A Comparative Performance of Optimizers and Tuning of Neural Networks for Spoof Detection Framework

Ankita Chadha
School of Computer Science,
Taylors University, Subang Jaya,
Selangor, Malaysia 47500

Azween Abdullah
School of Computer Science,
Taylors University, Subang Jaya,
Selangor, Malaysia 47500

Lorita Angeline
School of Computer Science,
Taylors University, Subang Jaya,
Selangor, Malaysia 47500

***Abstract*—The breakthroughs in securing speaker verification systems have been challenging and yet are explored by many researchers over the past five years. The compromise in security of these systems is due to naturally sounding synthetic speech and handiness of the recording devices. For developing a spoof detection system, the back-end classifier plays an integral role in differentiating spoofed speech from genuine speech. This work conducts the experimental analysis and comparison of up-to-date optimization techniques for a modified form of Convolutional Neural Network (CNN) architecture which is Light CNN (LCNN). The network is standardized by exploring various optimizers such as Adaptive moment estimation, and other adaptive algorithms, Root Mean Square propagation and Stochastic Gradient Descent (SGD) algorithms for spoof detection task. Furthermore, the activation functions and learning rates are also tested to investigate the hyperparameter configuration for faster convergence and improving the training accuracy. The counter measure systems are trained and validated on ASV spoof 2019 dataset with Logical (LA) and Physical Access (PA) attack data. The experimental results show optimizers perform better for LA attack in contrast to PA attack. Additionally, the lowest Equal Error Rate (EER) of 9.07 is obtained for softmax activation with SGD with momentum wrt LA attack and 9.951 for SGD with nestrov wrt PA attack.**

***Keywords*—Spoof detection; speech synthesis; voice conversion; convolutional neural networks; optimizers; gradient descent algorithm; spoofed speech; automatic speaker verification**

I. Introduction

The uniqueness of voice makes it a popular choice as a biometric for securing smart-phones, telephonic-verification for banking, online-shopping and interestingly, voice-based logins. This also requires the voice biometric based systems to be resilient to unauthentic access in the form of spoofing attacks. The Automatic Speaker Verification (ASV) systems are thriving to make the voice-based applications secured through countermeasures or spoof detection algorithms. These spoof detection systems are embedded in the ASV pipeline as standalone or as a sub-part of the security stream. Apparently, the spoofing attacks may be categorized as synthetic or replay attacks. The synthetic speech (SS) is generated by means of Voice Conversion (VC) or Text-to-Speech (TTS) generators. The replay speech is acquired through careful filtering of the target speech through a recording device. Apart from this category, the spoofing attacks may be direct or Physical Access (PA) attacks and indirect or Logical Access (LA) attacks. The former requires physical space i.e., microphone while the latter is conducted by direct injection of speech, exempting the sensor involvement. Lately, there is rise in spoofing attacks on the ASV systems due to impeccable quality of synthetic speech generators and cheaper high-end recording devices. Yet, these attacks are not preventable, but they can be detected by building countermeasures ensuring the safety of the ASV framework.

II. Related Literature

The robustness of a spoof detection system depends on its internal building blocks which includes the feature extraction and classification. During the training mode, the input speech is processed to reduce redundancy in data and filter out the required information from the speech, that is, naturality and speaker specific content. These unique features are then trained using appropriate machine learning algorithm to get a training rule or trained model. Consequently, in the testing mode the appropriate features are extracted from the test samples and fed to the trained model following which the samples are categorized as genuine or spoofed. For building a model, the number of samples and types of attacks must be considered in a dataset as they will help boast the validation accuracy. The datasets available for PA attacks are Red Dots [1], VoicePA [2] and ASV spoof 2017 [3] while for LA attack, SAS [4] and ASV spoof 2015 are popular . The ASV spoof 2019 [5], [6] has all three kind of attacks samples including synthetic and replay speech. Hence, it is chosen over all other datasets for developing an anti-spoofing algorithm.

With regards to capturing human and synthetic traits from the speech, the glottal excitation and source-filer parameters have been extracted along with prosodic features. The Linear Prediction Co-efficient (LPC) [7], Linear Prediction-Residual parameters [8] and Line Frequency Cepstral Co-efficient (LFCC) [9] based spectral features are found to represent speech quite profoundly. Additionally, the perceptual parameters have also been explored as they have similarity with human perceptual filter bank. The Mel Frequency Cepstral Co-efficient (MFCC), Constant Q-factor Cepstral Co-efficient (CQCC) [10] and Cochlear Filter Cepstral Coefficients with Instantaneous Frequency (CFCCIF) [9] are successfully tested perceptual features for LA attack. The Constant Q transform (CQT) unlike the standard Fourier Transform has irregular frequency bins that allows it to maintain a constant Q-factor

throughout the spectrum [11]. This promotes evident discrimination within the spectrum as spoofing related characteristics are revealed distinctly. Hence, considering the efficiency of CQT parameters, we are using these as the feature extraction technique.

