A Comparative Analysis of Machine Learning Models for First-break Arrival Picking

Mohammed Ayub[1]
College of Computer Sciences and Engineering
King Fahd University of Petroleum and Minerals
Dhahran 31261, Saudi Arabia

SanLinn I. Kaka[2]
College of Petroleum Engineering and Geosciences
King Fahd University of Petroleum and Minerals
Dhahran 31261, Saudi Arabia

***Abstract*—First-break (FB) picking is an important and necessary step in seismic data processing and there is a need to develop precise and accurate auto-picking solutions. Our investigation in this study includes eight machine learning models. We use 1195 raw traces to extract several features and train for accurate picking and monitoring the performance of each model using well-defined evaluation metrics. Careful investigation of the scores shows that a single metric alone is not sufficient to evaluate the arrival picking models in real-time. Correlation analysis of predicted probabilities and predicted classes of machine learning models confirm that the performance metrics that use predicted probabilities have higher score value than those that use predicted classes. Our study which incorporates comparisons of different machine learning models based on different performance metrics, training time, and feature importance indicates that the approach we developed in this study is helpful and provides an opportunity to determine the real-time suitability of different methodologies for automatic FB arrival picking with clear deep insight. Based on performance scores, we bench-marked the Extra Tree classifier as the most efficient model for FB arrival picking with accuracy and F1-score above 95%.**

***Keywords*—First-break arrival picking; seismology; neural networks; machine learning; feature ranking**

I. Introduction

Detection of the first arrival from seismic phases plays an important role in solving many seismic exploration problems. In fact, picking the first arrivals is the first step in seismic data processing [1], [2]. However, the task is challenging due to the ever-increasing seismic data volume and therefore, manual picking is very time consuming and difficult for human experts. Moreover, in seismology, it is crucial to pick the First Break (FB) for many applications including imaging the subsurface, travel time tomography, understanding near-surface complexities, hydrocarbon and mineral exploration, microseismic monitoring of oil and gas-reservoir, and investigating the earth's crustal structure [3], [4]. Moreover, accurate FB picks help in inverting for a good near-surface velocity model in seismic processing. In fact, many algorithms of FB picking have been proposed including short-term average/long-term average algorithms [5], [6], auto regression with Akaike Information Criterion [7], higher-order statistics [8], [9]. Although these traditional automatic arrival picking algorithms are helpful for many applications and their performance cannot overtake that of manual picks, leaving them behind would be less useful for seismic imaging. Another problem with traditional methods is that they usually require a threshold, making them difficult to implement in complex seismic regions. On the other hand, manual or interactive FB picking methods can help improve performance in terms of quality and accuracy, requiring longer time and extensive effort, especially when the dataset is large. Due to the availability of huge seismic data and the inclusion of more difficult data acquisition areas, more robust, accurate and better automated first break picking techniques are essential for obtaining subsurface information from seismic data. To that end, the use of machine learning-based picking models bring a significant advantage in terms of cost and time.

Many research works have been in the literature for FB picking using machine learning including recently evolved deep machine learning. Unfortunately, there are many issues that need to be investigated. For example, comparative studies among the different models from the perspective of FB picking are not adequately explored. Different machine learning model exhibits different performance due to their underlying working principles. Again, one single performance metric cannot be used to bench mark the performance of the particular algorithm because scores of evaluation metric vary across the models. More than that, what needs to be emphasized for the efficiency of the FB picking model for real-time usability is an accurate prediction with minimal resources both in terms of time and budget. Hence, the FB picking model that optimize training time with acceptable performance scores of different established metrics needs to be explored. Consequently, we rank features using the automatic feature ranking method. To realize these objectives, we design, develop and evaluate eight machine learning algorithms in addition to three automatic feature selection techniques by which we search the features that reduce the training time, data acquisition and data processing costs.

Our approach involves the following steps to optimize FB picking: (1) Investigate FB picking by deploying eight machine learning models. We analyze the suitability of the same in terms of performance score and training time for real-time deployment of first break arrival picking on noisy and original seismic trace data by using five evaluation metrics. (2) Bench-mark the highest performance score of about 95% for accuracy and F1-score for Extra Tree. Besides, we extract and recommend the most common important features based on experimental results obtained by fitting three powerful ensemble classifiers on noisy data. (3) Correlate/evaluate

machine learning classifiers by means of predicted classes and predicted probabilities.

The rest of the paper is organized as follows: Section II describes the related works and Section III discusses the methods and materials used in this paper. Experiment details and results are given in Section IV followed by discussion and analysis in Section V. The conclusion and future work is given in Section VI.

