ARTEFATO DISPONÍVEL
SBSeg25
FOZ DO IGUAÇU
CONFIRA OS SELOS AQUI ↑

How does reducing the dimension of feature vectors impact Biometric Systems that use Homomorphic Encryption?

Andreis G. M. Purim

Institute of Computing - State University of Campinas

a213095@dac.unicamp.br

Abstract. *Homomorphic Encryption enables biometric systems to perform matching directly on encrypted feature vectors, preserving user privacy throughout the process. However, the high computational cost of encrypted-domain operations, especially on high-dimensional inputs, remain a major barrier to real-world use. This study presents a concrete example of how reducing the dimensionality of biometric feature vectors affects both matching accuracy and runtime, and discusses how system designers could estimate the trade-off between time savings and accuracy loss when choosing a target dimension. The code is available on Github*[1].

1. Introduction

Biometric systems rely on numerical feature vectors that encode traits such as faces or fingerprints to authenticate users. These vectors are sensitive: if leaked, they may allow attackers to impersonate users or reconstruct biometric data - and unlike passwords, biometric traits cannot be changed. Protecting this data during both storage and matching is therefore essential. To address this, there have been several proposed techniques for creating *privacy-preserving biometric systems*. One such technique is using Homomorphic Encryption (HE), which enables secure matching by allowing computations to be performed directly on encrypted data. A server can compute similarity scores between encrypted samples without ever knowing the underlying feature vectors. In principle, this achieves ideal confidentiality: the biometric match occurs without the data ever being exposed [Melzi et al. 2024, Arman et al. 2024].

However, HE is computationally expensive, especially Fully Homomorphic Encryption (FHE) schemes. Matching encrypted feature vectors requires large volumes of data (in ciphertext) and costly arithmetic operations. In practical systems, latency scales with vector length. For instance, comparing two 128-dimensional vectors may take hours - far too slow for real-time applications, and even slower in constrained environments. One way to reduce this cost is to shorten the feature vectors before encryption. Dimensionality reduction is a common technique in traditional biometric systems, where methods such as Gaussian Random Projection (GRP) or Principal Component Analysis (PCA) are used to reduce storage, runtime, and computational load with minimal accuracy loss [Zebari et al. 2020]. However, while some recently proposed biometric systems with HE apply compression before encryption, such as HERS [Engelsma et al. 2022], none has systematically analyzed the trade-off between accuracy and runtime across varying dimensions. Figure 1 presents a simplified overview of the feature extraction, dimensionality reduction, encryption, and matching process analyzed in this study.

[1] Code can be found at `https://github.com/AndreisPurim/HEDimensionality` or contacting the author in both his institutional or personal email (`andreispurim@proton.me`).