In addition, the classification techniques have also caught attention of researchers for building a robust spoof detection system. Usually, the Gaussian Mixture Models (GMM) and Universal Background Models (UBM) are employed as they perform well with almost all the feature extraction techniques [8]. Yet, the main shortcoming of this arrangement is that the GMM and UBM are trained independently making them unaware of each other's learning rule. To overcome this shortcoming, the GMM and Support Vector Machines (SVM) were explored as they were more versatile and performed better than the GMM-UBM duo [12]. Additionally, the GMM classifier is limited to perform better with features of low dimensionality [13]. Furthermore, the Deep Learning models have been investigated and perform comparatively well in contrast to the shallow models [14]. Of course, they require a lot of labelled data for the training but also fail to capture temporal information simultaneously. This lead to exploring other deep models like Convolutional Neural Networks (CNN) [15], Recurrent Neural Networks (RNN) [16], Long Short-Term Memory (LSTM) [17], Gated Recurrent Units (GRU)[15] and Generated Adversarial Networks (GAN) [10]. In [16], [18], CNN-RNN are combined to explore effectiveness of both models individually. Thus, the CNNs are used when spatial learning is of importance while in case of temporal learning, RNNs and its variants are used. Moreover, the CNNs are powerful in handling large amount data at the cost of high training time and large number of parameters. To overcome this, the Light-CNN (LCNN) architecture is introduced to avoid repetition in the parameters and ultimately improve training resources [19]. The LCNN based fusion architecture achieved best results in the ASV spoof 2017 challenge with lower EER for replay attack [10]. Then onwards, LCNN gained popularity and has been tried for synthetic speech detection as well [20]. Following which, in the ASV spoof 2019 challenge, an improved version of this architecture with angular-softmax was presented for both LA and PA attack [21]. The major highlight of the LCNN is its ability to achieve generality for variation in data distribution such as recording conditions, noisy speech, speaker variations, etc [19]. This is possible by endowing in optimization of these networks. This leads to wider range of applications of the network along with impeccable theoretical proofs. Even though the optimization algorithms have been existing for more than two decades, through continuous refinement for highly complex networks with large data size, a defined reassessment of these state-of-the-art optimizers is the need of the day. Inspite of the popularity of LCNNs, according to author's knowledge, there is no significant work found in optimizing the LCNN for spoof detection task. Moreover, tuning of the hyperparameters improves the performance of the network with faster convergence. This work also dedicates its attention to various activation functions as they hold a crucial role in deciding the category of the unknown test sample which ultimately contributes to lower model loss and increase performance accuracy. Also, a precise choice of activation might prove to enhance the training time by making the network learn complex patterns easily. Different combinations of learning rates, activations and optimizers have been investigated in this work to determine most suitable model parameters for spoof detection task. Thus, the objectives of this work can be summarized as following:

i An extensive comparison of various optimizers is performed using ASVspoof 2019 dataset for LA and PA attacks. The common optimizers compared include Adaptive moment estimation (Adam), Adaptive-gradient (Adagrad), Adaptive-delta (Adadelta), Nesterov Accelerated Adam (Nadam), Root Mean Square Propagation (RMSprop), Stochastic Gradient Descent (SGD), SGD with nesterov accelerated gradient (NAG) and momentum. This unravels certain unexpected results as against the usual classification problem where the RMS prop performs equally well with the gradients and delta versions of Adam.

ii Exploring activation functions popularly used in training the transfer function are compared with variations in optimizers to suit the spoof detection application for the LCNN framework. These activations include Softmax, Rectified Linear Unit (ReLU) and Logistic function.

iii The experimental results are compared with state-of-the-art softmax-Adam optimizer and evaluated using Equal Error Rate (EER) along with Receiver Operating Characteristics (ROC) Curve.

The article is arranged as follows: Section III ASV based spoof detection framework and Section IV describes the LCNN architecture and hyperparameter testing. The Section V includes the experimental results and discussion while Section VI is the Conclusion of the work.

III. Spoof Detection System

The proposed spoof detection framework is portrayed in Fig. 1 showing two phases of operation. The training phase is also called enrolment phase where known authentic speaker samples are enrolled along with spoofed speech samples. Initially, the CQT features are extracted to obtain a spectrographic representation of speech [22]. The two dimensional spectrogram along with their labels is then fed to the LCNN architecture to obtain the trained speaker model using a loss function. Furthermore, in the testing phase, the unknown test samples which are a mixture of genuine and spoof speaker samples, are represented using CQT features. This feature set is then tested using LCNN classifier and categorized as spoof or genuine. The CQT based features extraction and LCNN classifier are explained in following sub-Sections III-A and III-B.

A. Front-end CQT Features

The CQT features were introduced few decades ago as an alternative to short-time Fourier transform (STFT) [22]. The STFT being a filtering technique for a long spectrum broken down into shorter windows leads to an increasing Q-factor towards higher frequencies. This is exactly opposite to the human speech perception model where the Q-factor is found to be constant from 500Hz to 20kHz. Thus, STFT fails to represent the human perception model and CQT is preferred.

