Study of Deep Neural Networks for Fingerprint Anti-spoofing

Jyothi kiran Gavini

Electrical Engineering

University of Missouri-Kansas City, MO-64110

jkgdm3@mail.umkc.edu

Declaration- I Mr. Jyothi kiran gavini, wrote paper by myself in my own words from the experiments conducted. Any data used in context is referred properly at the end.

Abstract

This paper covers usage of different features like LBP,BSIF and BGP taken from a LivDet 2011 database on deep neural network. Extracted features have live and spoof data in their vectors. These operations are performed on digital and Sagem sensor. Extracted features have training and test data for each senor and 20% of training data is used for validation. The deep neural networks have two layers with auto encoders and a softmax layer used for classification. Different random experiments are taken on to achieve parameters with highest AUC. The parameter's that gave better results were taken to experiment on sensors and plot the ROC curves for different combinations of sensor data.

Introduction

Fingerprint detection is one of the important technologies in the field of security. It has many applications ranging from cell phones to well-maintained lockers. These finger prints are claimed to be mostly unique but there are some exceptions. Print patterns, ridge points and minutia points are generally used to characterize different finger prints. Live det 2011 dataset is used to test different features like local binary patterns(LBP), Binary Gabor patterns(BGP), Binarized statistical image features(BSIF) extracted from the images tested on live and spoof images. These features have over 2000 images separated in half for live and spoof images for both training and dataset. Validation data is taken about ten percent from training data set, considering half of the data from live and spoof images. Binary label is created for live and spoof images in order to test accuracy for neural network on (Receiver operation curve)ROC curve ([1], page 388).

Deep Learning is one of the trending topics in todays technology, it has applications in automobiles, commercial industry, drones, translation and many other innovative technologies. The main behavioral aspect of a neural network is how well it tries to learn itself using different step sizes to achieve the desired output. Deep learning is already advanced on image classification tasks on huge datasets like ImageNet, CIFAR etc. These networks has deep convolutional blocks which led to overfitting after a certain number of layers, these are recently over taken by residual layers, bottleneck layers and fast R-CNN[2]. Deep networks can also be implemented in other fields of engineering to achieve better results. These neural networks involves large number of arithmetic operations and are well performed on (Graphics processing unit)GPU rather than (Central Processing unit)CPU.

Fingerprint detection is one of the important tasks in bio-security appliances. These features are often spoofed using different materials like latex, silicone and gelatin [3]. Since these features are able to pass the security systems, liveness detection has to be improved in order to avoid spoofing. However, these fingerprints can also suffer from aging, injury and other factors. So, apart from improving liveness detection features, having another security feature can enhance security. Liv-det 2011 has finger print images enrolled by different people through different sensors and has spoof images which could be done using materials described above

Section 2 of this paper explains about different methods and parameters used for auto encoders in deep neural networks. In section 3, the results obtained are discussed and evaluated for required performance on ROC curves. In Section 4, Conclusions were drawn based on the results obtained. Section 5 has references used to support the paper and section 6 contains different observation tables, formulas and figures used in the paper

.

Methods

Different features are extracted from the Liv-det 2011 database and are sent to the deep neural network. Here the deep neural network uses feed forward sampling the weights to the outputs and then uses back propagation ([1], page 149) to update the weights in order to achieve the desire output. Here the data is sampled to binary format because neural networks use ones and zero to classify the output. Training data is sent to the auto-encoders ([4], page 12), auto encoders are used for reduction in dimension in order to classify better with few options to choose from the inputs. Auto encoders are generally used to classify unlabeled data and target values are equal to input values. Auto encoders have four parameters in order to regulate the data passing through them. Hidden layers are used to create weights with respect to input in order to update them during backpropagation and these values are randomly generated. L2W ([4], page 6) is a weight regularization and weight decay method, which is used to reduce the degree of weights and helps in to prevent overfitting. Sparsity ([4], page 14) is used to find good relation between the inputs. Sparsity proportion is used to activate the neurons and to make note of average number of activations used in the network. Sparsity regularization helps in selecting the input variables that are best described to the outputs. These coefficients and weights are updated after each iteration ([1], page 246). The input features are compressed after it pass through each auto encoder, which will leave small number of features to classify at the output.

In the dataset, LBP has 54 features, BGP has 216 features and BSIF has 512 features. The output is classified as one and zero for live and spoof images. When the features are sent to the auto encoders, they result the same output features but the image values are compressed to a great extent, these compressed features are again sent to another auto encoder for further compression. The number of auto encoders need to selected is based on trail and error and depends on the required accuracy for a system. Taking more auto encoders might result in a saturated output or a reduction in accuracy after a certain extent. These compressed features are sent to binary classifier soft max layers which classifies the outputs as ones and zeroes depending on their matrix values for live and spoof images. Here, after the processing of auto encoders the data is sent to soft max layer for classification. Now all the blocks are stacked and a deep neural network is trained based on the training data along with the stacked block. These values are fine-tuned by passing them trained deep network again. This process involves backpropagation for the update of weights. Once the network is trained, test data is used in the already trained deep network along with binary labels to plot for the performance curve.

Results and Discussions:

First, different extracted test and train features are loaded and stored in the work space. Once the input data is chosen from a set of features, a random set of operations are performed in order to observe the best configuration and to see how different parameters like hidden layer, L2W weight regularization, Sparsity proportion, Sparsity regularization, softmax layers and epoch effect the training and testing conditions. ROC curve is used to evaluate the performance which is taken on spoof accept rate and live reject rate based on the confusion matrix. Here a combination of sensor features are evaluated on all the features in order to study the effect of parameters to opt them for further tasks in the project.

Hidden layers are used to create mappings, hidden layer sizes are improved from 100 to 300 in the first auto encoder and then used these values to estimate the AUC. Testing ROC curve is improved to 60.3 to 82.1 when the hidden layer size is gradually increased. Although, increasing hidden layers have a significant effect on ROC curve, the values doesn't differ by a large amount after 300 hidden layers (Refer Table1). So, in order to reduce computational time through back propagation, this is chosen as the best configuration. Having less number of hidden layers plots less weights while the increase in hidden layers shows a large dense of features grouped together.

Auto encoders are selected based on random observations performed in between one to three auto encoders. Selecting a single auto encoder yielded 50.6 accuracy in ROC curve while two auto encoders yielded 60.3 accuracy. This accuracy is again increased with respect to other parameters retrospectively. Taking three auto encoders has 65.6 percent accuracy compared to 65.3 percent accuracy of two auto encoders with same parameters, which is not a significant increase to introduce an another auto encoder, so two auto encoders are selected to operate on all the features. (Refer Table1)

L2W is used for weight regularization, L2W is helping to increase the accuracy when it is changed from 0.004 to 0.008 for both auto encoders from 81.1 to 83.4. When the value is changed to 0.012 in the first auto encoder the performance is degraded to 75.6. So, the L2W is set to 0.004 in all the features. (Refer Table1)

Sparsity regularization is measured for different features ranging from one to three. An accuracy of 49.1 is noted when the sparsity regularization is one, when it is increased to 2, an increase in ROC is noted from 49.1 to 50.6. Considering different hidden layers and other parameters, for the same set of features the values are reduced when the measure is increased from two to three from 75.6 to 65.3. So, two is chosen as sparsity regularization factor for all other features. (Refer Table1)

Sparsity proportion is used to activate the neurons. When we increase the value of proportion, the values of ROC are increased. When a proportion of 0.3,0.4 and 0.5 are used to evaluate the performance, the respective ROC's are 62.8,68.8 and 70.2. The performance used in all the features is 0.4 because there is a huge rise in accuracy as compared to others. These configurations are done similarly for BGP and BSIF features to evaluate the performance. The values are kept constant after configuration to test which feature has better performance compared to other. (Refer Table1)

