Power System Technology
ISSN:1000-3673

Received: 16-01-2024 Revised: 12-02-2024 Accepted: 07-03-2024

Exploring the Potential of Secure Signature Verification by employing a Trio Integrated Approach using GAN, Kernelized Biohashing and BiLSTM

GD Makkar[1], Suman Pant[2]*
gdmakkar@gmail.com[1], suman_joshi_1@yahoo.co.in[2]
[1,2]School of CA & IT, Shri Guru Ram Rai University, Dehradun, Uttarakhand – 248001, India
*Corresponding Author

Abstract: In the realm of digital authentication, the security and accuracy of signature verification systems are paramount. This paper proposes a new approach for signature verification that is able to greatly improve the security and reliability through the synergy given by Generative Adversarial Networks (GANs), Kernelized Biohashing, and Bidirectional Long Short-Term Memory (BiLSTM) networks. The system proposed in this paper is based on cancellable biometrics which is a novel method that converts biometric data into new, secure, revocable domains, while allowing the user's privacy to be protected from any kind of misuse of data. First, we used GANs to generate a large and diverse synthetic signature dataset. This dataset is not only rich in quantity and diversity but also closely modeled on the genuine signatures' complex patterns and variations. The synthetic signatures go through Kernelized Biohashing, successfully converting them into a secure and encrypted form, preserving privacy while being able to capture essential integrity in signature biometric verification processes. The biohaseddata isanalyzed through the verification process using the BiLSTM networks. BiLSTM networks allow analysis of temporal dynamics with other attributes of each signature, in processing sequences and obtaining long-term dependencies. Our system applies these networks in bio-hashed data and consequently attains good accuracy in the verification of signature, thus distinguishing well between a genuine and a forged signature. In this integrated framework, the identified mechanisms assure the security and reliability of the digital authentications not only for the acute security and reliability challenges being faced by the traditional signature verification systems—such as forgery, data breach, and others—but they also set a new benchmark altogether. Our approach takes a significant leap forward in protecting user signatures while ensuring, through the application of cancelable biometrics and advanced machine learning techniques, the integrity of verification processes.

Keywords: Generative Adversarial Networks, Synthetic Data, Kernelized Biohashing, Bi-directional Long Short-Term Memory, Signature Verification.

I. Introduction

Signatures become very important, not only to ensure the authenticity of identity but also to authorize transactions. Thus, the signature data should have to be secured due to the misuse that could arise from cases involving fraudulent activities. Signature verification is one of the

Power System Technology
ISSN:1000-3673
Received: 16-01-2024 Revised: 12-02-2024 Accepted: 07-03-2024

critical processes in order to ensure that all the signatures used in different applications are genuine, so that the integrity of the transactions and the safety of the person from identity theft can be maintained.

The proposed approach integrates Generative Adversarial Networks (GANs)[1] with Kernelized Biohashing[6][47] and Bi-directional Long Short-Term Memory (BiLSTM) networks [46]. Broadly, this paper covers a broad methodology that tries to include the potential of such advanced technologies, which can become handy tools in this age, to address the challenges presented in traditional methods of signature verification and the need to cover those challenges of reaching high accuracy in biometric verification.

Generative Adversarial Networks (GANs) have grown to be a viable method of producing synthetic data that can closely match real-world datasets with high accuracy [3], consequently, to some extent, offering a strategic solution to challenges in using sensitive personal information. As a result, GANs provide an important means for the creation of synthetic signatures that still retain the inherent features of real signatures while protecting privacy, hence enhancing security in biometric data. In this case, the use of GANs has made it possible to reduce risks that may be exposed to the exposure of private personal details.

Kernelized Biohashing is applied after the generation of synthetic signatures to derive its form in a secured, compact, binary manner. This transformation process greatly increases the security level of the data because it creates a biohash [21][23], which can resist many attack categories and assures biometric data integrity. Kernelized biohashing[34] is an advanced version of traditional techniques for biohashing. It brings in extra flavors of security, which can capture complex patterns present in data to make it adversarially difficult in the reconstitution of original data from its biohashed form.

In the final stage, BiLSTM networks [46] are applied to the biohashed data in order to be used for signature verification. BiLSTM networks are best to learn from sequential and temporal data to capture the dynamic features of a signature, which are very useful for its accurate verification. It applies the bidirectional processing of BiLSTM networks to enable quite accurate discernment between authentic and forged signatures, hence making up for the drawbacks often suffered by the traditional verification method from high variations in signature styles and conditions.

We present in this paper that GANs, in combination with kernelized Biohashing and BiLSTM networks, would present a promising approach that would yield a robust, secure, and efficient system for signature verification. This paper presents the contribution of each of these technologies to explores their potential in bringing revolution through advanced techniques in the field of biometric verification.

Power System Technology
ISSN:1000-3673

Received: 16-01-2024 Revised: 12-02-2024 Accepted: 07-03-2024

II. Literature survey

E. Brophy et al. [1] give an overview of Generative Adversarial Networks (GANs) in the application to time series data, describing the development, use, and associated challenges, including the differentiation of GANs types for discrete and continuous data, and discuss roles in data privacy. M. Arjovsky et al. [2] propose WGAN (Wasserstein GAN) to further stabilize training by avoiding mode collapse, improved loss metrics, and training methodologies. Further, D. Garcia Torres [3] highlighted in his thesis the potential of GANs in creating synthetic data that exactly replicate the statistical properties, therefore solving the problems with both benefits and computation challenges, giving a new solution to the conventional problems of data generation. All these works illustrate progress, practical applications, and persisting challenges within the field of GANs, from basic training improvements to novel techniques in synthetic data generation.

A novel Speech Retrieval Algorithm based on Bio-Hashing is proposed by Y.-B. Huang et al. [4], which refines retrieval accuracy and speed by segmenting the audio signals and removing silence and consequently processing the signal effectively in large databases. H. OtroshiShahreza et al. [5] explore the new territories of biometric security with an in-depth assessment of cancelable biometrics for all traits, which use measures such as BioHashing and MLP-Hashing to meet their stringent security standards. M. Kumar et al. [6] have undertaken a study for cancelable biometrics, focusing on several techniques of template generation that could help improve security and privacy for the biometric system. Their joint efforts now represent substantial advancements both in biometric security and speech retrieval technologies, enunciating significant contributions toward system efficiency, reliability, and, most importantly, user protection of privacy.

H. Li et al. [7] propose the Adversarial Variation Network (AVN) to enhance the performance of data generation and variation. The verification task of handwritten signatures, which requires better feature extraction and accuracy, H. Li et al. tried to resolve difficulties such as sparsity and style variation. N. Sharma et al. [8] describe a Siamese Convolutional Neural Network model for the task of writer-independent offline signature verification and demonstrate it over the GPDS dataset. M. Okawa [9] applied the Modified Dynamic Time Warping (MDTW) algorithm to the online signature verification task, accepting both local and global weighting in dealing with intra- and inter-variabilities. It has been evidenced that significant improvements in signature verification are realized on the SVC2004 Task2 dataset. Further, Okawa et al. [10] proposed a mean template set and boosting-based method of dynamic time warping distances gradient to improve verification accuracy by using mean information for intra-user variability. Their approach was able to improve on the current state-of-the-art on standard datasets. A. Singh et al. [11] applied a CNN-RNN to the task of verifying the online signature and obtained a testing accuracy of 97.05% on SVC 2004 and SigComp2009.

Power System Technology
ISSN:1000-3673

Received: 16-01-2024 Revised: 12-02-2024 Accepted: 07-03-2024

III. Methodology

We used the SVC2004 dataset [12], which contains signature time series from 44 users. Each of the users had submitted 20 genuine and 20 forged signatures, cumulatively amounting to 1760. The feature set consists of 7 features in total: X-coordinate, Y-coordinate, Time stamp, Button status, Azimuth, Altitude, and Pressure. Out of these original features, 25 features were further derived from the original, and after this, 62 aggregate features were computed from the signatures.