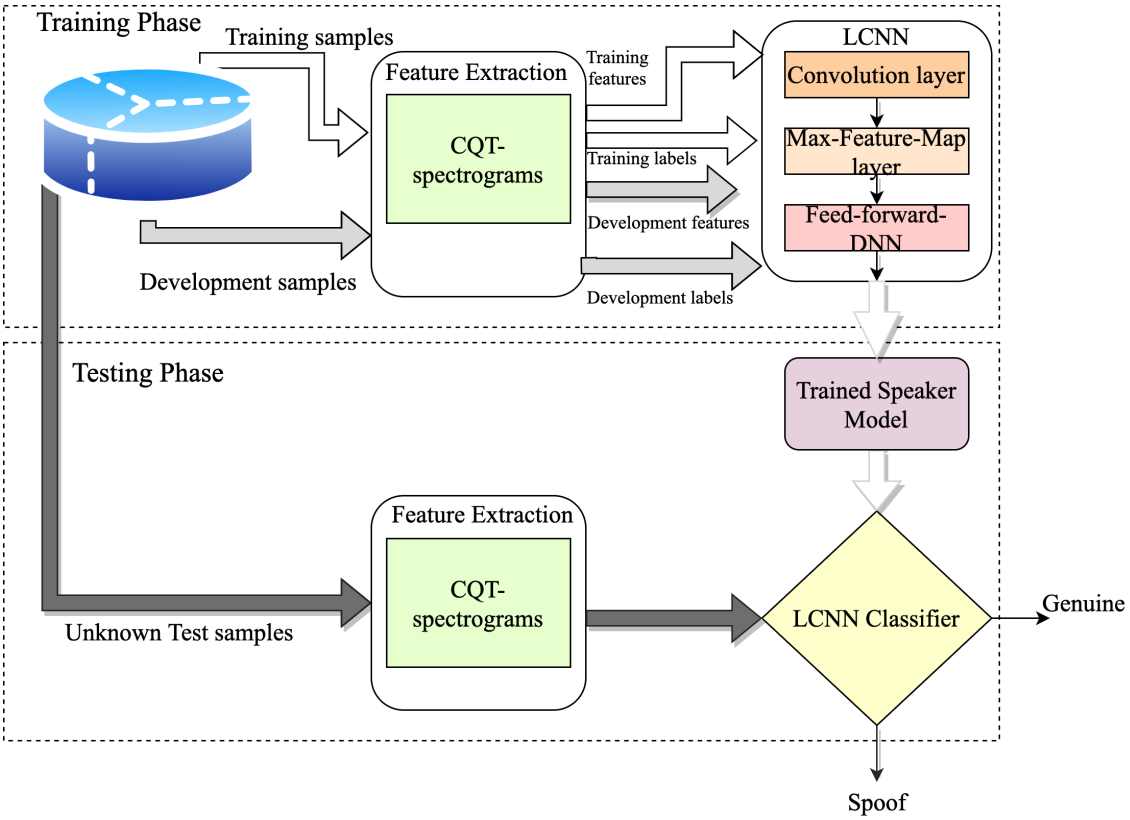

Fig. 1. Proposed Spoof Detection System.

The CQT based features, $C(i,m)$ are computed as shown in Eq. 1.

$$C(i,m) = \sum_{k=m-[M_i/2]}^{k=m+[M_i/2]} s(k) z_i^*(k-m+M_i/2) \quad (1)$$

Where, $i = 1$ to I, is the index of frequency bins, M_i is window length which is a variable and z_i^* is complex conjugate of basis. Hence, at lower frequencies, a high resolution is obtained wrt frequencies and at higher frequencies, high temporal resolution is possible. Thus overcoming the shortcoming of STFT with fixed time-frequency resolutions [21], [23].

B. Back-end LCNN Classifier

The LCNN are popular due to their reduction in network parameters with nearly similar error rates as the CNNs [19]. In this work, we have employed a compact version of LCNN structure [21] using the Maximum Feature Mapping activation (MFM) layer. It is based on the Max-out activation which has the ability to choose the right features for problem solving purposes. The combination of MFM and multiple Batch Normalization (BN) layers form a LCNN structure with dense layer at end that wraps up the overall output from the previous layers. Also, after alternate MFM layer, a max pooling layer is added which picks out max value out a patch of feature map rather than input feature map. A more detailed information can be found in Section IV.

IV. LCNN Structure and Hyperparameter Tuning

The conventional CNN uses activation function in the convolution layer, typically ReLU [15]. The CNN with MFM activation triggers two neurons and ignores one (in case of 2/1 MFM). This is termed as competitive relationship; hence MFM acts as a fine feature selection algorithm embed inside a CNN. The LCNN network used in this work has nine MFM-convolution, 4 max Pooling, 7 BN layers and 2 Fully Connected (FC) layers as shown in Fig. 2.