Epoch is used for backpropagation to count the number of iterations, generally epoch is twice the iterations back and forth in the neural network. When the number of epoch is increased the performance is increased because the data is trained for a longer time but it necessarily does not it will have huge variations in the data. The training accuracy increased from 82.1 to 83.4 when epoch is increased from 100 to 200. Here, the values can be higher if the epoch values are more than 1000 but it requires lot of system memory speed to compute these evaluations. (Refer Table1)

In task1, digital and sagem sensors are used to evaluate the performance on the same sensor. Here, training data is taken across auto encoders for both digital and sagem sensors. Validation data is split by taking 100 images from live data and 100 images from spoof data to evaluate the performance across same parameters for same sensors. In digital sensor, BGP has the best accuracy for testing, training and validation curves. It has 96.8 training accuracy, 84.3 testing accuracy and 99.3 percent validation. These scores are higher than LBP and BSIF features. Similarly, when Sagem sensor is used to test these features BGP has highest accuracy in all the three representations. It has 99.91 percent training accuracy which is less by 0.07 than BSIF features, but it doesn’t matter a lot because it has an average increase in other fields. BGP has 89 percent testing accuracy and 96.6 percent validation accuracy. So, BGP has better liveness detection compared to LBP and BSIF after evaluation. These values are presented in the table as referred below. (Refer Table4)

In task2, same features are used from different sensors to test ROC curves. Here, performance is evaluated by using fixed features as used in task 1. BSIF has 66.5 testing accuracy when digital data is used for training and sage data is used for testing. BSIF has higher accuracy in training, testing and validation. BSIF has 99.8 training accuracy and 84.53 validation accuracy. LBP has comparatively lower values compared to other two features, which states that this is not efficient as other two features for detection. While, Sagem sensor is used for training and digital is used for testing, BSIF has higher testing and training accuracy by 59.9 percent and 99.9 percent. The testing values are comparatively less compared to task1 because of different test data used. LBP has higher validation by 98 percent. However, BSIF has comparatively higher rates than other when opposite sensors are used for testing. When same sensors are used for testing, BGP has higher accuracy. (Refer Table5)

Fusion of features are taken on same sensor which has different set of values concatenated together. Values are validation and test are taken respectively along with the concatenated features of training data. Few samples from each feature is taken to validate for both live and spoof images. These values are performed on digital sensor. After successful evaluation, training is considerably in all the three cases but testing and validation is degraded to half performance. On the other side, all of these results are similar to one another. This could be because of mixed values given to training data from different features. Mixing two features is not a better way to work with liveness detection. (Refer Table6)

Conclusion:

In this project, liv-det 2011 dataset is used and features like LBP, BSIF and BGP are extracted with respective test and train datasets. Different configurations for auto encoders are considered and evaluate. In order to get optimal performance two auto encoders are used with hidden layers size of 300 and 100, L2W of 0.008, sparsity proportion of 0.4 and sparsity regulation of 2. These features are tested on same and different sensor data. BGP has better performance when same sensor data is used and BSIF has better performance when different sensors are used but it has degraded performance since the data is different. When combination of features are taken on same sensor, Performance remains almost same in all cases with degraded performance.

References

1. J. C. Príncipe, et al., Neural and adaptive systems: fundamentals through simulations. New York: Wiley, 1999.
2. He, Kaiming, et al. "Deep residual learning for image recognition." *arXiv preprint arXiv:1512.03385* (2015).
3. Rattani, Ajita, and Arun Ross. "Automatic adaptation of fingerprint liveness detector to new spoof materials." *Biometrics (IJCB), 2014 IEEE International Joint Conference on*. IEEE, 2014.
4. Sparse auto encoder by Andrew Ng

Appendices

Table 1- AUC's of LBP Configuration

Sensor	Features	Hidden Layer1	Hidden Layer2	Hidden Layer3	No of Auto-encoders	Epoch	L2 Weight Regularisation	Sparsity Regulisation	Sparsity Proportion	Max Epoch	Testing AUC
Digital+Sagem	LBP	100	-	-	1	100	0.004	3	0.3	100	62.8
Digital+Sagem	LBP	100	-	-	1	100	0.004	3	0.4	100	68.8
Digital+Sagem	LBP	100	-	-	1	100	0.004	3	0.5	100	70.2
Digital+Sagem	LBP	100	-	-	1	100	0.004	2	0.4	100	50.6
Digital+Sagem	LBP	100	100	-	2	100	0.004, 0.004	2	0.4	100	60.3
Digital+Sagem	LBP	200	100	-	2	100	0.004, 0.004	2	0.4	100	81.1
Digital+Sagem	LBP	300	100	-	2	100	0.008, 0.008	2	0.4	100	82.1
Digital+Sagem	LBP	300	100	-	2	200	0.008, 0.008	2	0.4	200	83.4
Digital+Sagem	LBP	300	100	-	2	200	0.012, 0.008	2	0.4	200	75.6
Digital+Sagem	LBP	300	100	-	2	100	0.008, 0.008	1	0.8	100	65.3
Digital+Sagm	LBP	300	100	100	2	100	0.008, 0.008, 0.008	1	0.8	100	65.6

Table 2-AUC’s of BSIF Configuration

Sensor	Features	Hidden Layer1	Hidden Layer2	No of Auto-encoders	Epoch	L2 Weight Regularisati on	Sparsity Regulisation	Sparsity Proportion	Max Epoch	Testing AUC
Digital+Sagem	BSIF	100	100	2	100	0.004, 0.004	2	0.4	100	77.8
Digital+Sagem	BSIF	200	100	2	100	0.004, 0.004	2	0.4	100	77.2
Digital+Sagem	BSIF	300	100	2	100	0.008, 0.008	2	0.4	100	79.4
Digital+Sagem	BSIF	300	100	2	200	0.008, 0.008	2	0.4	200	80
Digital+Sagem	BSIF	300	100	2	100	0.008, 0.008	1	0.8	100	75.7

Table 3-AUC’s of BSIF Configuration

Sensor	Features	Hidden Layer1	Hidden Layer2	No of Auto-encoders	Epoch	L2 Weight Regularisati on	Sparsity Regulisation	Sparsity Proportion	Max Epoch	Testing AUC
Digital+Sagem	BGP	100	100	2	100	0.004, 0.004	2	0.4	100	78.9
Digital+Sagem	BGP	200	100	2	100	0.004, 0.004	2	0.4	100	80.8
Digital+Sagem	BGP	300	100	2	100	0.008, 0.008	2	0.4	100	82.1
Digital+Sagem	BGP	300	100	2	200	0.008, 0.008	2	0.4	200	83.91
Digital+Sagem	BGP	300	100	2	100	0.008, 0.008	1	0.8	100	79.2

Table 4-AUC’s of train, test and validation of same sensor

Sensor	Features	Hidden Layer1	Hidden Layer2	No of Auto-encoders	Epoch	Area Under Curve(Training)	Area Under Curve(Testing)	Area Under Curve(Validation)
Digital	LBP	300	100	2	200	9.9/90.1	20/80	03.8/96.2
Digital	BSIF	300	100	2	200	4.1/95.9	15.7/84.3	6.1/93.9
Digital	BGP	300	100	2	200	3.2/96.8	15.7/84.3	0.7/99.3
Sage	LBP	300	100	2	200	3.4/96.6	12.9/87.1	4.8/95.2
Sage	BSIF	300	100	2	200	0.02/99.98	14.3/85.7	10.1/89.9
Sage	BGP	300	100	2	200	0.09/99.91	11/89	3.4/96.6