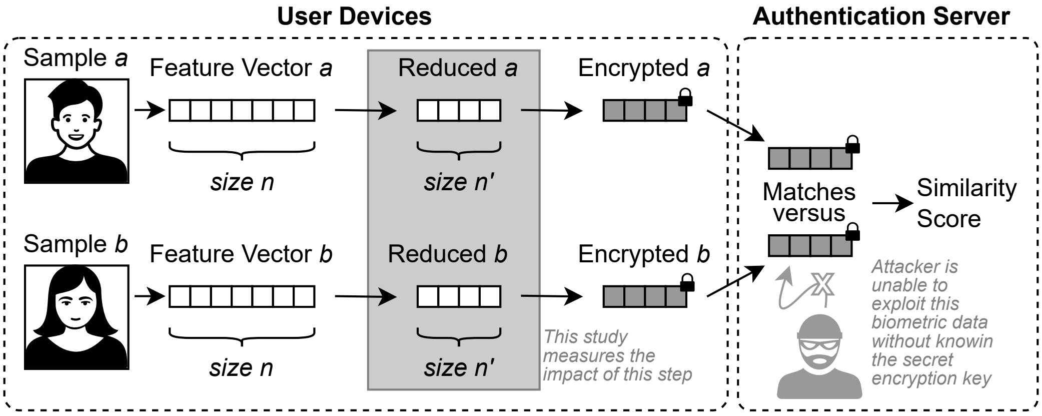

Figure 1. Experimental results using PEBA

In non-encrypted biometric systems, dimensionality reduction is commonly used to improve computational speed and reduce memory usage, often at the cost of decreased accuracy. As the feature vectors become smaller, discriminative information is lost, leading to higher error rates. When applying dimensionality reduction in encrypted systems, however, additional questions emerge: **(i)** Are the effects of dimensionality reduction in homomorphic systems equal to those in plaintext systems? **(ii)** Does the choice of HE scheme influence the effects of dimensionality reduction? **(iii)** Do encryption-specific factors, such as noise growth, ciphertext packing, or arithmetic precision, increase the accuracy loss? **(iv)** What are the efficiency gains in runtime when reducing feature length? **(v)** How far can dimensionality be reduced before the accuracy loss outweighs the computational benefit? **(vi)** And finally, is it possible to estimate this trade-off in advance, during system design, before performing full encrypted computation?

This study presents a quantitative analysis of dimensionality reduction in encrypted biometric matching by generating synthetic 128-dimensional vectors, applying Gaussian Random Projection (GRP) to reduce their size, and measuring accuracy and runtime at each dimension using the PEBA system matcher [Pradel and Mitchell 2021]. The results show a clear trade-off: runtime decreases linearly with feature size, while error rates increase super-linearly, following an exponential-like growth pattern. These results, along with preliminary insights into the questions above, are discussed in the following sections.

2. Related Work

Recent studies have explored the intersection of homomorphic encryption, biometric matching, and dimensionality reduction. A survey by Yang et al. reviewed the current landscape of privacy-preserving biometric systems that employ homomorphic encryption and highlighted feature dimensionality as a key factor influencing performance in the encrypted domain [Yang et al. 2023].

Approaches to dimensionality reduction vary depending on the research objective. In some cases, reduction is combined with additional processing steps such as quantization or normalization; in others, it is applied independently. Techniques may rely on classical methods like PCA or random projection, or on learned transformations, and can be either domain-specific or general-purpose. The following studies exemplify these different strategies:

Reduce-then-Encrypt: Boddeti proposed a face matching system using Fully Homomorphic Encryption, applying PCA to reduce 512-dimensional embeddings before encryption. This approach enabled matching in approximately 0.01 seconds with minimal accuracy loss [Naresh Boddeti 2018]. Building on this idea, Engelsma et al. introduced the HERS system, which uses DeepMDS++, a learned dimensionality reduction technique that compresses biometric features to near-intrinsic dimensionality prior to encryption, achieving a 6× speed-up with only a 0.1% drop in rank-1 accuracy on 192-dimensional vectors [Engelsma et al. 2022].

Coefficient Packing and Reduction: Bauspieß et al. proposed an efficient system that combines coefficient packing with dimensionality reduction to improve the scalability of homomorphically encrypted biometric identification. Their method applies feature compression to reduce ciphertext size and enables multiple biometric comparisons within a single ciphertext operation, achieving a quadratic reduction in computational workload with respect to the number of templates [Bauspies et al. 2022]. Meanwhile, Jindal et al. applied random projection to reduce 128-dimensional FaceNet embeddings before encryption and packed multiple augmented samples into a single Cheon-Kim-Kim-Song (CKKS) ciphertext [Cheon et al. 2017], enabling efficient encrypted matching via cosine similarity without quantization [Jindal et al. 2020].

Encrypt-then-Reduce: While most prior work applies dimensionality reduction before encryption, Ma et al. proposed a PCA algorithm that operates directly in the encrypted domain, using optimized homomorphic matrix multiplication [Ma et al. 2024]. Similarly, Sperling et al. introduced a system that performs all steps (including feature concatenation, dimensionality reduction via a learned linear projection, normalization, and match score computation) entirely within ciphertexts, enabling end-to-end encrypted processing of biometric vectors [Sperling et al. 2022].