The BN layer is appended after every convolution layer as it leads to faster convergence and improved accuracy. For an input convolution layer, $v^k \in R^{h \times w}$, where $k = \{1, 2, \ldots, 2K\}$, w is spatial width while h is spatial height. The MFM activation is given as shown in Eq. 2.

$$\hat{v}_{a,b}^m = \max(v_{a,b}^m,\ v_{a,b}^{m+K}) \quad (2)$$

Where $2K$ is number of channels specific to input layer, m, a and b are indices for channel, width and height respectively. Therefore, the output dimension is $R^{h \times w \times K}$ and the gradients are calculated as shown in Eq. 3 and Eq. 4.

$$\frac{\delta \hat{v}_{a,b}^m}{\delta v_{a,b}^m} = \begin{cases} 1, & if\ \ v_{a,b}^m \geq v_{a,b}^{m+K} \\ 0, & elsewhere \end{cases} \quad (3)$$

$$\frac{\delta \hat{v}_{a,b}^m}{\delta v_{a,b}^{m+K}} = \begin{cases} 0, & if\ \ v_{a,b}^m \geq v_{a,b}^{m+K} \\ 1, & elsewhere \end{cases} \quad (4)$$

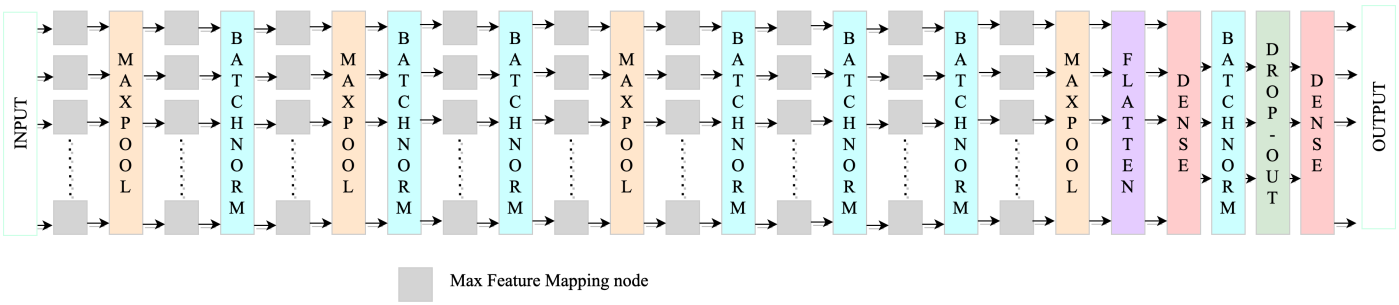

Fig. 2. Light Convolutional Neural Network Architecture for Spoof Detection System.

TABLE I. VARIOUS OPTIMIZERS WITH THE UPDATE RULE

Optimizer	Update Rule	Description
Adam	$\Theta_{s+1} = \Theta_s - \frac{\alpha}{\sqrt{\hat{v}_s}}\hat{m}_s$	s = time step, α = learning parameter, Θ_s = model parameters, m_s and v_s are decaying average gradients of past gradients and square of the gradients respectively.
Adadelta	$\Theta_{s+1} = \Theta_s + \triangle\Theta$ $\triangle\Theta = \frac{RMS[\triangle\Theta]_{s-1}}{RMS[g]_s} g_s$	g_s = gradient of objective function
Adagrad	$g_{s,i} = \bigtriangledown_\Theta J(\Theta_{s,i})$ $\Theta_{s+1,i} = \Theta_{s,i} - \alpha g_{s,i}$	$J(\Theta_{s,i})$ = objective function i =. individual parameter
SGD	$\Theta = \Theta - \alpha \bigtriangledown_\Theta J(\Theta; T^i; O^i)$	T^i = i^{th} training sample O^i = i^{th} label
SGD with momentum	$\Theta = \Theta - v_s$ $v_s = \sigma v_{s-1} + \alpha \bigtriangledown_\Theta J(\Theta)$	σ =. momentum factor
SGD with NAG	$\Theta = \Theta - v_s$ $v_s = \sigma v_{s-1} + \alpha \bigtriangledown_\Theta J(\Theta - \sigma v_{s-1})$	v_{s-1} = square of the previous decaying gradient
Nadam	$\Theta_{s+1} = \Theta_s - m_s$ $m_s = \sigma v_{s-1} + \alpha g_{s,i}$	α = learning rate Θ_{s+1} = future model parameter
RMSprop	$\Theta_{s+1} = \Theta_s - \frac{\alpha}{\sqrt{E(g^2)_s + \rho}} g_s$	$E(g^2)_s$ = running average at time step s, ρ = smoothing term

TABLE II. DETAILS OF ASV SPOOF 2019 DATASET

	Logical Access		Physical Access	
	Genuine	Spoofed	Genuine	Spoofed
Training Set	2580	22800	5,400	48,600
Development Set	2548	22296	5,400	24,300
Total	5128	45096	10800	72900
	50224		83500	
	133724 Samples			

Apparently, half the information bearing neurons are acquired by $2/1$ MFM activation. Thus, implying 50% reduction in comparison to conventional CNN architecture. This is due to the element-wise maximum computation for all the feature channels. Hence, leading to sparser connections.

Additionally, this work experiments with different renowned optimizers to calculate the loss function including SGD, with momentum and NAG, Adaptive gradient techniques such as Adam, Adadelta, Adagrad and RMSprop optimizers [24]. Furthermore, we experimented by changing the activation functions from state-of the-art softmax-Adam optimizer [20] based LCNN architecture to ReLU and logistic activations. The aim to try out various optimizers and activations is to investigate the appropriate combination of individual optimization algorithms with respective activation functions. Many parameters have different working scenarios to perform best and this gives the reason to explore other optimizers and activations specific to spoof detection scenario.