Table 5-AUC's of train, test and validation of different sensor

Sensor	Features	Hidden Layer1	Hidden Layer2	No of Auto-encoders	Epoch	Area Under Curve(Training)	Area Under Curve(Testing)	Area Under Curve(Validation)
Train-Digital. Test-Sage	LBP	300	100	2	200	14.28/85.72	51.92/48.08	15.95/84.05
Train-Digital Test-Sage	BSIF	300	100	2	200	0.2/99.8	33.50/66.5	15.477/84.53
Train-Digital Test-Sage	BGP	300	100	2	200	6.07/93.93	34.6/65.4	15.06/84.94
Train-Sage Test-Digital	LBP	300	100	2	200	3.3/96.7	51.3/48.7	1.4/98.6
Train-Sage Test-Digital	BSIF	300	100	2	200	0.4/99.6	40.1/59.9	6.0/94
Train-Sage Test-Digital	BGP	300	100	2	200	3.4/96.6	54.19/45.81	2.5/97.5

Table 6-AUC's of train, test and validation of mixed feature values from same sensor

Sensor	Hidden Layer1	Hidden Layer2	No of Auto-encoders	Epoch	Area Under Curve(Training)	Area Under Curve(Testing)	Area Under Curve(Validation)
Digital LBP+BGP	300	100	2	200	99.9993	50.7	53
Digital BSIF+BGP	300	100	2	200	99.9993	50.1	53
Digital LBP+BSIF	300	100	2	200	99.9993	50.1	53