This work does not propose a new algorithm or technique for dimensionality reduction. Instead, it explores a "naïve" Reduce-then-Encrypt approach using existing methods for both homomorphic biometric matching and dimensionality reduction in a controlled setup, to better understand how feature vector length affects accuracy and runtime under FHE. While other studies incorporate dimensionality reduction as part of broader system designs, this work focuses specifically on analyzing the trade-off between efficiency and accuracy in a systematic and isolated manner. This paper aims to fill that methodological gap through a simple, empirical study.

For homomorphic biometric matching, this study uses the PEBA algorithm introduced by Pradel and Mitchell, which implements a squared Euclidean distance function in C++ using the TFHE library [Chillotti et al. 2018]. While PEBA demonstrates secure matching over encrypted face vectors, it also reveals the high computational cost of such operations: in their experiments, computing a single 128-dimensional distance took 33,536 seconds (over 9 hours) [Pradel and Mitchell 2021]. PEBA was chosen for this study due to its clear performance limitations, reproducible results, and accessible open-source implementation.

Although several more efficient homomorphic matching algorithms have been proposed (such as those mentioned previously in the related work section) these alternatives are not explored here. By intentionally adopting a slower, well-understood baseline, this study isolates the specific impact of dimensionality reduction on runtime and accuracy. Future work could extend this analysis to faster algorithms or approximate schemes, such as those based on CKKS, to evaluate whether a similar trade-off as the ones evaluated in this study hold.

3. Methodology, Results and Discussion

This study adopts the perspective of a naïve system designer, someone who does not modify the internal components of the system but instead relies on an existing homomorphic biometric matcher. The same designer also depends on pre-existing feature extractors to generate the input vectors. This constrained setup is intentional: by limiting the variables, it becomes possible to isolate and analyze the direct impact of dimensionality reduction on both runtime and matching accuracy under FHE. The goal is not to optimize the pipeline, but to understand how the simple design decision of reducing feature vector length affects overall system behavior.

The following steps were carried out to measure how dimensionality reduction affects matching accuracy and execution time in a system based on PEBA. All experiments were performed using PEBA's default parameters on a Linux Mint 21.2 (64-bit) machine with 8 GB RAM and a quad-core Intel Core i5-4670 CPU running at 3.40 GHz.

- **Step 1.** A synthetic dataset of 100 feature vectors was generated, each of length $n = 128$, with integer values in the range $[0, 255]$. The vectors were divided into 10 classes (users), with 10 samples per class. These vectors simulate latent embeddings from models such as FaceNet or VGGFace2, without relying on any specific extraction or encoding scheme.
- **Step 2.** For each target dimension $n' \in \{128, 112, 96, \ldots, 16, 8, 4, 2, 1\}$, the dataset was reduced using Gaussian Random Projection. The resulting vectors were rescaled to the range $[0, 255]$ and rounded to the nearest integer, producing compressed vectors of length n' compatible with PEBA's encryption interface.
- **Step 3.** To establish the expected distances and error rates, the squared Euclidean distance function $d(v_i, v_j) = \sum_{k=1}^{n'} (v_i[k] - v_j[k])^2$ was implemented in Python and used to compute all pairwise distances in plaintext. For each target dimension n', each of the 100 vectors was compared against all others, yielding $100 \times 100 = 10.000$ distance values per dimension.
- **Step 4.** A decision threshold τ was varied from zero to the maximum observed distance. For each value of τ, a match was labeled as an *accept* if $d < \tau$, and as a *reject* otherwise. Using the known class labels, the False Accept Rate (FAR) was computed as the proportion of inter-class pairs incorrectly accepted, and the False Reject Rate (FRR) as the proportion of intra-class pairs incorrectly rejected. The Equal Error Rate (EER) was defined as the point where FAR equals FRR.
- **Step 5.** The PEBA repository was cloned[2], and its C++ matching function was adapted to accept two vectors of variable length n' ($n' \leq n$) and compute their match using homomorphic encryption. The function encrypts both vectors and calculates their squared Euclidean distance in the encrypted domain using the TFHE library[3] [Chillotti et al. 2016]. The same pairwise matches performed in

[2] https://github.com/lab-incert/peba1/tree/da28120

[3] https://github.com/tfhe/tfhe/tree/bc71bfa

Step 3 were repeated under encryption[4], confirming that both the distances and error rates remained identical to the non-encrypted matches[5]. To evaluate the average runtime t', the runtime of each homomorphic distance computation was measured, excluding all preprocessing steps such as encryption, decryption, or vector encoding.