A. Gradient Optimization Algorithms

The optimization of hyper-parameters is an essential step in training any Deep Learning framework. In this work, we have tested various gradient optimizers for overcoming the challenges of tuning learning rate, slow convergence, over-fitting of the model and lower accuracy. The SGD algorithms are derived from Gradient descent optimizers with noisy stochastic convergence at each iteration for a particular sample [25]. This implies that it can capture generality without the network to complete the training on the entire training set. On the other hand, the SGD algorithms might experience overshooting due to improper choice of learning rate. A small value of learning rate leads to slow convergence while a big value might lead to no convergence at all. To overcome this issue, momentum, NAG and adaptive optimizers are investigated. The momentum increases the speed of convergence towards steeper direction as against less steeper ones. The typical value of momentum is 0.9 [25]. Additionally, the NAG with momentum stores future gradients to speed-up the convergence by improving the learning rate to higher or lower values accordingly.

Apart from momentum and NAG, a vivid way of improving the performance of gradient optimizers is through adaptive gradient techniques. The AdaGrad [26] is one such optimizer that makes larger updates for not so frequent parameters while small updates for frequent ones. This also leads to accumulation of past gradients ultimately leading to a zero learning rate. In contrast to Adagrad, the Adadelta [26] uses a fixed

TABLE III. EXPERIMENTAL RESULTS FOR LA AND PA ATTACK VARIOUS OPTIMIZERS AND ACTIVATION FUNCTIONS

Type of System	Type of Attack		PA			LA		
	Activation	Optimizer	Epochs	Learning rate	EER	Epochs	Learning rate	EER
Baseline	**Softmax**	**Adam**	100	0.00001	11.949	50	0.00001	11.282
Proposed	**Softmax**	**Adadelta**	100	0.001	15.989	100	0.001	15.433
		Adagrad	100	0.001	11.022	50	0.001	11.559
		SGD	100	0.001	21.438	100	0.001	15.687
		SGD momentum	100	0.001	12.844	100	0.001	**9.055**
		SGD-nesterov	100	0.001	**9.951**	100	0.0001	10.671
		Nadam	100	0.00001	12.091	50	0.0001	10.312
		RMSprop	100	0.00001	11.834	100	0.00001	11.387
	ReLU	**Adam**	100	0.00001	53.87	50	0.00001	43.303
		Adadelta	100	0.001	38.978	100	0.001	45.721
		Adagrad	100	0.001	48.814	100	0.001	43.981
		SGD	100	0.0001	48.385	100	0.001	58.71
		SGD momentum	100	0.0001	59.073	100	0.001	50
		SGD-nesterov	100	0.0001	56.507	100	0.001	50
		Nadam	100	0.0001	44.681	100	0.0001	49.986
		RMSprop	100	0.00001	48.033	50	0.00001	54.692
	Sigmoid	**Adam**	100	0.00001	12.245	100	0.00001	**10.015**
		Adadelta	100	0.001	18.844	100	0.001	21.39
		Adagrad	100	0.001	15.85	100	0.001	13.984
		SGD	100	0.0001	30.07	100	0.001	14.379
		SGD momentum	100	0.0001	11.386	100	0.001	13.382
		SGD-nesterov	100	0.0001	12.27	100	0.001	10.535
		Nadam	100	0.0001	15.779	100	0.0001	18.275
		RMSprop	100	0.00001	**11.317**	100	0.00001	19.893

window to refrain from past gradient accumulation. Similarly, the RMSprop tries to fix the past gradient issue by averaging the square of the gradients. Furthermore, the Adam optimizer estimates learning rate for every parameter value. It is a fusion of RMSprop with momentum. Additionally, the amalgam of Adam with NAG is Nadam optimizer [25]. Further, we have applied an Early stopping condition by tracking the validation error with some patience to see if it is experiencing any changes; if not then, training is halted. Table I summarizes the update rule for all the discussed optimizers.

B. Activation Functions

The basis of any neural network to function the intended way is through activation. The activation functions lead the input to the output that speeds-up the training for capturing complex nature of the patterns within the input data. Usually the softmax and arg-softmax activations have been used in the LCNN architectures [21]. In this work, we have considered the combination of softmax, ReLU and logistic activation function to observe the loss characteristics with the chosen optimizers. The ReLU activation has been a popular choice amongst larger CNN as it overcomes the issue of vanishing gradients but at the same time experiences the dead neuron issue. While the logistic function is useful for binary classification tasks. The softmax is an extension of logistic activation as it works for multi-class problem [26].

V. EXPERIMENTAL RESULTS AND DISCUSSION

The performance of these optimizers is measured using the Equal Error Rate (EER) [27] and the ROC curve is plotted to check goodness of the classification algorithm signifying the Area Under Curve (AUC) value [28]. The EER must be ideally as low as possible. The system is trained and evaluated using the ASV spoof 2019 dataset that has synthetic (TTS and VC) and replay speech samples along with genuine speaker samples. There are 20 speakers (male and 12 female) including more than one hundred thousand samples with LA and PA attacks. This is currently the only large scale dataset with all three attack types and genuine samples. Also, in this work the development dataset is used for validation purposes. The test data or evaluation data has a lot of variation in environment conditions for replay speech and synthesizers of synthetic speech. Thus ensuring an unbiased testing scenario. Table II shows the details of ASV spoof 2019 dataset. Table III shows the EER along for various optimizers and activation functions for both the attacks while Fig. 3 and Fig. 4 portrays the ROC curve for individual activations and optimizers for LA and PA, respectively.