II. Related Works

The reference [25] integrated traditional seismic methods and machine learning for picking. Geophysical techniques were first used for preliminary picking and then applied CNN to identify, remove and fix poor picks. In [26], arrival picking problem was studied by formulating it binary image segmentation problem. Arrival was picked using U-net architecture which is based on pixel-wise CNN. Like [26], the authors in [27] proposed FB picking models using deep learning technique. They deployed seven-layered U-Net architecture with skip connection. In [28], U-Net was used for segmentation of seismic image and Recurrent Neural Network (RNN), for arrival picking. Additionally, the authors proposed a simple weight adaptation method for generalization of the model in unseen data.

In SC-PSNET [29], the authors extend 3C seismograms processing with CNN to 1C seismic processing. Their study showed that CNN in combination with RNN is more promising for P- and S- detection when there are not enough training data available. To mitigate high intensive labour and thus high cost of manual seismic picking, the study [30] transferred the PhaseNet model and incorporate it with double-difference tomography. The results showed that the model's prediction was nearly as accurate as the result of a human expert with very low time and cost. The reference [31] proposed a Faster-RCNN based P-wave picking method using local window extracted from seismic waveform to enhance the accuracy of arrival picking. Faster-RCNN is an object localization algorithm based on Regions Proposal Network (RPN) commonly used to detect object of interest in the complex background.

In [10], Chen et al. investigated the automatic seismic waveform classification and arrival time picking using novel anti-noise Convolutional Neural Network (CNN) and K-means Clustering (KC) techniques. The authors used Mean Absolute (MA), Mean Square (MS), Short-Term Average Ratio (STAR) and Long-Term Average Ratio (LTAR) as features. Prior to this, the same first author of [10] in [11] studied FB picking with the same features in an unsupervised machine learning manner where it was showed that the method developed had much better performance than the traditional STA/LTA method in noisy data. In [12], Mezyk and Malinowski proposed a Multi-pattern FB picking method using Deep Neural Net-work (DNN), Support Vector Regression (SVR) and Extreme Gradient Boosting (XGBoost). The models were trained and tested using different features such as STA/LTA, entropy, and fractal dimension and a few others. Their experiment results showed that the DNN classifier outperformed SVR and XGBoost. Yuan et al. [13] adopted CNN for the classification of seismic waveforms, thereby locating FB using a threshold, first local minimum rule, and median filter in a sequential manner. The experiment results from synthetic and field data showed that the use of CNN using the time-space sub-image as inputs has efficient classification and picking capability.

Another convolutional image segmentation based FB picking was studied by Wu et al. [14]. Their idea was first to convert the microseismic trace into a 1D gray-scale image and pick the first arrival manually. Thereafter, based on that time index, the traces were labeled to train SegNet which was built based on encoder and decoder neural network concept. Similarly, PickNet was introduced in the work of [15] with the inspiration of the VGG-16 image recognition model to pick P and S arrival time. As an overall performance, the model could pick high-quality P and S wave arrival times in real datasets with potential generalization capabilities to other data collected using different seismic networks. Another seismic wave arrival time picking model (PhaseNet) was designed and tested by Zhu and Beroza [17] using a CNN. Their model was adapted from U-net which is a biomedical image processing framework built on Deep Neural Network (DNN).

Different from all the above works, the poor pick identification using a CNN was investigated in the work of [16] with cross-correlation of adjacent traces as the solution to fix the poor picks. P-wave arrival picking using vertical component and classification of first-polarity was explored in [18] by training two different CNNs for picking of P-wave arrival and first-polarity classification, respectively. The model's prediction was much more accurate than that of the analyst with the highest score of classification 95% in terms of precision. Gao et al. studied FB picking using fuzzy C-means, where they first utilize the vertical and horizontal sliding window to determine the first-arrival range and then Particle Swarm Optimization (PSO) to locate cluster centers [19]. Unlike all studies discussed above, the authors of [20] deployed Variational Auto-Encoder (VAE) and a Generative Adversarial Network (GAN) for automatic FB picking using seismic shot gather images as input. In their work [22], Duan and Zhang claimed that seismic traces are correlated with one another and the same is underutilized. They proposed a multi-trace multi-attribute analysis method for FB picking using a Support Vector Machine (SVM) to improve automatic picking. Another image segmentation-based FP picking as in [14] was explored in [21] who used 2D pixel-wise CNN. In their work, raw seismic images were first treated as gray scale images with normalized pixel values between 0 and 255. Then the resulted images were converted into binary images by tagging the pixels before arrival as zero and after arrival as 1. Though the model exhibited the highest accuracy of 96%, it was not suitable for smaller seismic traces. Hollander et al. [23] proposed a five-layered deep neural network composed of one convolutional layer, one pooling layer, one dense layer, and one output layer, for identification of the first break from a seismic trace. The model was trained on augmented data to classify the trace and thereby locating the first break by the use of maximal energy ratio. Transfer learning can save a significant amount of training time by enabling the reuse of the CNN model trained in other domains. The author in [24]