Our methodology is divided into three phases:

1. Synthetic Data Generation with GANs
2. Kernelized Biohashing
3. Signature Verification with BiLSTM

3.1 Synthetic Data Generation with GANs

Generative Adversarial Networks (GANs) generate new datasets that will look like the real data;thus, it ensures the safety of real data for security purposes [13]. It has two main components: the Generator and Discriminator. The objective of Generator to generate a new instance of data that is statistically as close as possible to real, authentic data, such that telling the two apart would be hard [14]. The authenticity of the new generated data by the Generator is evaluated by the Discriminator.

It endeavours to distinguish between genuine data derived from the training set and counterfeit data synthesized by the Generator [15].The generator takes random noise as input and producing synthetic data. The objective is to ensure that the generated data closely resembles the real data present in the dataset. While there exists a wide range of GAN types, we choose to use Wasserstein GAN (WGAN) for the stable training task.WGAN was introduced by M. Arjovsky et al.[2]proposed a new approach in Generative Adversarial Networks using the Wasserstein distance for more stable training, which can tackle issues like mode collapse and instability, usually found in a normal GAN. The method provides meaningful loss metrics, increases the stability of training, and reduces mode dropping—all of which lead to further diverse data generations, while not necessarily perfectly balanced required between discriminator and generator. This improvement vastly enhances the quality and diversity in the generated data, thus making WGAN quite a leap over conventional GANs. During training, the following are the key components of the GAN that are used:

(i) Wasserstein Loss Function: The basic Wasserstein Generational Network (WGAN) comprises the Wasserstein Loss Function. The Wasserstein's lost functions is obtained from the Wasserstein distance that is also known as s Earth Mover's distance. It measures the distance between the distribution between the synthetic data generated by GAN and real data [16].

(ii) Gaussian Noise Function

The objective of adding Gaussian noise is to increase variability and randomness in data, which can help further in avoiding the problem of overfitting and makes the

Power System Technology
ISSN:1000-3673
Received: 16-01-2024 Revised: 12-02-2024 Accepted: 07-03-2024

model more robust [17]. We used 'add_gaussian_noise' function to add the Gaussian noise in data.

(iii) Generator and Discriminator

The Generator's primary function is to generate new data instances that resemble the genuine data as closely as possible. We used **build_generator** function in Python for the generator. It takes a random noise vector (from latent space) as input and produces output with the same number of features as the real data. We used the dense layer and LeakyReLU Activation to build the generator.

It is the responsibility of the Discriminator to differentiate between authentic and fake data. The system assesses the integrity of the input data and categorises it as either genuine (as per the dataset) or faked (generated by the Generator) It is created using the **build_discriminator** function. It takes either real data or fake data generated by the Generator. Multiple dense layers, LeakyReLU Activation, and Dropout are required to build the discriminator.

3.1.1 Data Preprocessing:

Data preprocessing is a crucial step in machine learning. It involves transforming raw data into a format that is more suitable for modelling [18]. We used **MinMaxScaler** function from **Scikit-learn** is used to normalize of data. This scaler transforms each feature to a given range, which is typically between -1 and 1 [18][19].

3.1.2 Hyperparameter Tuning

Hyperparameters are the settings or configurations that govern the training process and structure of the model[20].Table (1) exhibits the hyperparameters tuned in our model to generate the synthetic data by GAN.

Table 1: Hyperparameters used to generate synthetic data by GAN

Dense layers with different numbers of neurons	256, 512, 1024 in the generator
Learning rate	0.00005
Beta Parameters for Adam Optimizer	beta_1=0.5 and beta_2=0.9
Dropout Rate	0.5
Batch size	32
Number of epochs	3000
Latent space dimensionality	100
Standard Deviation for Gaussian Noise	0.1

The table (2) displays the top 5 rows of the real data and synthetic data created using a Generative Adversarial Network (GAN) for the user 1.

Power System Technology
ISSN:1000-3673
Received: 16-01-2024 Revised: 12-02-2024 Accepted: 07-03-2024

Table 2: Data before and after applying GAN

Real Data							
Users	Pressure_mean	Azimuth_mean	Altitude_mean	Velocity in x_mean	Velocity in y_mean	duration	total_distance
User1_Sig1_Genuine	581.1235294	1179.470588	565.7647059	4.09987692	-1.37372549	1793	27.89976959
User1_Sig2_Genuine	684.2208589	1180.368098	592.6380368	4.808205521	-1.297784594	1742	28.45133667
User1_Sig3_Genuine	675.1309524	1218.154762	583.75	5.05487273	-1.473809524	1833	32.32231295
User1_Sig4_Genuine	715.96875	1181.9375	554.4375	5.374523046	-1.403958333	1732	29.59932544
User1_Sig5_Genuine	701.7175141	1205.367232	575.1977401	4.398216462	-1.648404786	1852	28.80141052

Synthetic Data generated by WGAN							
Users	Pressure_mean	Azimuth_mean	Altitude_mean	Velocity in x_mean	Velocity in y_mean	duration	total_distance
User1_Sig1_Genuine	989.80273	1690.1327	403.8358	-1.3837441	-9.883473	780	123.375854
User1_Sig2_Genuine	989.80273	1690.1327	403.8356	-1.3837441	-9.883764	780	123.375854
User1_Sig3_Genuine	480.68216	1237.1445	509.62323	2.8863056	-0.4031584	6429.0654	7.7303863
User1_Sig4_Genuine	326.68933	1128.7113	538.9315	1.8549639	-0.24179718	6818.9473	8.7800045
User1_Sig5_Genuine	440.68222	1099.4778	548.3561	1.2861543	-0.9684231	3370.738	20.192404

The figure (1) illustrates a visual comparison between real and synthetic data through histograms. It indicates that both real and synthetic data share similar distribution shapes across parameters like Pressure_mean, Azimuth_mean, Altitude_mean, Velocity in x_mean, Velocity in y_mean, duration, and total_distance, with central peaks and roughly normal distributions. However, synthetic data often shows narrower distributions, fails to fully capture the tails of real data distributions, and sometimes misaligns peaks, especially in metrics like Velocity and total_distance. While both data types generally align in central tendencies, discrepancies in spread, skewness, and tail length highlight differences in their distributional characteristics. Although the synthetic data generated by the GAN captures the general shape of the distributions in most cases, but there are differences in the spread, skewness, and tails of the distributions. This difference is important as we generated the synthetic data with some noise.