From Table III, the efficiency of sigmoid and softmax are similar in contrast to ReLU where it fails to capture generality in both the kind of attacks. The EER for ReLU is almost 50 for most of the optimizers implying that gradient is stuck in local minima rather than global minima. Additionally, the improper scaling of weights in ReLU function leads to loss of actual data being considered. The sigmoid and softmax are related as the latter is just an extension for multi-class problems. This results in similar performance by both the activations. Further, the spoofed samples have multiple types of attack generated from various VC and TTS sources. Hence, the softmax gives slightly better EER of 9.005 for LA and 9.951 for PA attack. Infact, the softmax is observed to converge faster with as low

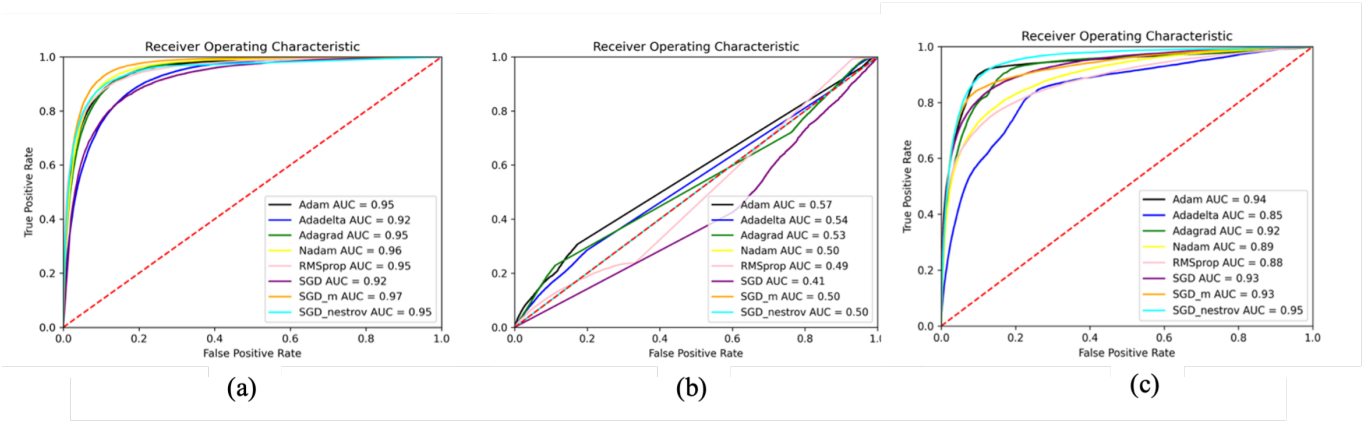

Fig. 3. ROC Curve for Various Activations and Optimizers for LA Attack (a) Softmax (b) ReLU (c) Sigmoid.

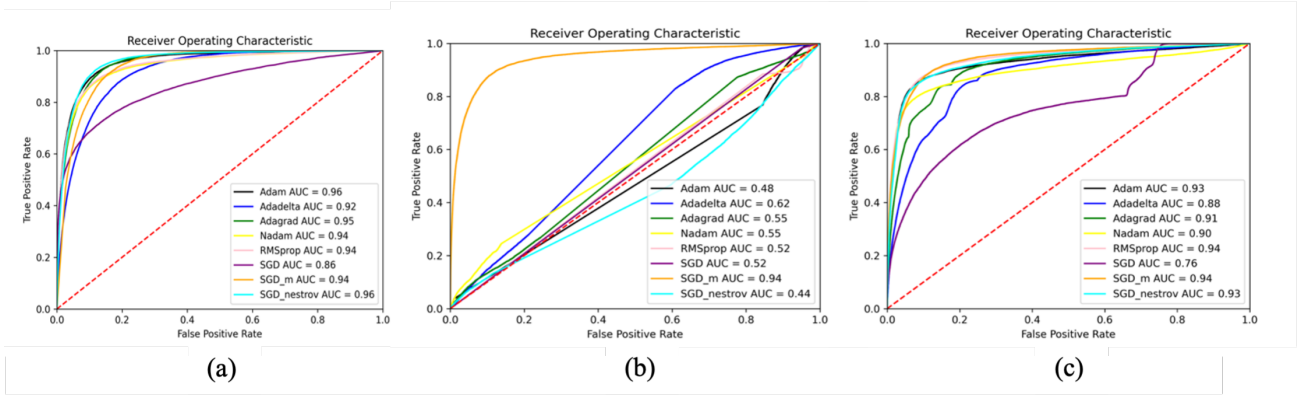

Fig. 4. ROC Curve for Various Activations and Optimizers for PA Attack (a) Softmax (b) ReLU (c) Sigmoid.

as 50 epochs for Adam and Adagrad for LA attack only.