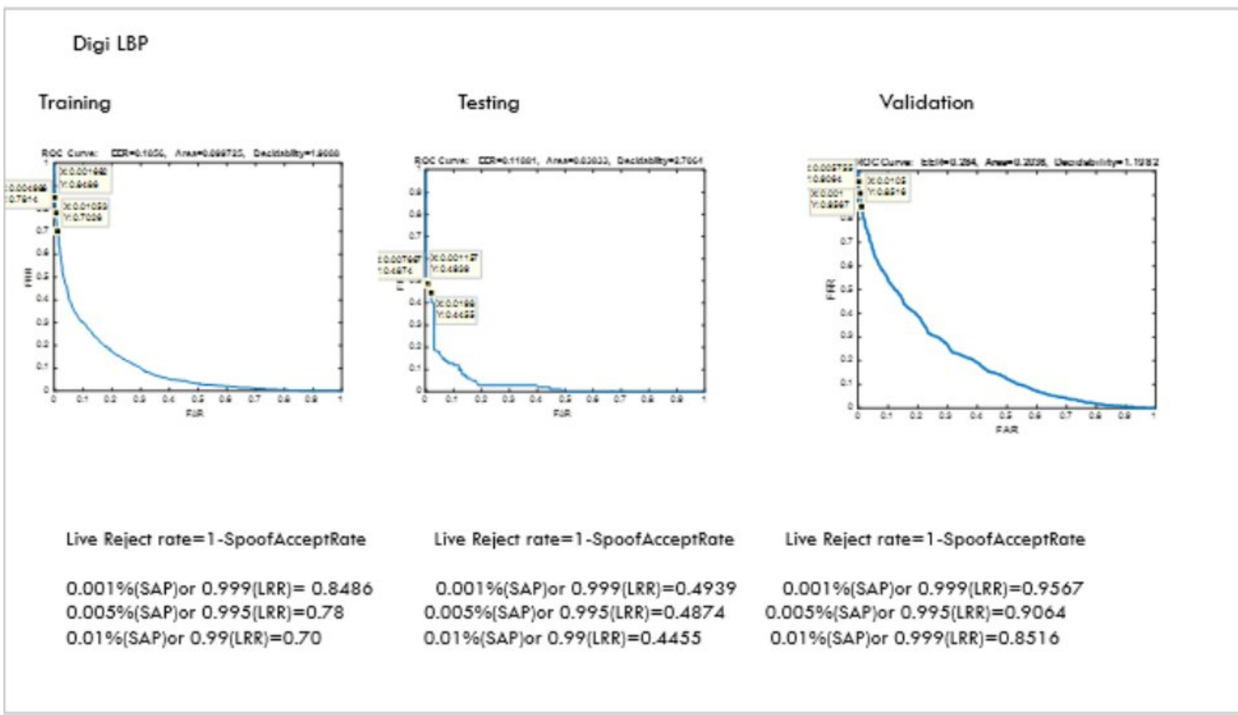

Fig1: Training Digital LBP and testing with Digital LBP

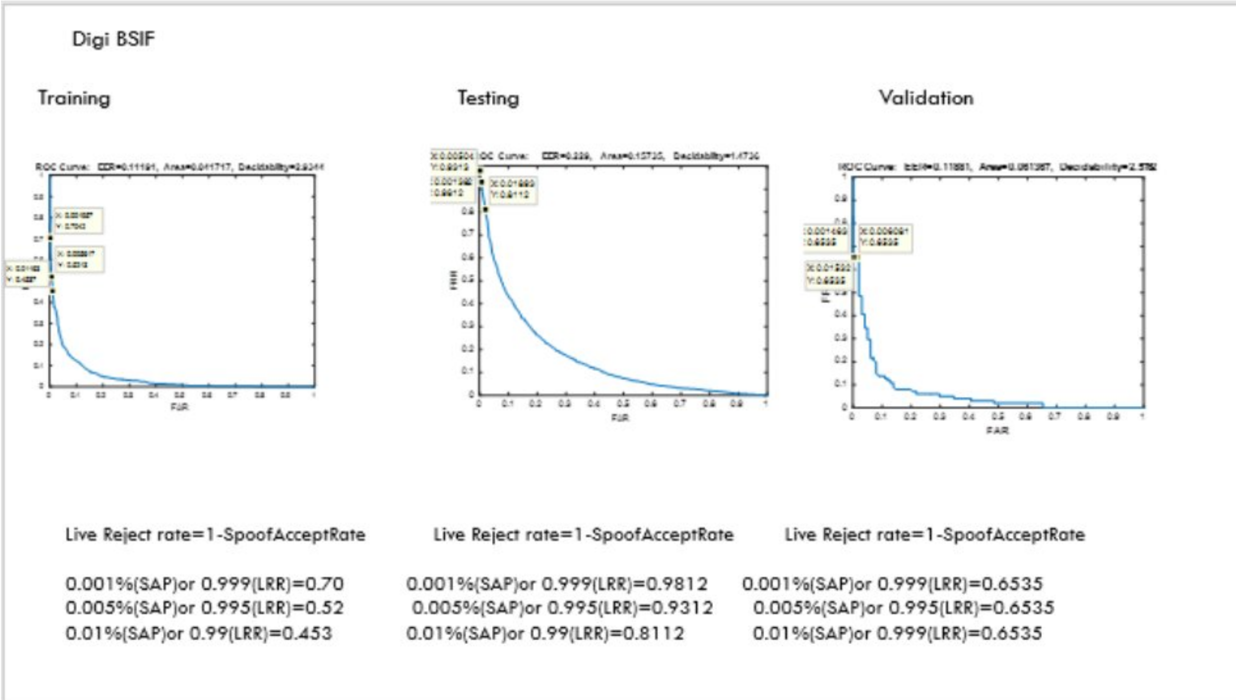

Fig2: Training Digital BSIF and testing with Digital BSIF

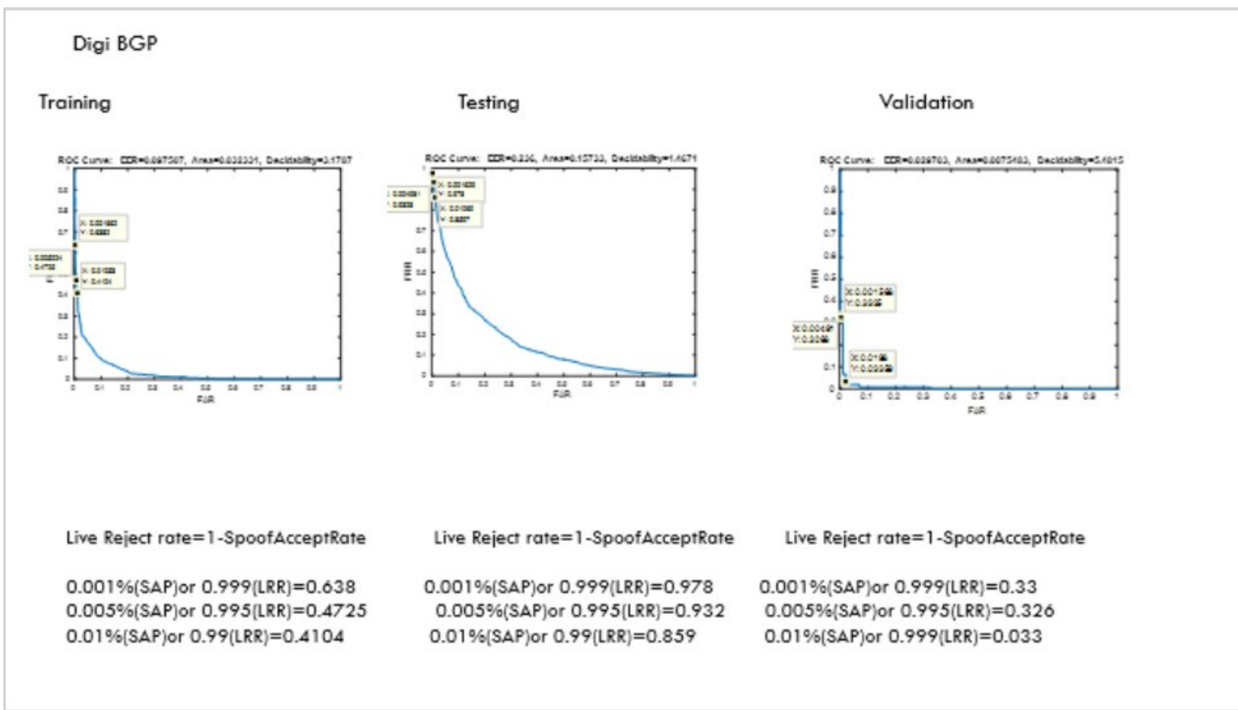

Fig3: Training Digital BGP and testing with Digital BGP