Power System Technology
ISSN:1000-3673

Received: 16-01-2024 Revised: 12-02-2024 Accepted: 07-03-2024

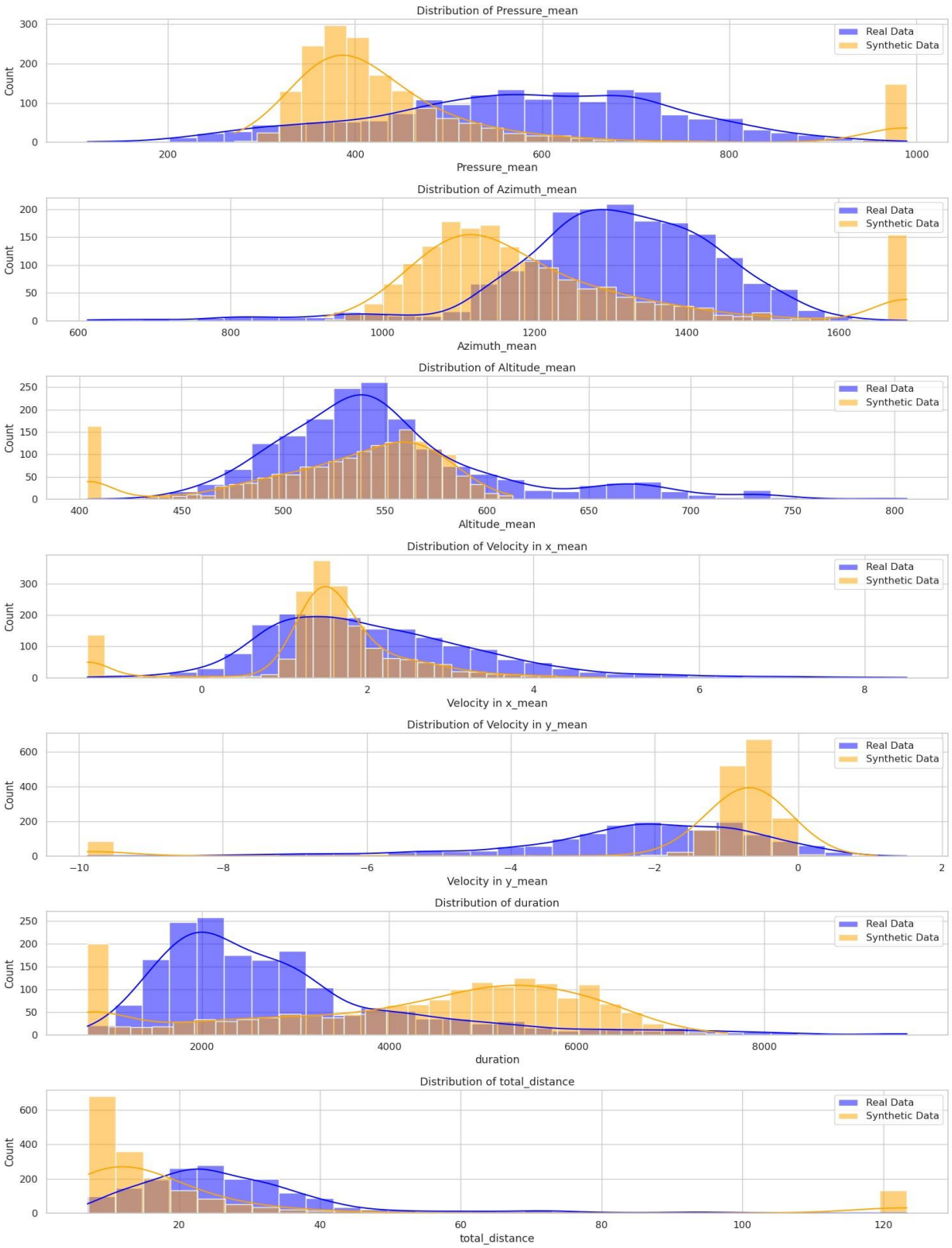

Figure 1: Visual comparison between real and synthetic data

Power System Technology
ISSN:1000-3673
Received: 16-01-2024 Revised: 12-02-2024 Accepted: 07-03-2024

3.2 Kernelized Biohashing

Biohashing is a cancelable biometrics technique that involves generating a biometric template by combining the biometric data with a user-specific token (like a password or a key). This process results in a secure and unique biohash that can be used for authentication purposes. When biohashing is applied to the biometric data such a signature's time series data, it produces new data i.e. template that can be revoked and replaced if compromised. This makes sure that actual biometric data is never stored or transmitted in its original form, which will provide better user privacy and security. We made biohashing more effective with the combination of biohashing and the Radial Basis Function (RBF) kernel. These, in turn, strengthen data security as biohashed data is pushed into a higher-dimensional, non-linear space. This way, one more layer of protection is added, and it increases the robustness of data against any potential security risks. The strategy that we applied for biohashing our data has been elaborated below.

3.2.1 Preprocessing and Normalization

In this stage, standardization of the data scale and format is applied to solve variability and heterogeneity problems common in raw biometric data, such as the value of the features of the signature [21]. Normalization gives each feature an equal chance to contribute to the resulting analysis, not allowing one single feature to dominate because of its scale. Such uniformity becomes important for the process of effectiveness and fairness in biohashing. We normalized the dataset using RobustScaler. RobustScaler methos is usually used for datasets with outliers, and it scales the feature values statistically that are robust to outliers [22]. RobustScaler uses the median and the interquartile range (IQR) in place of standard scaling methods that use mean and variance; hence, it is not affected by outliers when scaled.

3.2.2 Token Generation

Token generation is an important step in biohashing in order to enhance the level of security and privacy provided by the biometric authentication process. A token is generated randomly, which is unique to the user and will be combined with the biometrically normalized data of each user [23].Our dataset comprises 44 distinct user signature features. Each token must be unique to each user to ensure individualized security. we required to generate 44 tokens representing each individual user [24].We used pseudo-random number generator to generate random token for each user. Table (3) below shows the random token generated for five users.

Table 3: Random token generated for biohashing

User_ID	Token
User1	[0.01968858 0.09225385 0.53856012 0.11814145 0.89845163 0.68730998 0.45801974]
User2	[0.73212488 0.230313 0.88934299 0.29859344 0.44206529 0.09416282 0.85227507]
User3	[0.49341778 0.13459212 0.12154688 0.76645574 0.19611553 0.02909798

Power System Technology
ISSN:1000-3673
Received: 16-01-2024 Revised: 12-02-2024 Accepted: 07-03-2024

	0.34044791]
User4	[0.61438432 0.93693458 0.69508934 0.47672457 0.57418766 0.57231484 0.35697695]
User5	[0.91081989 0.39374395 0.71675897 0.32660566 0.27497934 0.39063227 0.56608073]

3.2.3 Feature Transformation

Feature transformation integrates the user's biometric data with their generated unique token in a way that ensures the resulting data is both secure and uniquely tied to the user. This step enhances the security of the biohashing process. We achieved feature transformation by applying matric multiplication [25]**.**

A token matrix ***T*** is generated randomly. ***T*** will have dimensions ***7 × n,*** where *n* is the total number of users i.e., 44 in our case. Matrix multiplication is applied between thenormalized data ***D*** and token matrix ***T*** that result into new matrix ***M*** of dimensions **1760×44**, where each element $\boldsymbol{M_{ij}}$ is calculated as follows [25]:

$$M_{ij} = \sum_{k=1}^{7} D_{ik} \cdot T_{kj}$$

Where

D_{ik} represents the element in the *i*-th row and *k*-th column of matrix *D*
T_{kj}represents the element in the *k*-th row and *j*-th column of matrix *T*.

Matrix multiplication transforms the data to new feature space. Since the token matrix is unique for each user, the matrix multiplication process customizes the data representation accordingly. The figure(2) illustrates a comparison between the normalized data and the results after feature transformation is applied:

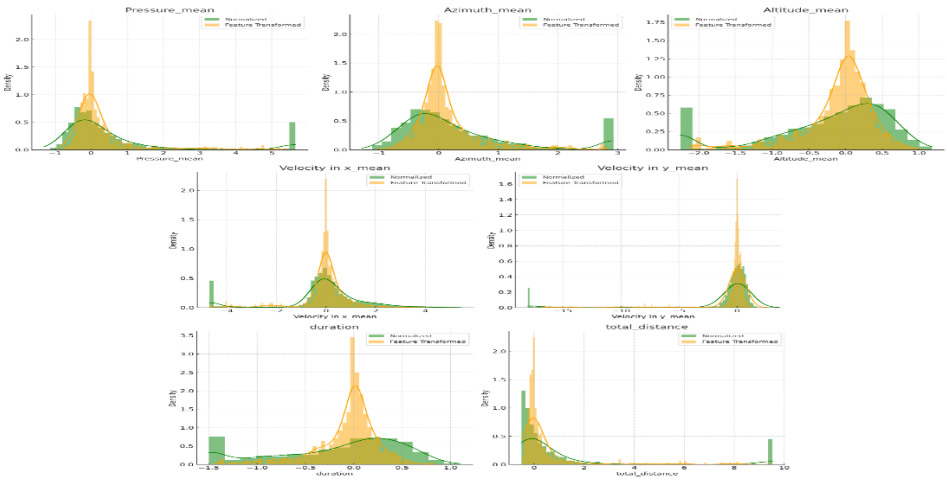

Figure 2: Visual comparison between the normalized data and data obtained after feature transformation

Power System Technology
ISSN:1000-3673
Received: 16-01-2024 Revised: 12-02-2024 Accepted: 07-03-2024

Both datasets, normalized and feature transformed, exhibit changes in their distributions due to the feature transformation process, with the feature transformed data showing different spread and density characteristics compared to the normalized data.Figure (3) below exhibit the PCA visualizations between the normalized data and the feature-transformed data to show the dimensionality reduction.