In case of optimizer efficiency, Adam, RMSprop, SGD with momentum and NAG lead in EER for all the activations. The test condition comprise of noise in the dataset, hence RMSProp is the obvious performer for sigmoid activation with EER of 11.317. The SGD optimizer performs inconsistently with large variations in EER ranging from 14.379 to 30.070. This is due to lack of convergence and difficulty in adapting to convex problems. Thus as oppose to the SGD optimizer, the SGD with momentum and NAG are found to have a lower EER for both attacks. Hence, they are suitable for capturing generality like in the spoof detection task. The Adam optimizer performs consistently well with EER ranging from 10.015 to 11.949. So, it may be explored where generality is not of critical importance. Adadelta and Adagrad are not the shining performers but Adagrad gives a 0.3% improvement in EER than Adadelta; yet they perform poorly in comparison to Adam. The Nadam performs well for softmax optimizer while its performance worsens with increase in EER for sigmoid activation. The overall choice of activation will be softmax with any optimizer from the ones leading. Also, the EER for LA attack is lower than PA attack. Thus, the network efficiency is explicitly achieved for LA attack.

To confirm the performance of various optimizers the Receiver Operating Characteristics (ROC) curve with AUC are shown in Fig. 3 and Fig. 4 for LA and PA attack respectively. The required value of AUC is between 0 and 1 with values closer to 1 implying a good classifier. The Fig. 3(a) shows ROC for softmax function where all the optimizers perform well. The SGD with mometum has exceptional AUC of 0.97. In Fig. 3(b), none of the optimizers are able to form a learning rule in case of ReLU activation implying the the ReLU classifiers are not suitable for spoof detection task. The Fig. 3(c) confirms that the Adam and SGD with NAG have same AUC of 0.95 which is best amongst the other optimizers for sigmoid activation. Simiarly for PA attack, from Fig. 4(a) the ROC for softmax function shows all the optimizers perform well except SGD which has AUC of 0.86, while in Fig. 4(b), no significant efficiency is observed for ReLU activation. Lastly, Fig. 4(c), in case of sigmoid activation, the RMSprop and SGD momentum have same AUC of 0.94 which are better amongst the rest of the optimizers.

VI. Conclusion

The goal of conducting this study was proving that initialization of the network prior to training and tuning of parameters during the training improves the network accuracy. Thus in this work, a comprehensive comparison of various optimizers was carried out on LA and PA attack data. The rationale for conducting such a study was to signify the role of optimizers in classifying the test samples accurately.

Moreover, the activation functions were also considered in this comparative work to highlight their role based on nature of input—output data. The softmax and sigmoid prove to be better as against the ReLU function in the LA attack. Also, the networks converged faster with less number of epochs for Adam optimizers. In case of PA attack, the softmax function performed not so well and so did the ReLU function; while sigmoid showed significant improvement in accuracy in comparison to the other two. Further, it was evidently found that the RMSprop performed consistently well amongst all the others;while the SGD with momentum performed better than SGD but not so well against SGD with NAG. On the whole, the choice of optimizer, learning rate and activation affect the accuracy of the training network and thus the overall performance of the spoof detection system. In future, this work may be extended to experimenting with more activations like leaky-ReLU, Exponential linear unit and parametric ReLU; while optimizers such as AMSGrad may be explored to solve the issues of current adaptive algorithms.

ACKNOWLEDGMENT

The authors are grateful for research support from School of Computer Science and funding from Taylor's University, Malaysia.

REFERENCES

[1] H. Zeinali, H. Sameti, and L. Burget, "HMM-Based Phrase-Independent i-Vector Extractor for Text-Dependent Speaker Verification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 7, pp. 1421–1435, 2017.

[2] P. Korshunov and S. Marcel, "A Cross-Database Study of Voice Presentation Attack Detection," in *Handbook of Biometric Anti-Spoofing*, 2019, pp. 363–389.

[3] T. Kinnunen, M. Sahidullah, H. Delgado, M. Todisco, N. Evans, J. Yamagishi, and K. A. Lee, "The ASVspoof 2017 Challenge: Assessing the Limits of Replay Spoofing Attack Detection," in *INTERSPEECH*. ISCA, August 2017, pp. 2–6.

[4] Z. Wu, A. Khodabakhsh, C. Demiroglu, J. Yamagishi, D. Saito, T. Toda, and S. King, "SAS: A speaker verification spoofing database containing diverse attacks," in *International Conference on Acoustics, Speech and Signal Processing - ICASSP*. IEEE, 2015.

[5] J. Yamagishi, M. Todisco, M. Sahidullah, H. Delgado, X. Wang, N. Evans, T. Kinnunen, K. Aik Lee, V. Vestman, and A. Nautsch, "ASVspoof 2019: Automatic Speaker Verification Spoofing and Countermeasures Challenge Evaluation Plan," in *ASV Spoof 2019 Challenge*, 2019.

[6] M. Todisco, X. Wang, V. Vestman, M. Sahidullah, H. Delgado, A. Nautsch, J. Yamagishi, N. Evans, T. Kinnunen, and K. A. Lee, "ASVSpoof 2019: Future horizons in spoofed and fake audio detection," in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2019, pp. 1008–1012.