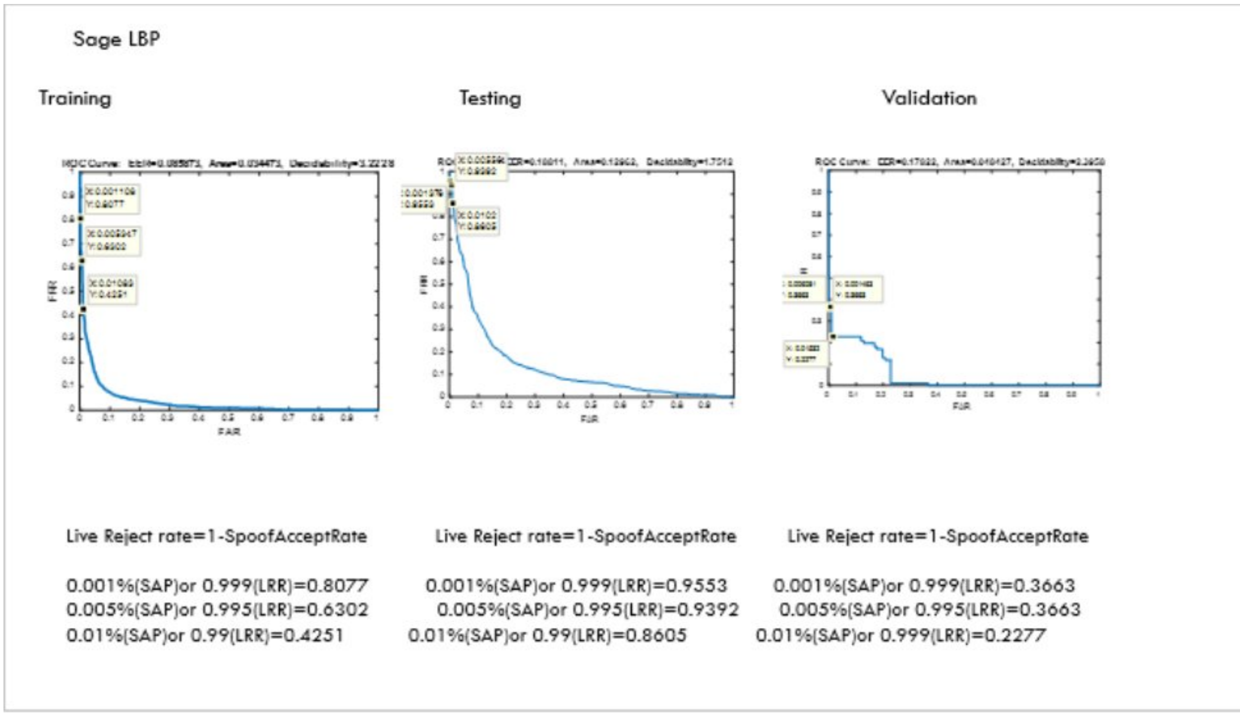

Fig4: Training Sage LBP and testing with Sage LBP

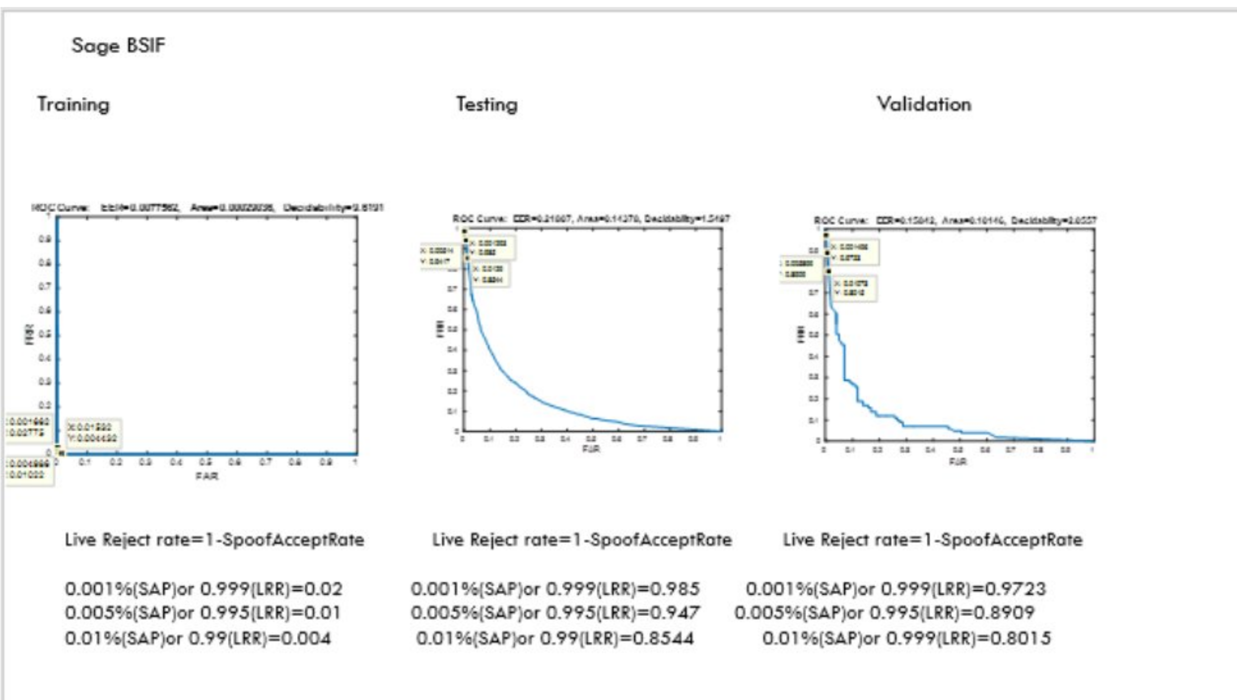

Fig5: Training Sage BSIF and testing with Sage BSIF

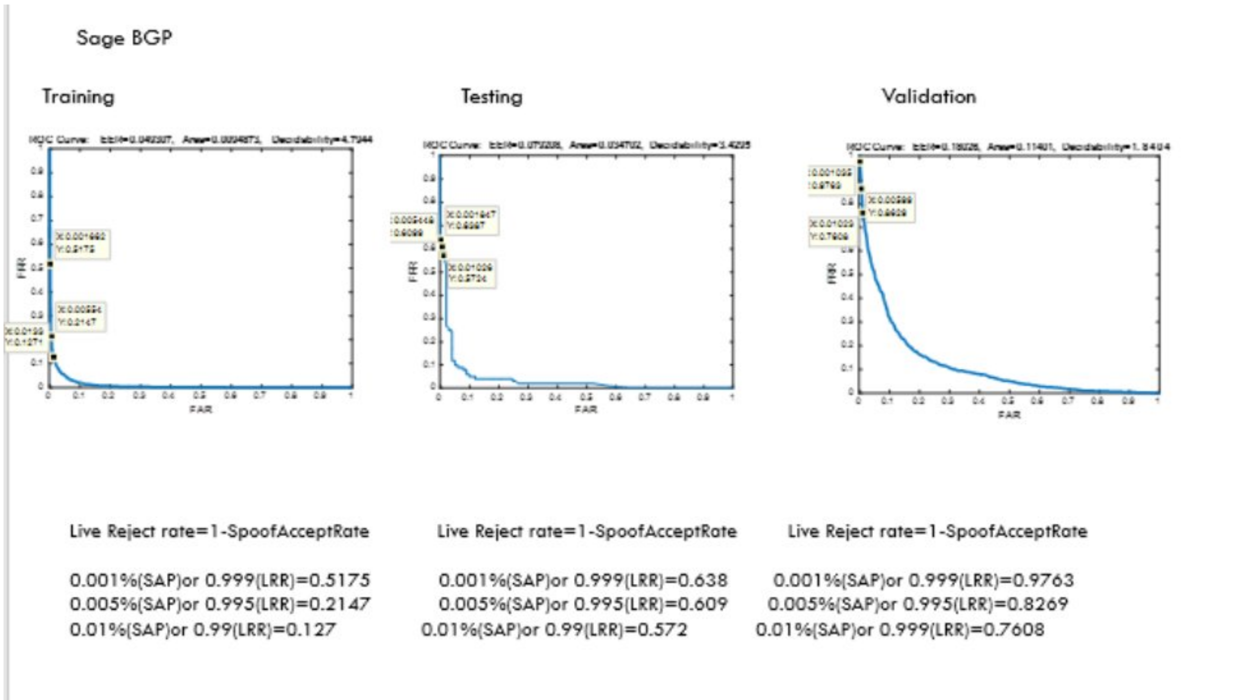

Fig6: Training Sage BGP and testing with Sage BGP

