with minimal false acceptances and rejections.

Conclusion

This project examined acceleration-based behavioural biometrics for continuous user authentication, comparing Time Domain, Frequency Domain, and Time–Frequency Domain (TDFD) features with a Feedforward Multi-Layer Perceptron (FFMLP). Pre-optimisation testing showed TDFD provided the best balance of accuracy, FAR, FRR, and EER, making it the strongest candidate for refinement.

Optimisation steps - including a deeper network architecture, feature scaling, threshold tuning, and ReliefF feature selection - significantly improved performance. Post-optimisation, TDFD achieved average accuracies of 98.19-99.03% with very low error rates, and several users reached perfect accuracy with zero errors. These results indicate strong potential for secure, unobtrusive, and continuous authentication in practical applications.

However, the study's scope was limited by a small dataset of ten users and a narrow activity range, which may not represent the diversity of real-world usage. Performance could also be affected by sensor inconsistencies and environmental noise.

Future work should expand the dataset, include more varied devices and activities, and investigate adaptive learning to handle behavioural changes over time. Broader testing in less controlled conditions would provide a more realistic evaluation, strengthening the case for deploying TDFD-based authentication in everyday use.

References

- Al-Naffakh, N., Clarke, N.L. and Li, F. (2020) Activity-Based User Authentication Using Smartwatches. Cham: Springer.
- Dos Santos, U.J.L., Munoz, R., Do Nascimento, E.P. and Oliveira, A.L.I. (2022) ‘Trends in user identity and continuous authentication’, IEEE Security & Privacy, 20(3), pp. 52–61. doi:10.1109/MSEC.2021.3136878
- Kang, T., Ji, S., Jeong, H., Zhu, B. and Kim, J. (2019) ‘WearAuth: Wristwear-assisted user authentication using wavelet-based multi-resolution analysis’, IEICE Transactions on Information and Systems, E102.D(10), pp. 1976–1992. doi:10.1587/transinf.2019EDP7021.
- Pelto, B., Vanamala, M. and Dave, R. (2023) ‘Your identity is your behaviour: Continuous user authentication based on machine learning and touch dynamics’, Machine Learning and Knowledge Extraction, 5(2), pp. 550–570. doi:10.3390/make5020030.
- Siddiqui, N., Dave, R., Vanamala, M. and Seliya, N. (2022) ‘Continuous user authentication using mouse dynamics and deep learning’, Machine Learning and Knowledge Extraction, 4(1), pp. 105–126. doi:10.3390/make4010006.

Appendix

MATLAB CODE (TDFD After Optimisation)

```
clear;

numUsers = 10;
numFeatures = 131;  % Total features in TDFD
topK = 50;          % Number of selected features

% Initialize performance metric lists
accuracyList = zeros(1, numUsers);
farList = zeros(1, numUsers);
frrList = zeros(1, numUsers);
eerList = zeros(1, numUsers);

for user = 1:numUsers
    % Load training and testing data (TDFD)
    trainData = load(sprintf('User%02d_TDFD_train_Template.mat', user)).trainData;
    testData  = load(sprintf('User%02d_TDFD_test_Template.mat', user)).testData;

    % Combine for feature selection
    X = [trainData(:, 1:numFeatures); testData(:, 1:numFeatures)];
    Y = [trainData(:, end); testData(:, end)];

    % Apply ReliefF to rank features
    [rankedIdx, ~] = relieff(X, Y, 10);
    selectedIdx = rankedIdx(1:topK);

    % Use only selected features
    X_train = trainData(:, selectedIdx)';
    Y_train = trainData(:, end)';
    X_test  = testData(:, selectedIdx)';
    Y_test  = testData(:, end)';

    % Normalize features
    [X_train, ps] = mapminmax(X_train);
    X_test = mapminmax('apply', X_test, ps);

    % Create FFMLP network
    net = patternnet([30 20 15]);
    net.performFcn = 'crossentropy';           % Set performance function to avoid
warning
    net.trainFcn = 'trainlm';                  % Levenberg-Marquardt
    net.trainParam.epochs = 100;
    net.trainParam.showWindow = true;
    net.divideParam.trainRatio = 1.0;
    net.divideParam.valRatio = 0;
    net.divideParam.testRatio = 0;

    % Train the network
    [net, ~] = train(net, X_train, Y_train);

    % Predict outputs
    scores = net(X_test);

    % Threshold sweep for FAR and FRR
    thresholds = linspace(0, 1, 100);
    FAR = zeros(size(thresholds));
    FRR = zeros(size(thresholds));

    targetScores = scores(Y_test == 1);
    imposterScores = scores(Y_test == 0);

    for i = 1:length(thresholds)
        t = thresholds(i);
        FAR(i) = sum(imposterScores > t) / length(imposterScores);
        FRR(i) = sum(targetScores <= t) / length(targetScores);
    end

    % Equal Error Rate (EER)
    [~, idx] = min(abs(FAR - FRR));
    optimalThreshold = thresholds(idx);
    EER = (FAR(idx) + FRR(idx)) / 2;

    % Final prediction
    predictions = scores > optimalThreshold;
    acc = mean(predictions == Y_test) * 100;

    % Store metrics
    accuracyList(user) = acc;
    farList(user) = FAR(idx) * 100;
    frrList(user) = FRR(idx) * 100;
    eerList(user) = EER * 100;

    % Display per-user result
    fprintf('User %02d → Accuracy: %.2f%% | FAR: %.2f%% | FRR: %.2f%% | EER: 
%.2f%%\n', ...
        user, acc, FAR(idx)*100, FRR(idx)*100, EER*100);
end

% Final summary
fprintf('\n=== Overall FFMLP Performance (TDFD with Feature Selection) ===\n');
fprintf('Average Accuracy: %.2f%%\n', mean(accuracyList));
fprintf('Average FAR: %.2f%%\n', mean(farList));
fprintf('Average FRR: %.2f%%\n', mean(frrList));
fprintf('Average EER: %.2f%%\n', mean(eerList));
```

Project Files: Ashen Pulle-10899670