Figure 2 presents the results of the experiments. Subfigure (2.a) plots the FAR and FRR curves for three selected feature vector sizes: $n' = 128$ (no dimensionality reduction), $n' = 16$ (moderate reduction with very good performance), and $n' = 2$ (extreme reduction). At $n' = 2$, the EER is close to 50%, indicating that the classifier performs no better than random guessing. The x-axis has been normalized for visualization purposes. Subfigure (2.b) shows how the EER changes as a function of the feature vector dimension. The EER increases nonlinearly as the dimensionality decreases, starting near 0% and approaching 50% in the most compressed dimensions (an EER of 50% means the matcher is equivalent to a random classifier, unable to tell genuine and impostor samples apart). This reflects the degradation in matching scores caused by excessive dimensionality reduction.

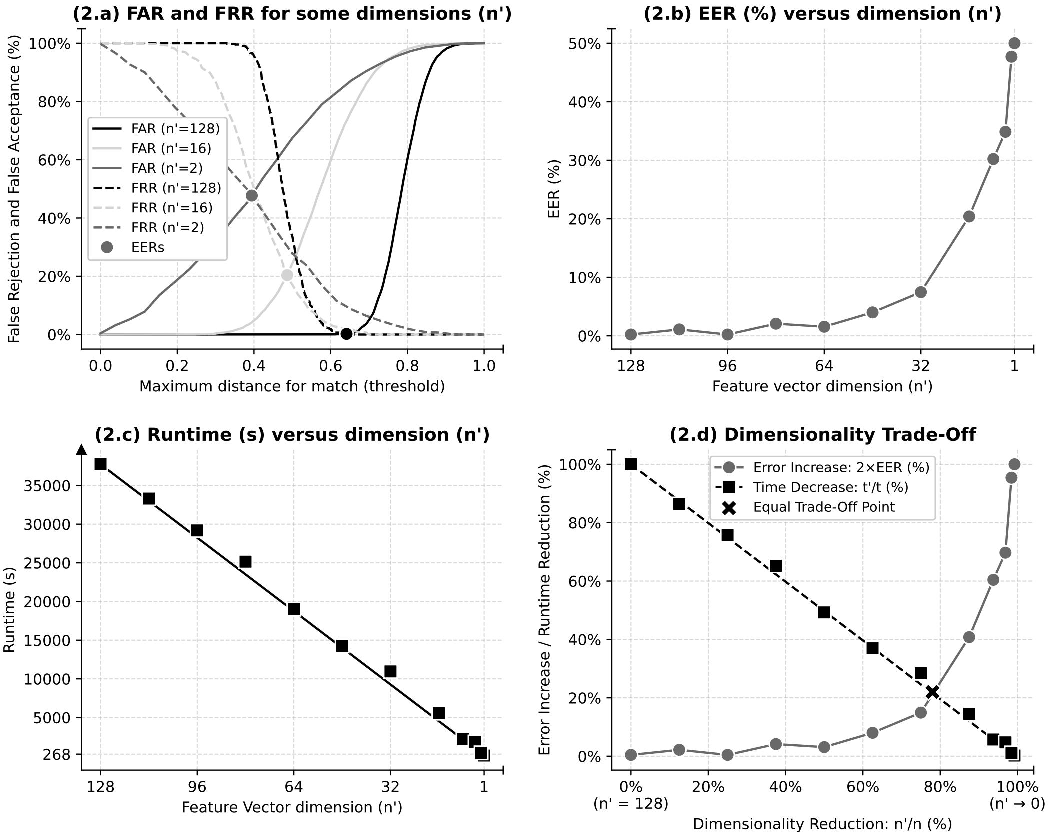

Figure 2. Experimental results of accuracy and runtime using PEBA

[4] Without parallelization or multiprocessing, performing all pairwise encrypted matches across all dimensions may require several weeks. Readers are encouraged to parallelize the workload for faster execution.