applied the idea for FB identification and arrival time picking using Continuous Wavelet Transform (CWT) as input features for AlexNet, GoogleNet and SqueezeNet. Though the models had superb performance compared to STA/LTA and Adaptive Multiband Picking Algorithm (AMPA), the accuracy was only about 90%.A summary of all related work is given in Table I. Note that most studies used CNN and the accuracy lies in the range of 78% and 98%.

TABLE I. SUMMARY OF ARRIVAL TIME PICKING METHODS

Problem, Ref.	Features	Algorithm	Datasets	Acc.
Waveform classification and arrival picking[10]	MA, MS, STA/LTA	KC, CNN	Synthetic and real data	98.6%
Event picking [11]	Mean, Power, STA/LTA	FC	Synthetic and real data	-
FB picking [12]	11 features of STA/LTA, entropy,	ANN, SVR, XGBoost	Real data	95%
	fractal dimension			
Waveform classification and FB picking [13]	time-space sub-image	CNN	Synthetic and real data	-
Semi-automatic FB picking[14]	1D seismic trace image	CNN	Synthetic and field data	
Arrival time picking[15]	3C Seismic waveform	CNN (PickNet)	Real-world data	78%
Poor pick identification[16]	Seismic record image	CNN	Real-world data	95%
Arrival time picking[17]	3C seismic waveform	DNN (PahseNet)	NCSN	89.6% (F1)
P-wave picking and First-motion classification[18]	1C seismic waveform	CNN	SCSN	95%(P)
First arrival picking [19]	Seismic trace	Fuzzy C-means	Field data	96.5%
First arrival picking [20]	Seismic image	Deep learning (VAE+GAN)	Field Data	-
First arrival picking [21]	Seismic image	CNN	Field data	96%
First-break picking[22]	advance, multi-trace correlation	SVM	pseudo-synthetic, real	2.4ms-14ms (RMS)
First-break identification[23]	energy ratio	CNN	Private	96%
FB arrival time identification[24]	CWT	AlexNet, GoogleNet, SqueezeNet	Real data	Above 90%
Poor pick identification, remove and fix [25]	Multitrace	CNN	Private	97.8%
Automated arrival picking [26]	Seismic image	CNN (U-Net)	Filed data	-
Automatic FB picking [27]	Seismic image	CNN (U-Net)	Seismic data	-
FB picking [28]	Seismic image	CNN (U-Net), RNN	Synthetic data	-

III. METHODS AND MATERIALS

A. Problem Formulation

We formulate FB picking as a binary classification problem; 1 (True) for the FB event and 0 (False) for the non-FB event. We explain the details of machine learning models used in this study with the metrics used to evaluate performance and automatic feature ranking techniques. An illustration of FB and Non-FB is given in Fig. 1.

B. Machine Learning Models

In this subsection, eight machine learning classifiers are experimented for the classification of first-break arrival picking. These include Feed-forward Neural Network (FNN), K-Nearest Neighbors (KNN), Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), Extra Trees (ET), Gradient Boosting Trees (GB), and XGB Classifier (XGB).

1) Feed-forward Neural Network (FNN): Feed-forward Neural networks are a set of neurons interconnected in the form of layers. The inputs are passed forward by multiplying with certain weights and adding the bias. The neurons in hidden layers and output layers are activated using different activation functions such as linear, tanh, sigmoid, softmax and many others. At the output layer, the error is calculated based on the actual label and is back-propagated to minimize it by the use of some mechanism called gradient descent algorithm. In this manner, all the input samples are trained until a specified number of epochs is reached. The architecture of FNN is designed by including two hidden layers of 32 neurons each, with kernel initializer from normal distribution and activation as ReLU [32], which help in tackling the gradient vanishing problem. As we have two classes to predict, a one-neuron dense layer with Sigmoid activation is put at the end of the architecture.