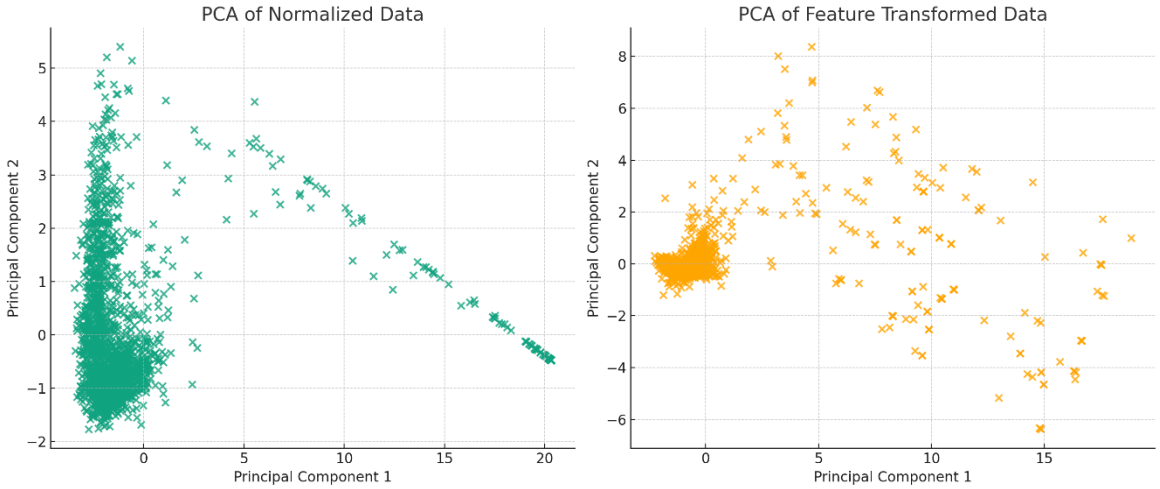

Figure 3: PCA visualizations between the normalized data and the feature-transformed data to show the dimensionality reduction.

The PCA of the normalized data displays a certain spread and clustering of data points. This distribution reflects the variability and structure in the data post-normalization, indicating how features interact and contribute to the variance. While the PCA plot for the feature-transformed data shows a distinct pattern of distribution compared to the normalized data. The spread and clustering of points may differ, suggesting that the feature transformation process has altered the data's underlying structure and the relationships among features.

3.2.4 Radial Basis Function (RBF) Kernel

The RBF kernel transforms the input feature space into a higher-dimensional space, potentially making it easier to linearly separate data points that are not linearly separable in the original space [26]. This gives one more security layer to our data.

To apply the RBF kernel to a dataset, we computed the kernel matrix K, where each element K_{ij} represents the RBF kernel between the i^{th} and j^{th} samples in the dataset. This matrix encapsulates the similarity between all pairs of samples based on the RBF kernel [27]. RBF kernel matrix is calculated as:

(i) Calculating the pairwise Euclidean distances between all samples in the dataset.
(ii) Applying the RBF kernel formula using the chosen bandwidth parameter ($\boldsymbol{\sigma}$) to transform these distances into similarity measures.

Power System Technology
ISSN:1000-3673
Received: 16-01-2024 Revised: 12-02-2024 Accepted: 07-03-2024

(iii) Assembling these similarity measures into the kernel matrix ***K***, which then serves as a transformed feature set for further analysis or machine learning tasks.

The choice of $\boldsymbol{\sigma}$ significantly influences the transformation's effectiveness. We used heuristic approach [28] to determine value of $\boldsymbol{\sigma}$, Heuristic approach uses the median of the pairwise distances among all samples. We computed the pairwise Euclidean distances among all samples using the **pdist** function from **SciPy**, which efficiently generates a condensed matrix of distances. The median of these pairwise distances was chosen as the value for $\boldsymbol{\sigma}$.Table (4) displays thevalues obtained after applying the pdlist function on the dataset:

Table 4: Value of bandwidth parameter (**σ)**

Minimum Pairwise Distance	0.0
Maximum Pairwise Distance	21.44025640013489
Mean Pairwise Distance	3.038118059398096
Median Pairwise Distance (σ)	0.9407331288673279

As we have 1760 signature data of all users. RBF kernel will have matrix of (1760×1760) dimension.

The Radial Basis Function (RBF) kernel plays an important role during the data preparation phase in BioHashing, because it is applied to the feature space of the biohashing method and map the data characteristics to a form which can further help to secure and efficient hashing. The RBF kernel applies the nonlinear transformation to the biohashed data in the mapping process from the original feature space to a space with more dimensions [29]. It improved separability, that's an important requirement for the BioHashing process [30]. The RBF kernel can effectively reduce the noise and outliers in the dataset by smoothing it [31]. The transformed data is robust toward small changes or perturbations, which helps to transform the data into a more stable and consistent state. The RBF kernel can effectively map the data into the high-dimensional space, thus, increasing the dimension of the dataset [32]. This extension is particularly useful for BioHashing, it allows for more difference between the data points. RBF kernel [33] amplifies the intrinsic data characteristics and hence BioHashing algorithm produces more distinguishable hashes. Structured data makes the hashes more secure and accurate.Figure (4) shows a plot Heatmap to visualize the RBF Kernel matrix:

Power System Technology
ISSN:1000-3673
Received: 16-01-2024 Revised: 12-02-2024 Accepted: 07-03-2024

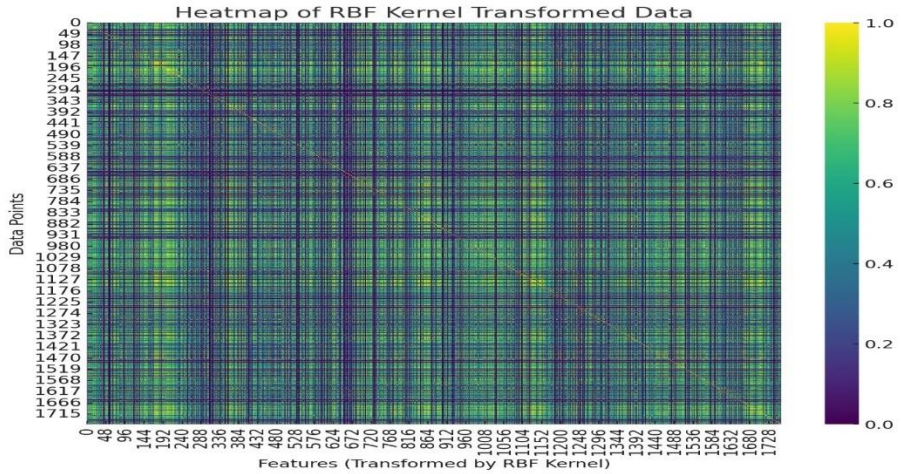

Figure 4: RBF Kernel matrix (1760 × 1760)

The diagonal is bright because the similarity of each point with itself is the highest (value of 1), as expected in a similarity matrix. There's noticeable variability in the similarity scores between different points, with some off-diagonal areas showing higher similarity (warmer colors) and others much lower (cooler colors).