[5] As discussed later, since TFHE performs exact integer arithmetic, the encrypted and non-encrypted distances are identical, making encrypted accuracy evaluation theoretically redundant. Nonetheless, both were performed in this study for completeness.

On the other hand, Subfigure (2.c) displays the average runtime t' (in seconds) required for homomorphic matching as a function of the feature vector dimension. A linear fit is plotted alongside the measurements to illustrate the approximately $O(n)$ behavior of the algorithm with respect to feature length. As the dimension decreases, the runtime drops proportionally, confirming the expected correlation between vector size and computational cost in a Euclidean distance algorithm. Finally, Subfigure (2.d) presents both the execution time and EER curves together for direct comparison. The execution runtime was normalized to the baseline value (t'/t, where $t = 37740s$ corresponds to the 128-dimensional vector average runtime), and EER values were scaled by a factor of 2 to map the range 0–50% onto 0–100%. The x-axis represents the dimensionality reduction ratio (n'/n, where $n = 128$).

As shown in Subfigure (2.d), the percentual decrease in runtime is a direct consequence of the $O(n)$ complexity of the matching algorithm (in this case, the squared Euclidean distance) and the size of the input vectors. Because this operation scales linearly with vector length, the runtime decreases approximately linearly as dimensionality is reduced. While the choice of dimensionality reduction method may affect the slope of this curve, it does not change its overall behavior. Thus, the speedup from reducing dimensions depends on the complexity of the matching function, not on the HE scheme or the reduction algorithm itself.

The intersection of the two curves (where the percentage decrease in runtime matches the percentage increase in EER) was marked as a *"Dimensionality Equal Trade-Off"* point, intended mostly as a descriptive metric (analogous to the EER) that illustrates the shape of the trade-off curve. However, it is not meant to serve as a prescriptive recommendation for system design[6]. In the experiments, this point occurs at around 20% EER, which is clearly too high for most biometric applications. Determining what constitutes an acceptable error rate depends heavily on the application context and associated risk tolerance. For example, one might accept 2–5% EER in high-security systems if it yields significant runtime improvements. Future work may propose better metrics or methodologies for selecting an appropriate dimension.

Meanwhile, the increase in error rates results from the loss of discriminative information in lower-dimensional vectors. Since TFHE performs exact integer arithmetic, the Equal Error Rate (EER) observed in ciphertext matches that of the plaintext domain. This indicates that the degradation in accuracy is determined by the dimensionality reduction and the data, not by the encryption scheme. Moreover, no encryption-specific artifacts, such as noise growth, were observed to influence either runtime or accuracy. Therefore, in this PEBA-based setup, dimensionality reduction behaves independently of the TFHE scheme. As a result, the same analysis and design considerations used in a non-encrypted system can be applied, bearing in mind the higher baseline runtime introduced by homomorphic computation.

This independence between the TFHE scheme and performance behavior in regards to dimensionality reduction allows key design trade-offs to be estimated in plaintext. Because runtime depends primarily on the inherent algorithmic complexity and vector size, and error metrics remain constant between plaintext and encrypted domains under

[6] This trade-off point should be interpreted as an indicator of *how the system behaves when the dimension is reduced*, rather than a suggestion that this point is optimal or desirable.

exact schemes like TFHE, a system designer could simulate trade-offs before implementing encryption. This could enable early-stage decisions, such as selecting a target feature length, to be guided by fast experiments without needing to benchmark the entire system in HE. It should be noted, however, that there is no guarantee this TFHE-independence in accuracy holds for schemes based on approximate arithmetic, such as CKKS (e.g., a matcher using a non-squared Euclidean distance). In such cases, the introduction of floating-point errors or noise accumulation may affect both accuracy and runtime differently. This remains an open question for future investigation.

4. Conclusion

This work examined how a naïve approach to dimensionality reduction affects both accuracy and runtime in homomorphic biometric matching systems. Using synthetic feature vectors and the PEBA implementation with TFHE, matching accuracy and execution time were measured across a range of reduced dimensions. The results demonstrated a consistent trade-off: as dimensionality decreases, computation time drops approximately linearly, while error rates increase super-linearly.