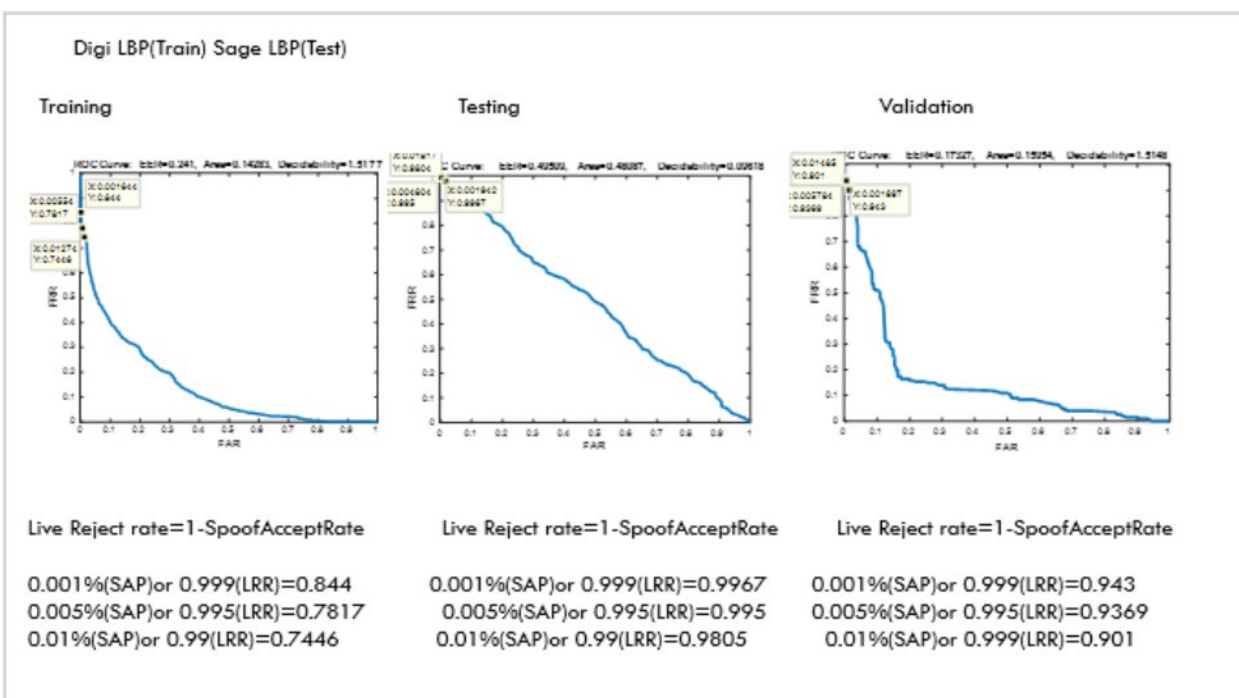

Fig7: Training Digi LBP and testing with Sage LBP

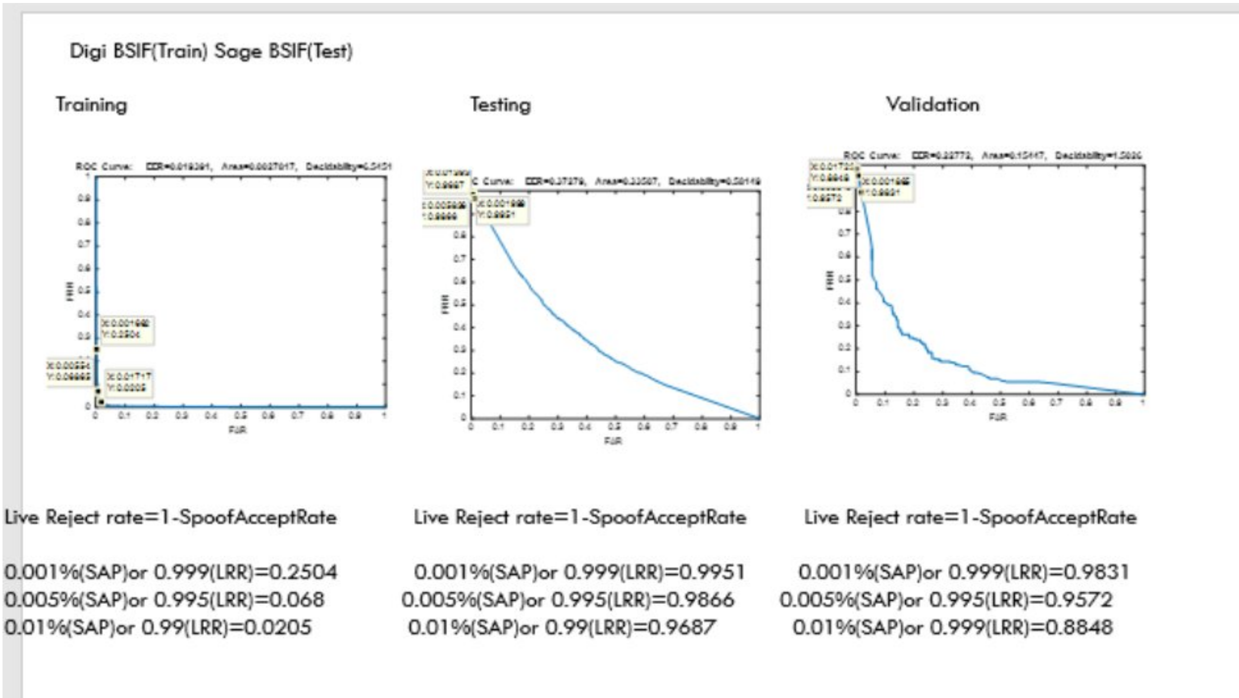

Fig8: Training Digi BSIF and testing with Sage BSIF

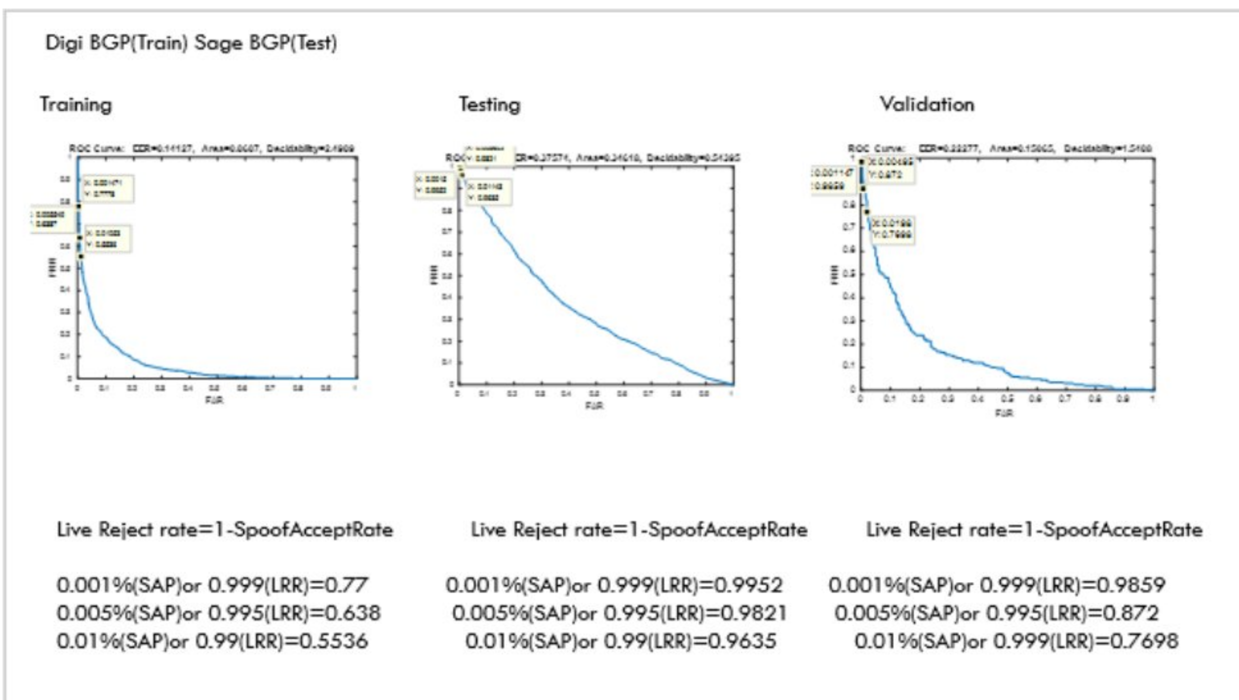

Fig9: Training Digi BGP and testing with Sage BGP