3.2.5 PCA Kernel

Kernel PCA is a powerful tool for dimensionality reduction, particularly suitable for complex datasets where linear methods fall short. This is usually applied to the situations when RBF kernel is not computationally feasible. In applications like signature verification, where capturing the nuances of the data is crucial, Kernel PCA can offer significant benefits in terms of feature extraction, model performance, and computational efficiency [34].

Kernel PCA is good for capturing the nonlinear relationships of features in datasets because it maps data into some higher-dimensional space in a nonlinear way, so that linear PCA can be applied. This thereby enables it to reduce the dimension of the data while still retaining the structure that would have been missed by linear PCA [35]. After applying RBF kernel, Kernel PCA extracts features that are more representative of the structure in the data, and thus, it is helpful for the tasks of classification, clustering, or visualization in which linear separability is not found in the original space [36].

Verification of such signatures is often carried out over complex or high-dimensional data, and for its noise reduction and improvement in accuracy, the role of Kernel PCA is quite instrumental and it enhances efficiency [37] [38]. After computation of the RBF kernel matrix (K), centering the kernel matrix (K') has to be done [39][40]. The centered kernel matrix K'undergoes eigendecomposition:

$$K'v_i = \lambda_i v_i$$

where$v_i$are the eigenvectors, and λ_i are the corresponding eigenvalues. The eigenvectors correspond to the principal components in the feature space, and the eigenvalues indicate the variance captured by these components [41]. Original data points are projected onto the

Power System Technology
ISSN:1000-3673
Received: 16-01-2024 Revised: 12-02-2024 Accepted: 07-03-2024

reduced-dimensional space using the selected eigenvectors. This projection is effectively the dot product between the original kernel matrix (before centering) and the eigenvectors associated with the largest eigenvalues [41].Data points are projected onto the k principal components by computing a new representation:

$$Z_i = [v_1(i), v_2(i), \ldots, v_k(i)]^T$$

where $v_j(i)$ is the i-th component of the j-th eigenvector, and k is the number of components (dimensions). These projections are the coordinates of the original data in the reduced-dimensional space formed by the principal components.

Figure (5) shows the dimensionality of RBF Kernel and Kernel PCA. Since the dataset of RBF kernel is highly dimensioned and it was not possible to vies it visually. We applied PCA to the RBF kernel-transformed dataset to reduce to two principal components so that clearer view of its structure.

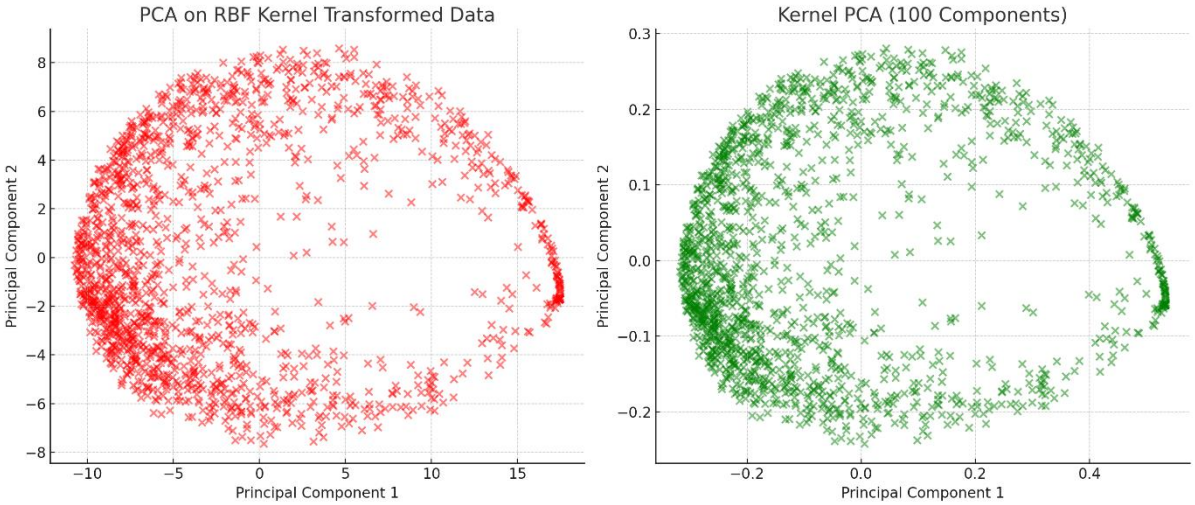

Figure 5: dimensionality of RBF Kernel and Kernel PCA

We also explored several metrics to quantify aspects such as cluster cohesion, separation, and overall structure on the RBF Kernel and Kernel PCA datasets. We computed Silhouette Score [42], Davies-Bouldin Index [43] and Calinski-Harabasz Index [44] and yielded the following results as shown in the table (5).

Table 5: Metrics to quantify overall structure of the RBF Kernel and Kernel PCA datasets

	RBF Kernel-Transformed Dataset	**Kernel PCA Dataset**
Silhouette Score	0.381	0.378
Davies-Bouldin Index	0.944	0.962
Calinski-Harabasz Index	1972.82	1759.78

Power System Technology
ISSN:1000-3673
Received: 16-01-2024 Revised: 12-02-2024 Accepted: 07-03-2024

The silhouette scores, Davies-Bouldin Indexes, and Calinski-Harabasz Indexes for both datasets show that the RBF kernel dataset marginally outperforms the Kernel PCA dataset in terms of cluster cohesion, separation, and definition. While both datasets exhibit good cluster quality, the RBF kernel's slightly better scores suggest it facilitates a more distinct grouping of data points, indicating its effectiveness in clustering transformations.

3.3 Signature Verification with BiLSTM

Bidirectional Long Short-Term Memory (BiLSTM) is an improvement on traditional LSTM, a type of RNN (Recurrent Neural Network) designed to capture long-term dependencies in sequence data. By processing data in both forward and backward directions, BiLSTM effectively enhances the information available to the model, leading to enhanced performance on various sequence-related tasks like natural language processing and time series analysis[45].Figure (6) shows the architectural view of biLSTM.

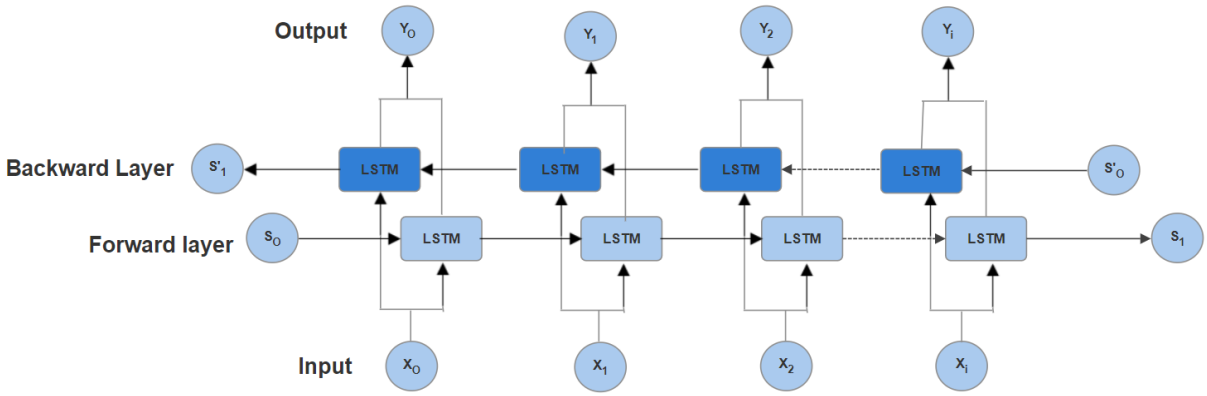

Figure 6: Architectural view of biLSTM

Using BiLSTM (Bidirectional Long Short-Term Memory) networks for signature verification, especially with features involves several critical steps [46]. Each feature plays a unique role in capturing the dynamic characteristics of a signature, which are crucial for distinguishing between genuine and forged signatures. Below are the steps for employing biLSTM for signature verification:

3.3.1 Data splitting

We utilized the biohashed dataset for the signature verification. Data is split in two different ways: by users and by signatures. In each of these cases, 80% of the data is used for training and the remaining 10% for both validation and testing. This makes sure the model is trained and tested over a disjoint set of individuals, thus reducing the chance of data leakage and overfitting. It is an important step to look how well our proposed model will generalize to new, unseen signatures.Table (6) and table (7) shows the splitting out dataset based on the users and signatures respectively.