[7] B. Nasersharif and M. Yazdani, "Evolutionary fusion of classifiers trained on linear prediction based features for replay attack detection," *Expert Systems*, vol. 38, no. 3, 2021.

[8] M. Singh and D. Pati, "Usefulness of linear prediction residual for replay attack detection," *AEU - International Journal of Electronics and Communications*, vol. 110, p. 152837, 2019.

[9] Y. Zhang, F. Jiang, and Z. Duan, "One-class Learning Towards Synthetic Voice Spoofing Detection," *IEEE Signal Processing Letters*, vol. 28, pp. 937 – 941, 2020.

[10] M. Alzantot, Z. Wang, and M. B. Srivastava, "Deep Residual Neural Networks for Audio Spoofing Detection," in *Annual Conference of the International Speech Communication Association, INTERSPEECH*. ISCA, September 2019, pp. 1078–1082.

[11] J. Yang and R. K. Das, "Improving anti-spoofing with octave spectrum and short-term spectral statistics information," *Applied Acoustics*, vol. 157, p. 107017, 2020.

[12] S. Duraibi, W. Alhamdani, and F. T. Sheldon, "Voice Feature Learning using Convolutional Neural Networks Designed to Avoid Replay Attacks," in *IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, December 2020, pp. 1845–1851.

[13] X. Xiao, X. Tian, S. Du, H. Xu, E. S. Chng, and H. Li, "Spoofing speech detection using high dimensional magnitude and phase features: The NTU approach for ASVspoof 2015 challenge," in *International Speech Communication Association, INTERSPEECH*. ISCA, 2015.

[14] B. Chettri, D. Stoller, V. Morfi, M. A. M. Ramirez, E. Benetos, and B. L. Sturm, "Ensemble Models for Spoofing Detection in Automatic Speaker Verification," in *Annual Conference of the International Speech Communication Association, INTERSPEECH*. ISCA, September 2019, pp. 1018–1022.

[15] M. Dua, C. Jain, and S. Kumar, "LSTM and CNN based ensemble approach for spoof detection task in automatic speaker verification systems," *Journal of Ambient Intelligence and Humanized Computing 2021*, pp. 1–16, 2021.

[16] F. Ye and J. Yang, "A Deep Neural Network Model for Speaker Identification," *Applied Sciences*, vol. 11, no. 8, pp. 1–18, 2021.

[17] F. Fang, J. Yamagishi, I. Echizen, M. Sahidullah, and T. Kinnunen, "Transforming acoustic characteristics to deceive playback spoofing countermeasures of speaker verification systems," in *International Workshop on Information Forensics and Security (WIFS)*. IEEE, December 2018, pp. 1–9.

[18] C. Zhang, C. Yu, and J. H. L. Hansen, "An Investigation of Deep-Learning Frameworks for Speaker Verification Antispoofing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 4, pp. 684–694, 2017.

[19] M. Volkova, T. Andzhukaev, G. Lavrentyeva, S. Novoselov, and A. Kozlov, "Light CNN Architecture Enhancement for Different Types Spoofing Attack Detection," in *Lecture Notes in Computer Science*, 2019, vol. 11658, pp. 520–529.

[20] X. Li, N. Li, C. Weng, X. Liu, D. Su, D. Yu, and H. Meng, "Replay and Synthetic Speech Detection with Res2Net Architecture," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, June 2021, pp. 6354–6358.

[21] G. Lavrentyeva, S. Novoselov, A. Tseren, M. Volkova, A. Gorlanov, and A. Kozlov, "STC Antispoofing Systems for the ASVspoof2019 Challenge," in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*. ISCA, September 2019, pp. 1033–1037.

[22] L. Liu and J. Yang, "Study on Feature Complementarity of Statistics, Energy, and Principal Information for Spoofing Detection," *IEEE Access*, vol. 8, pp. 141 170–141 181, 2020.

[23] M. Todisco, H. Delgado, and N. Evans, "A new feature for automatic speaker verification anti-spoofing: Constant Q cepstral coefficients," in *Odyssey 2016: Speaker and Language Recognition Workshop*, 2016.

[24] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," *Neurocomputing*, vol. 415, pp. 295–316, 2020.

[25] S. Ruder, "An overview of gradient descent optimization algorithms," *Lecture Notes in Computer Science*, vol. 11046, 2016.

[26] G. Habib and S. Qureshi, "Optimization and acceleration of convolutional neural networks: A survey," *Journal of King Saud University - Computer and Information Sciences*, 2020.

[27] Z. Chen, Z. Xie, W. Zhang, and X. Xu, "ResNet and Model Fusion for Automatic Spoofing Detection," in *Interspeech 2017*. ISCA, August 2017, pp. 102–106.

[28] I. Shahin, A. B. Nassif, N. Nemmour, A. Elnagar, A. Alhudhaif, and K. Polat, "Novel hybrid DNN approaches for speaker verification in emotional and stressful talking environments," *Neural Computing and Applications*, vol. 33, no. 23, pp. 16 033–16 055, 2021.