Importantly, these effects appear to be independent of the homomorphic encryption scheme itself. Since TFHE performs exact computations, the accuracy loss can be attributed to the reduced representational capacity of the feature vectors, not to encryption-induced artifacts. Likewise, the percentage runtime speedup stems from the PEBA's inherent $O(n)$ algorithmic complexity and vector size, not from the encryption scheme. This suggests that system designers could estimate these trade-offs in plaintext, enabling faster and more accessible design iterations. However, this may not hold for schemes or algorithms based on approximate arithmetic or floating-point algorithms.

Future work includes simulating and exploring other dimensionality reduction methods such as PCA or learned encoders, and other biometric datasets. Different matching algorithms could be evaluated to observe how the trade-off behaves under different runtime profiles and different reduction methods, as well as different encryption schemes such as CKKS.

5. Acknowledgements

The author would like to thank José Roberto Junior, Dr. Caue Zaghetto, and Dr. Hilder Vitor Lima Pereira for the insightful conversations that led to the creation of this paper, and Paulo Pacitti for his valuable feedback during its writing stages.

References

Arman, S. M., Yang, T., Shahed, S., Mazroa, A. A., Attiah, A., and Mohaisen, L. (2024). A comprehensive survey for privacy-preserving biometrics: Recent approaches, challenges, and future directions. *Computers, Materials Continua*, 78(2):2087–2110.

Bauspies, P., Olafsson, J., Kolberg, J., Drozdowski, P., Rathgeb, C., and Busch, C. (2022). Improved homomorphically encrypted biometric identification using coefficient packing. In *2022 International Workshop on Biometrics and Forensics (IWBF)*, page 1–6. IEEE.

Cheon, J. H., Kim, A., Kim, M., and Song, Y. (2017). Homomorphic encryption for arithmetic of approximate numbers. In Takagi, T. and Peyrin, T., editors, *Advances in Cryptology – ASIACRYPT 2017*, pages 409–437, Cham. Springer International Publishing.

Chillotti, I., Gama, N., Georgieva, M., and Izabachène, M. (August 2016). TFHE: Fast fully homomorphic encryption library. https://tfhe.github.io/tfhe/.

Chillotti, I., Gama, N., Georgieva, M., and Izabachène, M. (2018). TFHE: Fast fully homomorphic encryption over the torus. Cryptology ePrint Archive, Paper 2018/421.

Engelsma, J. J., Jain, A. K., and Boddeti, V. N. (2022). Hers: Homomorphically encrypted representation search. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 4(3):349–360.

Jindal, A. K., Shaik, I., Vasudha, V., Chalamala, S. R., Ma, R., and Lodha, S. (2020). Secure and privacy preserving method for biometric template protection using fully homomorphic encryption. In *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, page 1127–1134. IEEE.

Ma, X., Ma, C., Jiang, Y., and Ge, C. (2024). Improved privacy-preserving pca using optimized homomorphic matrix multiplication. *Computers And Security*, 138:103658.

Melzi, P., Rathgeb, C., Tolosana, R., Vera-Rodriguez, R., and Busch, C. (2024). An overview of privacy-enhancing technologies in biometric recognition. *ACM Computing Surveys*, 56(12):1–28.

Naresh Boddeti, V. (2018). Secure face matching using fully homomorphic encryption. In *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, page 1–10. IEEE.

Pradel, G. and Mitchell, C. (2021). Privacy-preserving biometric matching using homomorphic encryption.

Sperling, L., Ratha, N., Ross, A., and Boddeti, V. N. (2022). Heft: Homomorphically encrypted fusion of biometric templates.

Yang, W., Wang, S., Cui, H., Tang, Z., and Li, Y. (2023). A review of homomorphic encryption for privacy-preserving biometrics. *Sensors*, 23(7):3566.

Zebari, R., Abdulazeez, A., Zeebaree, D., Zebari, D., and Saeed, J. (2020). A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction. *Journal of Applied Science and Technology Trends*, 1(1):56–70.