Power System Technology
ISSN:1000-3673
Received: 16-01-2024 Revised: 12-02-2024 Accepted: 07-03-2024

Table6: Dataset splitting based on the user

Splitters	No. of Users (Total Users: 45)	Percentage	Users
Training set	36	80%	User12, User15, User35, User39, User28, User25, User21, User7, User23, User18, User24, User38, User34, User1, User37, User8, User14, User2, User4, User10, User5, User3, User11, User40, User31, User42, User19, User30, User27, User9, User29, User16, User22, User36, User45
Testing set	05	10%	User20, User32, User6, User44, User17
Validation set	04	10%	User33, User41, User43, User13

Table 7: Dataset splitting based on the Signatures

Splitters	No. of Signatures (Total Users: 1760)	Percentage
Training set	1408	80%
Testing set	176	10%
Validation set	176	10%

3.3.2 Data Preprocessing for Model Input

Normalization: Features are normalized to ensure that all input variables contribute equally to the model training process. We use min-max normalization that scalesw the features to have a mean of 0 and a standard deviation of 1 [48]. This step is important for models like neural networks, which are sensitive to the scale of input data.

3.3.3 Model Design and Compilation

Architecture Design: Our BiLSTM model used total 8 layers as given below. Dropout layers is used for regularization to prevent overfitting while the LSTM layers can capture the temporal dependencies in the signature data, making it suitable for the task [49]**.**

- **Input Layer:** It is defined by the shape of the input data to the first Bidirectional LSTM layer.
- **First Bidirectional LSTM Layer:** It internally consists of two LSTM layers processing the data in both forward and backward directions. It has 64 units.
- **Dropout Layer:** It follows the first Bidirectional LSTM layer to prevent overfitting. We used dropout layer with a rate of 0.5 to prevent overfitting.

Power System Technology
ISSN:1000-3673
Received: 16-01-2024 Revised: 12-02-2024 Accepted: 07-03-2024

- **Second LSTM Layer:** It is the standard (unidirectional) LSTM layer that processes the sequence data further. It has 32 units.
- **Dropout Layer:** Another dropout layer follows the second LSTM layer for additional regularization.
- **Dense Layer:** A fully connected layer with ReLU activation function for further processing the learned features. The Rectified Linear Unit (ReLU) is an activation function used for various tasks, including classification, regression, and feature extraction.
- **Dropout Layer:** An additional dropout layer after the dense layer to reduce the risk of overfitting.
- **Output Layer:** The final dense layer with a single unit and a sigmoid activation function for binary classification (genuine vs. forged) [50].

3.3.4 Model Training

- **Callbacks:** ModelCheckpoint and EarlyStopping callbacks are used to save the best model based on validation loss and to stop training if the model doesn't improve, preventing overfitting. Model is trained on the preprocessed training data using the batch size 64 and 30 epochs.
- **Training:** The model is trained using the training set, with the validation set used to monitor performance and adjust hyperparameters like learning rate if necessary.
- **Hyperparameter**
  The hyperparameters are crucial for defining the model's architecture and training behavior. Adjusting the values of hyperparameters can significantly impact the model's ability to learn from the training data and generalize to unseen data [51]. The table (8) exhibit the hyperparameters that are utilized in our model, along with their corresponding values.

Table 8: Hyperparameters used in biLSTM

Units in the First BiLSTM Layer	64 units
Units in the Second LSTM Layer	32 units
Dropout Rate	0.5
Units in the First Dense Layer	32 units, with ReLU (Rectified Linear Unit) activation
Units in the Output Layer	1 unit, with a sigmoid activation function
Optimizer	Adam with a learning rate of 0.001
Loss Function	Binary crossentropy
Batch Size	64
Epochs	30
Callbacks	ModelCheckpoint, EarlyStopping(patience=5)

Power System Technology
ISSN:1000-3673

Received: 16-01-2024 Revised: 12-02-2024 Accepted: 07-03-2024

3.3.5 Evaluation and Analysis

Table (9) shows the performance of our model with other models proposed by various researchers.

Table9**:** Performance Metrics of the models

Method	Dataset	Accuracy	Precision	Recall	F1 score	FAR	FRR	AUC	ERR
AnAVN Model for Handwritten Signature Verification [7]	CEDAR (English)	96.16%				3.26 %	4.42 %		3.77 %
Single-template matching + LG-DTW with DMPs [9]	SVC2004 Task2								1.78 %
Single-template strategy using a mean template and a weighting scheme [10]	SVC2004 Task2								2.98 %
Online Signature Verification using Deep Descriptors [11]	SVC 2004	90.65%				15.43 %	3.26 %	0.9689	5.22 %
GNU kernelized biohashing biLSTM signature verification [Proposed] Splitting by Users	SVC2004 Task2	95.50%	96.42 %	98%	97.20 %	7%	2%	0.96	7%
GNU kernelized biohashing biLSTM signature verification	SVC2004 Task2	71.02%	67.50 %	78.49 %	72.58 %	36.11 %	21.51 %	0.81	29%

Power System Technology
ISSN:1000-3673

Received: 16-01-2024 Revised: 12-02-2024 Accepted: 07-03-2024

[Proposed] Splitting by Signatures									

Figures (7) and (8) display the ROC curve and ERR rate, respectively, for the scenarios where the dataset is divided based on users and their signatures.

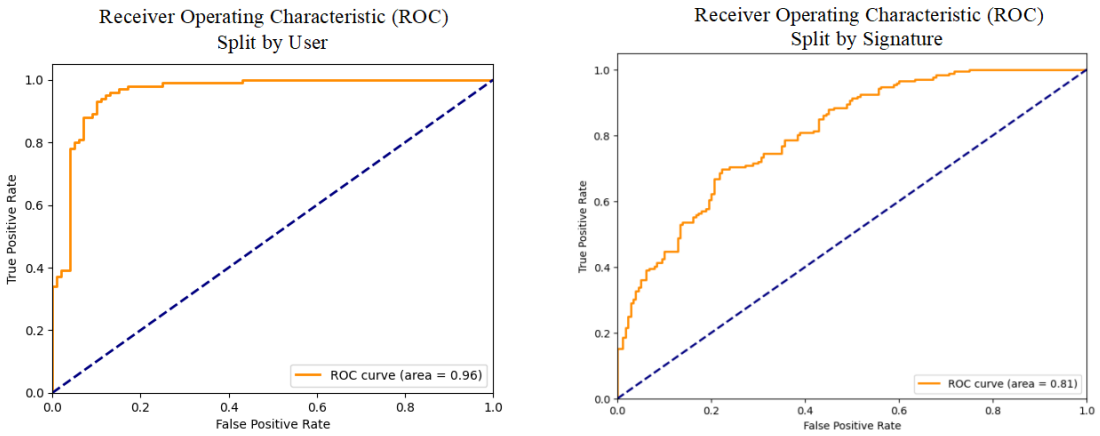

Figure 7: Receiver Operating Characteristic (ROC) curve, split by users and split by Signatures