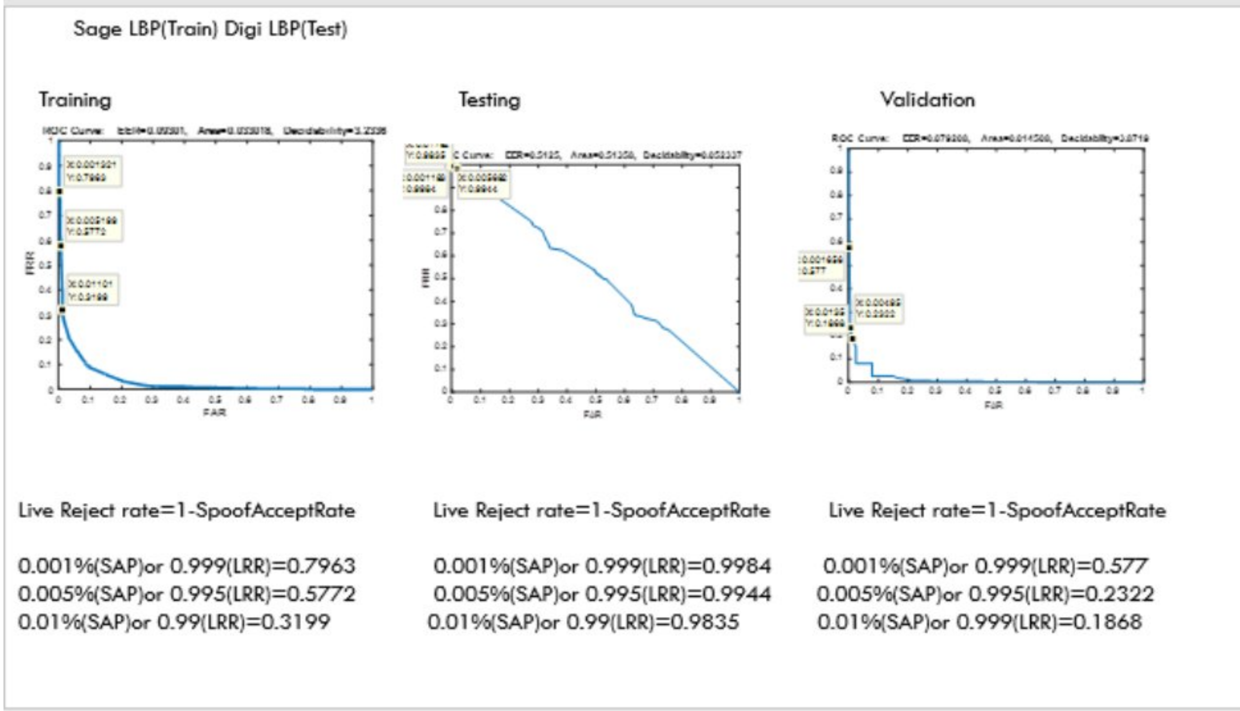

Fig10: Training Sage LBP and testing with Digi LBP

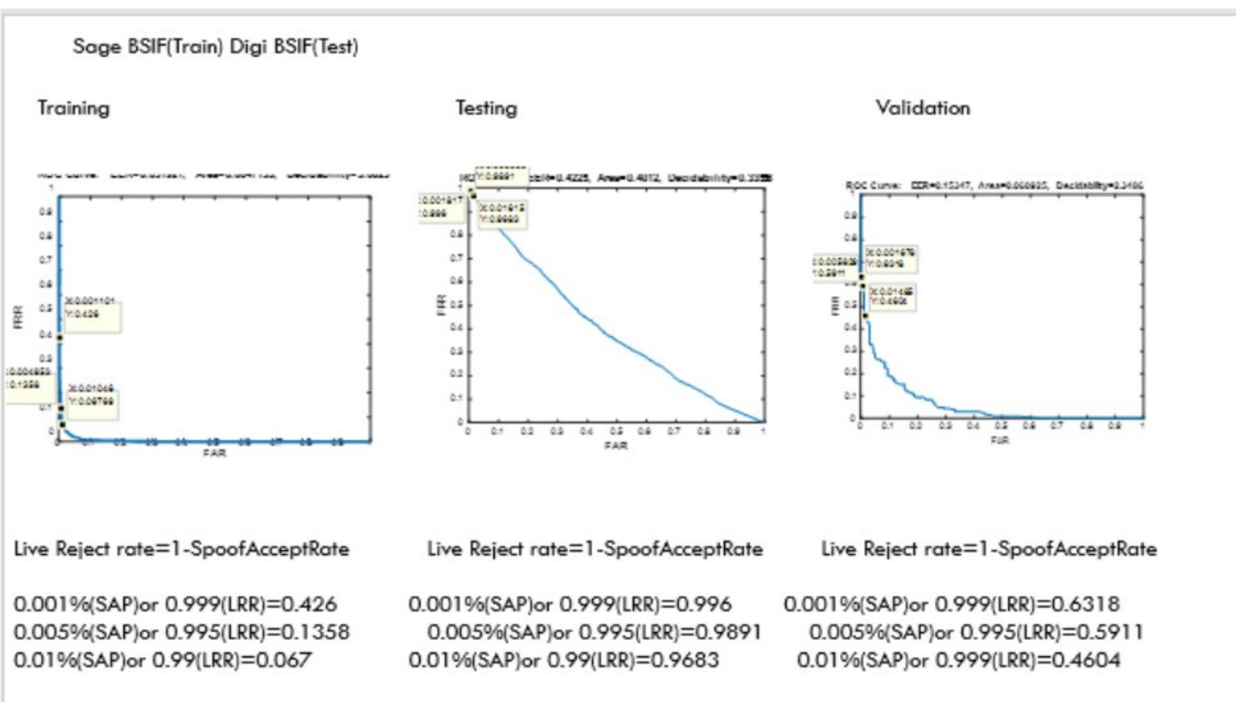

Fig11: Training Sage BSIF and testing with Digi BSIF

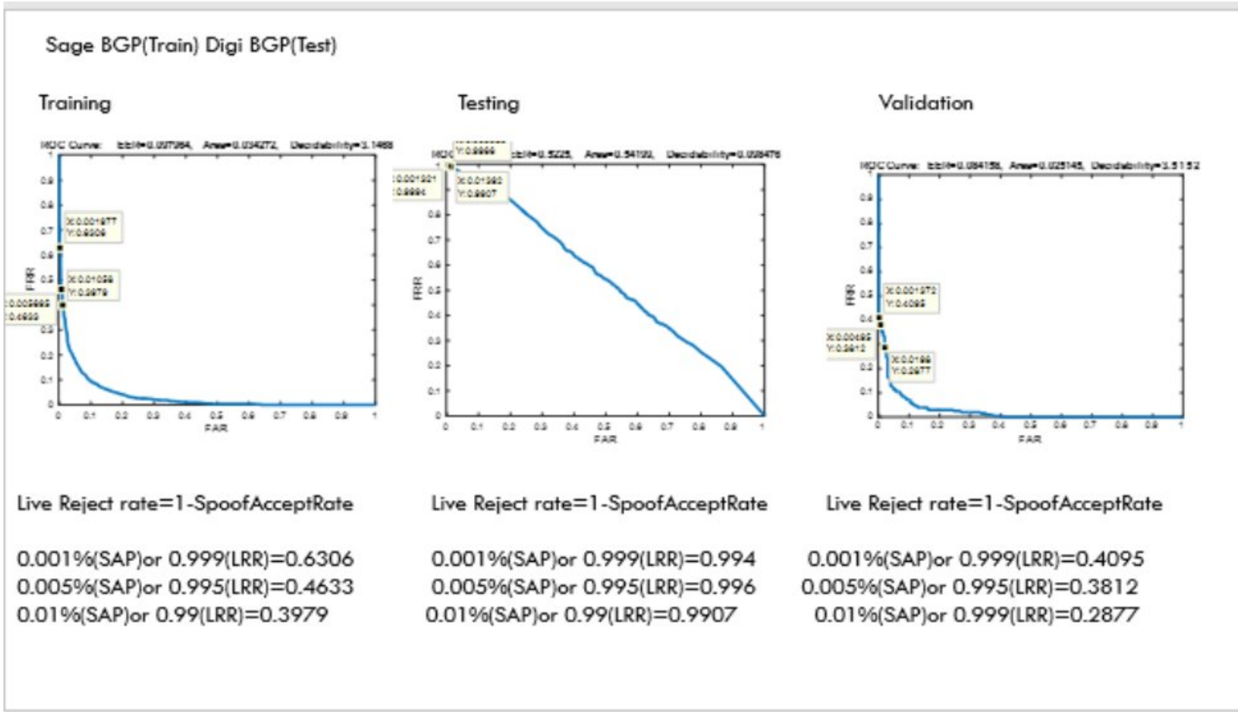

Fig12: Training Sage BGP and testing with Digi BGP