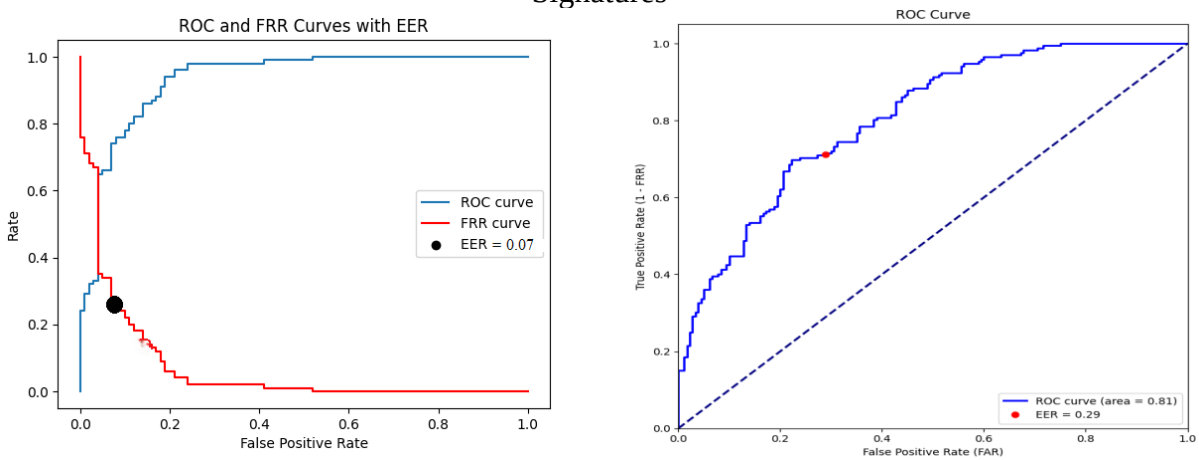

Figure 8: Equal Error Rate (EER), split by users and split by Signatures

Conclusion

The BiLSTM model demonstrated high accuracy (95.5%), precision (96.4%), recall (98%), and an F1 score (95.76%) when split by users, indicating its effectiveness in distinguishing between genuine and forged signatures. The balance between model complexity and the ability to generalize to unseen signatures was effectively managed through the use of dropout layers and careful hyperparameter tuning. The study highlighted the significance of dynamic features in signature verification, underscoring the need for comprehensive feature extraction and selection techniques. With a False Acceptance Rate (FAR) of 7% and a False Rejection

Power System Technology
ISSN:1000-3673
Received: 16-01-2024 Revised: 12-02-2024 Accepted: 07-03-2024

Rate (FRR) of 2%, the model showed promise for practical security applications, though there's room for improvement in reducing FAR.

References:

1. E. Brophy et al., "Generative Adversarial Networks in Time Series: A Systematic Literature Review," in ACM Computing Surveys, vol. 55, no. 10, Article 199, February 2023.
2. M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," arXiv:1701.07875v3 [stat.ML], Dec. 2017.
3. D. Garcia Torres, "Generation of Synthetic Data with Generative Adversarial Networks," Degree Project in Computer Science, KTH Royal Institute of Technology, Stockholm, Sweden, 2018.
4. Huang, Y.-B., Chen, D.-H., Hua, B.-R., & Zhang, Q.-Y. (2023). A high-performance speech BioHashing retrieval algorithm based on audio segmentation. *Journal of Biometric Security and Data Protection*, 11(2), 157-173. https://doi.org/10.1016/j.csl.2023.101551
5. Otroshi-Shahreza, H., Melzi, P., Osorio Roig, D., Rathgeb, C., Busch, C., Marcel, S., Tolosana, R., & Vera-Rodríguez, R. (2023). Benchmarking of Cancelable Biometrics for Deep Templates. ArXiv.
6. M. Kumar and N. Manisha, "Cancelable Biometrics: A Comprehensive Survey," ArtifIntell Rev, vol. 53, pp. 3403–3446, 2020. https://doi.org/10.1007/s10462-019-09767-8
7. H. Li, P. Wei and P. Hu, "AVN: An Adversarial Variation Network Model for Handwritten Signature Verification," in *IEEE Transactions on Multimedia*, vol. 24, pp. 594-608, 2022, doi: 10.1109/TMM.2021.3056217
8. N. Sharma, S. Gupta, H. G. Mohamed, D. Anand, J. L. V. Mazón, D. Gupta, and N. Goyal, "Siamese Convolutional Neural Network-Based Twin Structure Model for Independent Offline Signature Verification," Sustainability, vol. 14, no. 11484, 2022. https://doi.org/10.3390/su141811484
9. M. Okawa, "Modified Dynamic Time Warping with Local and Global Weighting for Online Signature Verification," *2021 IEEE 3rd Global Conference on Life Sciences and Technologies (LifeTech)*, Nara, Japan, 2021, pp. 124-125, doi: 10.1109/LifeTech52111.2021.9391879.
10. M. Okawa, "Online signature verification using single-template matching with time-series averaging and gradient boosting," in *Pattern Recognition*, vol. 2020, no. 107227, Jan. 2020. doi: 10.1016/j.patcog.2020.107227.

Power System Technology
ISSN:1000-3673

Received: 16-01-2024 Revised: 12-02-2024 Accepted: 07-03-2024

11. Singh and S. Viriri, "Online Signature Verification using Deep Descriptors," *2020 Conference on Information Communications Technology and Society (ICTAS)*, Durban, South Africa, 2020, pp. 1-6, doi: 10.1109/ICTAS47918.2020.233999.
12. D. Y. Yeung et al., "SVC2004: First International Signature Verification Competition," in Biometric Authentication. ICBA 2004. Lecture Notes in Computer Science, vol. 3072, D. Zhang and A. K. Jain, Eds. Berlin, Heidelberg: Springer, 2004. doi: 10.1007/978-3-540-25948-0_3
13. Goodfellow et al., "Generative adversarial networks," Communications of the ACM, vol. 63, pp. 139 - 144, 2014, https://doi.org/10.1145/3422622.
14. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," CoRR, abs/1511.06434, 2015.
15. X. Mao et al., "Least Squares Generative Adversarial Networks," in 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2813-2821.
16. Y. Li and Z. Dou, "When can Wasserstein GANs minimize Wasserstein Distance?" ArXiv, abs/2003.04033, 2020.
17. J. Chang et al., "Distributed Radar Target Detection Based on RF-SSA in Non-Gaussian Noise," Electronics, 2022, https://doi.org/10.3390/electronics11152319
18. H. Rangwani et al., "Class Balancing GAN with a Classifier in the Loop," Conference on Uncertainty in Artificial Intelligence, pp. 1618-1627, 2021. Access here
19. A. Rios and L. Itti, "Closed-Loop Memory GAN for Continual Learning," International Joint Conference on Artificial Intelligence, pp. 3332-3338, 2018, https://doi.org/10.24963/ijcai.2019/462
20. E. Nazari et al., "AutoGAN: An Automated Human-Out-of-the-Loop Approach for Training Generative Adversarial Networks," Mathematics, 2023, https://doi.org/10.3390/math11040977
21. Y. Huang, B. Wang, X. Pu, Y. Li, and Q. Zhang, "Research on ciphertext speech biohashing authentication based on chaotic system and improved public chain," Journal of Supercomputing, 2023. https://doi.org/10.1007/s11227-023-05693-3
22. H. Qian, Q. Wen, L. Sun, J. Gu, Q. Niu and Z. Tang, "RobustScaler: QoS-Aware Autoscaling for Complex Workloads," 2022 IEEE 38th International Conference on Data Engineering (ICDE), Kuala Lumpur, Malaysia, 2022, pp. 2762-2775, doi: 10.1109/ICDE53745.2022.00252.
23. C. Karabat and B. Topcu, "How to assess privacy preservation capability of biohashingmethods?: Privacy metrics," 2014 22nd Signal Processing and Communications Applications Conference (SIU), Trabzon, Turkey, 2014, pp. 2217-2220, doi: 10.1109/SIU.2014.6830705.
24. H. O. Shahreza, C. Rathgeb, D. Osorio-Roig, V. K. Hahn, S. Marcel and C. Busch, "Hybrid Protection of Biometric Templates by Combining Homomorphic Encryption and

Power System Technology
ISSN:1000-3673

Received: 16-01-2024 Revised: 12-02-2024 Accepted: 07-03-2024

Cancelable Biometrics," 2022 IEEE International Joint Conference on Biometrics (IJCB), Abu Dhabi, United Arab Emirates, 2022, pp. 1-10, doi: 10.1109/IJCB54206.2022.10007960.