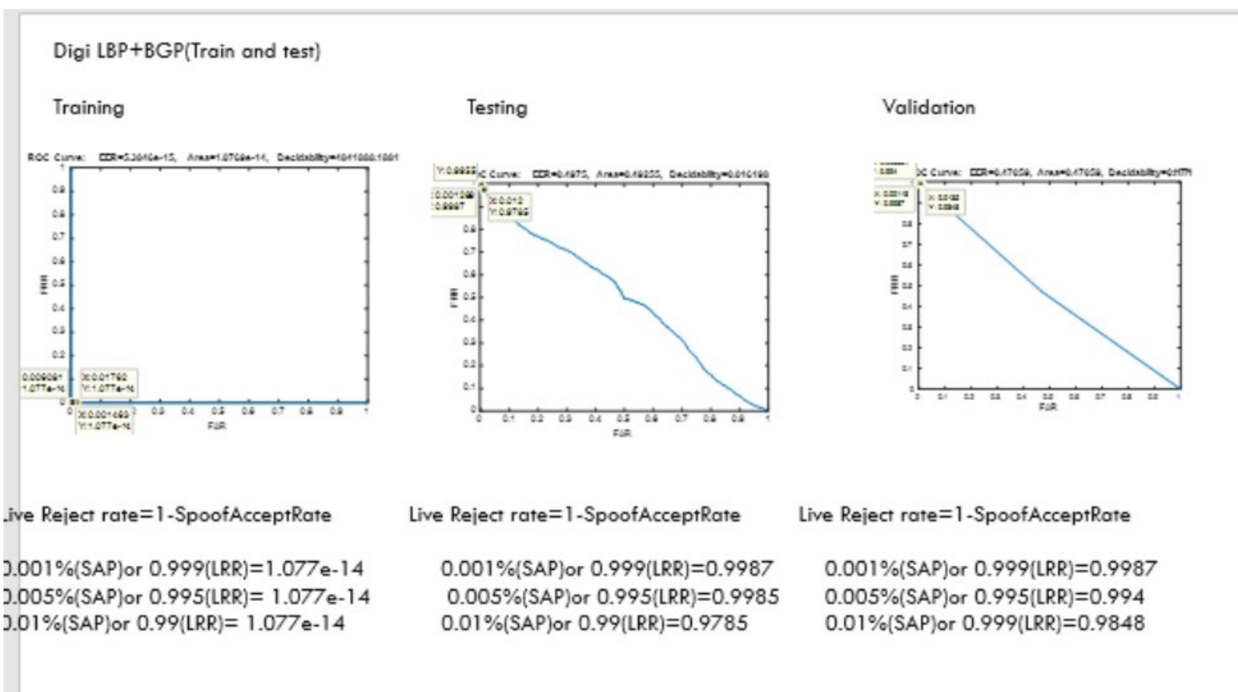

Fig13: Training and testing LBP+BGP on digital sensor

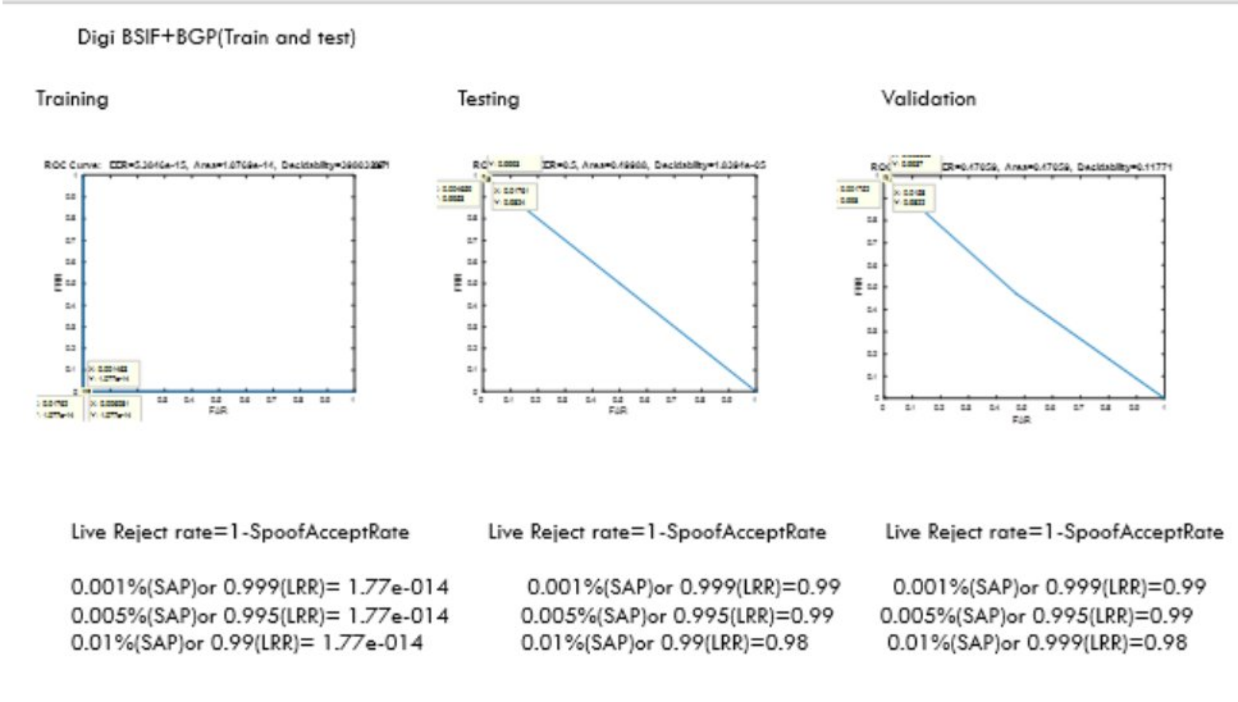

Fig14: Training and testing BSIF+BGP on digital sensor

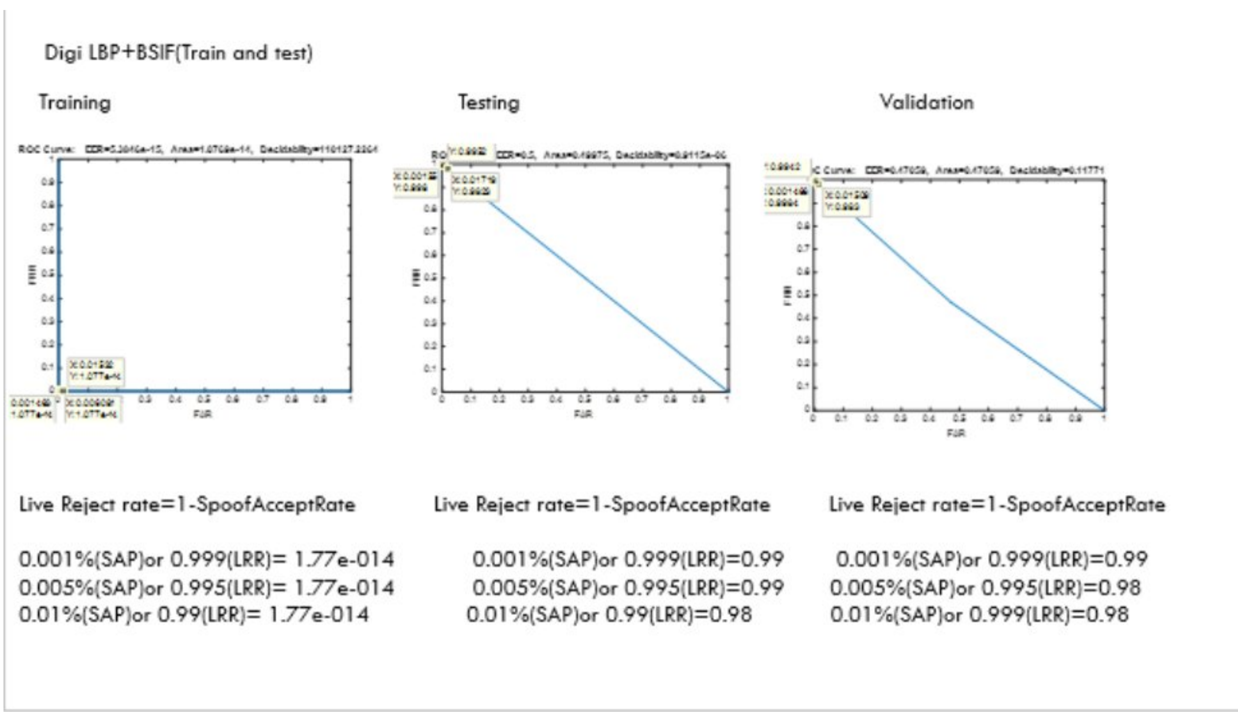

Fig15: Training and testing LBP+BSIF on digital sensor