25. K. Rinki, P. Verma, and R. Singh, "A novel matrix multiplication based LSB substitution mechanism for data security and authentication," Journal of King Saud University: Computer and Information Sciences, vol. 34, 2021, https://doi.org/10.1016/j.jksuci.2021.01.013
26. A. Morán, L. Parrilla, M. Roca, J. Font-Rossello, E. Isern and V. Canals, "Digital Implementation of Radial Basis Function Neural Networks Based on Stochastic Computing," in IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 13, no. 1, pp. 257-269, March 2023, doi: 10.1109/JETCAS.2022.3231708
27. Y. Liu and K. K. Parhi, "Computing RBF Kernel for SVM Classification Using Stochastic Logic," 2016 IEEE International Workshop on Signal Processing Systems (SiPS), Dallas, TX, USA, 2016, pp. 327-332, doi: 10.1109/SiPS.2016.64.
28. W. Zhou, H. Tao, F. Wang, and W. Pan, "The Optimal Bandwidth Parameter Selection in GPH Estimation," Journal of Mathematics, vol. 2021, Article ID 2876000, 2021, https://doi.org/10.1155/2021/2876000
29. Y. Li, Z. Wang, and H. Dai, "Improved Parkinsonian tremor quantification based on automatic label modification and SVM with RBF kernel," Physiological Measurement, vol. 44, 2023. DOI 10.1088/1361-6579/acb8fe
30. A. Gopi, R. N. Sravana Jyothi, V. L. Narayana, and K. S. Sandeep, "Classification of tweets data based on polarity using improved RBF kernel of SVM," International Journal of Information Technology, vol. 15, pp. 965-980, 2020, https://doi.org/10.1007/s41870-019-00409-4
31. F. He, M. He, L. Shi, X. Huang, and J. Suykens, "Enhancing Kernel Flexibility via Learning Asymmetric Locally-Adaptive Kernels," ArXiv, abs/2310.05236, 2023.
32. S. Atif, S. Khan, I. Naseem, R. Togneri, and Bennamoun, "Multi-Kernel Fusion for RBF Neural Networks," Neural Processing Letters, vol. 55, pp. 1045-1069, 2020, https://doi.org/10.1007/s11063-022-10925-3
33. F. He, M.-q. He, L. Shi, X. Huang, and J. Suykens, "Enhancing Kernel Flexibility via Learning Asymmetric Locally-Adaptive Kernels," arXiv.org, vol. abs/2310.05236, 2023.
34. Srinivas Mekala and B. Padmaja Rani, Kernel PCA Based Dimensionality Reduction Techniques for Preprocessing of Telugu Text Documents for Cluster Analysis, International Journal of Advanced Research in Engineering and Technology, 11 (11), 2020, pp. 1337-1352 doi: 10.34218/IJARET.11.11.2020.121 applications.
35. A. Rehman, A. Khan, M. A. Ali, M. U. Khan, S. U. Khan and L. Ali, "Performance Analysis of PCA, Sparse PCA, Kernel PCA and Incremental PCA Algorithms for Heart Failure Prediction," 2020 International Conference on Electrical, Communication, and

Power System Technology
ISSN:1000-3673
Received: 16-01-2024 Revised: 12-02-2024 Accepted: 07-03-2024

Computer Engineering (ICECCE), Istanbul, Turkey, 2020, pp. 1-5, doi: 10.1109/ICECCE49384.2020.9179199.
36. S. Suhas and Dr. C. R. Venugopal, "Feature Extraction and Classification of MRI Using Hybrid RBF Kernel and SVM," International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 2021.
37. S. Auddya, R. K. Singh and S. Sundaram, "Online Signature Verification using Time Warp Edit Distance based kernel," 2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR), Dortmund, Germany, 2020, pp. 319-324, doi: 10.1109/ICFHR2020.2020.00065.
38. G. Kumar, "Improving Digital Signature Verification Accuracy Through Support Vector Machine Learning: A Comparative Study," International Journal of Advance Scientific Research, Volume 03 Issue 05 pp.75-79, 2023, doi: https://doi.org/10.37547/ijasr-03-05-
39. N. Tsapanos et al., "Fast Kernel Matrix Computation for Big Data Clustering," International Conference on Conceptual Structures, pp. 2445-2452, 2015, https://doi.org/10.1016/j.procs.2015.05.352
40. X. Liu, X. Zhang, J. Xiong, F. Gu and J. Wei, "An Enhanced Iterative Clipping and Filtering Method Using Time-Domain Kernel Matrix for PAPR Reduction in OFDM Systems," in IEEE Access, vol. 7, pp. 59466-59476, 2019, doi: 10.1109/ACCESS.2019.2915354.
41. F. Hallgren and P. Northrop, "Incremental kernel PCA and the Nyström method," arXiv.org, abs/1802.00043, 2018.
42. G. Ogbuabor and N. UgwokeF., "Clustering Algorithm for a Healthcare Dataset Using Silhouette Score Value," International Journal of Computer Science and Information Technology, vol. 10, pp. 27-37, 2018.
43. B. J. D. Sitompul, O. S. Sitompul, and P. Sihombing, "Enhancement Clustering Evaluation Result of Davies-Bouldin Index with Determining Initial Centroid of K-Means Algorithm," Journal of Physics: Conference Series, vol. 1235, 2019, DOI 10.1088/1742-6596/1235/1/012015
44. S. P. Lima and M. D. Cruz, "A genetic algorithm using Calinski-Harabasz index for automatic clustering problem," Revista Brasileira de ComputaçãoAplicada, 2020.
45. P. Zhou et al., "Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification," in Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2016.
46. T. Longjam, D. R. Kisku, and P. Gupta, "Writer independent handwritten signature verification on multi-scripted signatures using hybrid CNN-BiLSTM: A novel approach," Expert Systems with Applications, vol. 214, 2022, Art. no. 119111.
47. H. Saikia, K. Sarma, "Offline Signature Verification Using Radial Basis Function with Selected Feature Sets," Advances in Communication, Devices and Networking. Lecture

Power System Technology
ISSN:1000-3673

Received: 16-01-2024 Revised: 12-02-2024 Accepted: 07-03-2024

Notes in Electrical Engineering, vol 462. Springer, Singapore. https://doi.org/10.1007/978-981-10-7901-6_62, 2018.

48. Henderi, T. Wahyuningsih, and E. Rahwanto, "Comparison of Min-Max normalization and Z-Score Normalization in the K-nearest neighbor (kNN) Algorithm to Test the Accuracy of Types of Breast Cancer," International Journal of Informatics and Information Systems, 4(1), 13-20, 2021, doi:https://doi.org/10.47738/ijiis.v4i1.73
49. K. Nandhini and G. T. Pavai, "An Optimal Stacked ResNet-BiLSTM-Based Accurate Detection and Classification of Genetic Disorders," Neural Process Lett 55, 9117–9138, 2023, https://doi.org/10.1007/s11063-023-11195-3
50. J. Park, J. -H. Kim and Y. Jung, "Binary Classification Fault Diagnosis for Octocopter Using Deep Neural Network," 2021 29th Mediterranean Conference on Control and Automation (MED), PUGLIA, Italy, 2021, pp. 121-125, doi: 10.1109/MED51440.2021.9480214.
51. B. Singh, J. Singh and G. Singh, "Enhancing Facial Emotion Detection with CNN: Exploring the Impact of Hyperparameters," 2023 4th IEEE Global Conference for Advancement in Technology (GCAT), Bangalore, India, 2023, pp. 1-5, doi: 10.1109/GCAT59970.2023.10353363.