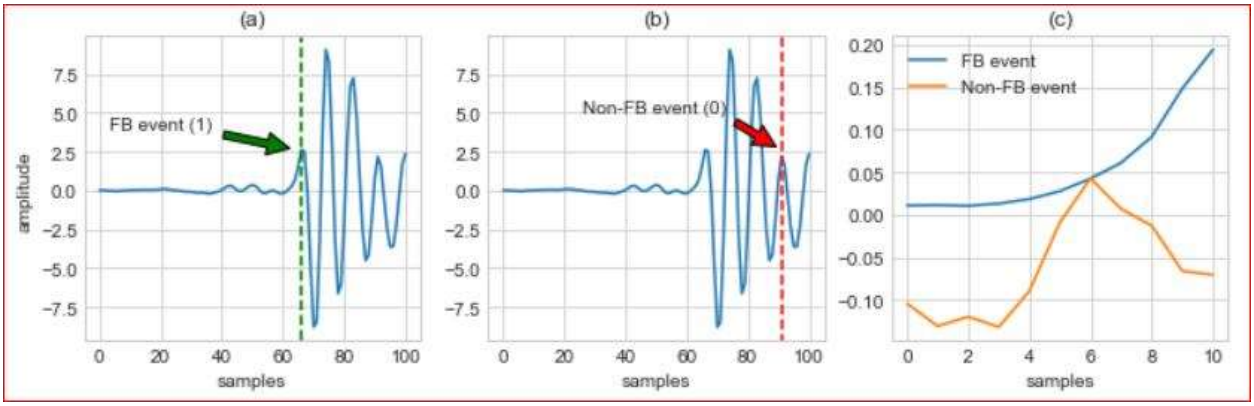

Fig. 1. FB and Non-FB Event; (a) Green Dashed-Line Showing First Break Event, (b) Red Dashed-Line Showing Non-First Break Event, and (c) Illustrative Signal Plot of Normalized Amplitude Values of both Events with 11 Samples.

2) K-Nearest Neighbor (KNN): KNN is a non-parametric algorithm that does not require model learning but makes a prediction on a lazy learning mode, meaning prediction is made just-in-time by calculating the distance between the inputs and training instance. Therefore, KNN requires almost all data while making predictions thereby making a memory burden. Using predefined k along with a new sample, the most common one with new a sample is chosen from the nearest k samples.

3) Logistic Regression (LR): The Logistic Regression (LR) is a probabilistic model that makes a prediction by training data on the logit function. It requires large sample sizes and independent variables need not be correlated with each other. For logistic regression, we tune hyper parameter using random search to get an optimized model which returned 'l2' regularization penalty against 'l1', 'none' class-weight against 'balanced' as the best fits for our problem of classification.

4) Decision Trees (DT): Decision Trees (DT) are flow chart-like diagrams with the terminal node representing the decision. The results of decision tree algorithms are easy to understand and calculate for the human expert and are computationally cheap as well. In DT, some relevant questions are asked at each node and based on how much information a feature provides for the class, the node is branched and this will continue until all the children nodes belong to the same class or the information gain is zero.

5) Random Forest (RF): Random Forest (RF) consists of a set of trees that are built by taking random training samples, where random subsets of features are used when splitting the nodes of decision trees. Then for prediction, the results of all trees are averaged with a technique call bagging, which is bootstrapping aggregating in long-form.

6) Extra Trees (ET): ET is extremely randomized trees and is the same as RF with differences; (1) It uses a random split of a tree, rather than best split as in RF, and (2) It builds multiple trees without bootstrapping, meaning with the replacement of samples. The maximum features considered to branch a given node is calculated based on the square root value of the total features.

7) Gradient Boosting Trees (GBT): GBT are a group of weak learners that are combined to make a stronger predictive model based on weighted minimization. In GBT, new trees are added to the model without manipulating the existing trees, and appending the result of the new tree to that of existing until the loss is minimized or predefined numbers of trees are reached. For our classification problem, GBT is trained on 100 trees with a maximum depth of 3 for each tree.

8) Extreme Gradient Boosting Tree (XGBT): Another boosting tree is designed based on the implementation of GBT but uses an accurate approximation to find fast and robust tree models. This is called XGBT that stands distinctly from other tree-based models with these two properties; (1) unlike other models that use the first-order derivatives of the loss function of the base model to minimize the overall error, XGBoosting finds the second-order derivatives of the loss function for better approximation, (2) It uses advanced regularization techniques such as L1 and L2.

C. Performance Metrics

Normally, we use accuracy for classification problems in machine learning. Sometimes, only the accuracy is typically not enough to evaluate a machine learning model. For example, in a dataset with a large class imbalance, the model will predict correctly the majority class and hence will have a high classification accuracy, which in practicality is misleading. In this case, additional evaluation metrics are required and some of the commonly used ones are explained in this subsection. In order to best explain and understand the metrics, a confusion matrix is shown in Fig. 2.

1) Precision (P): Precision is obtained by dividing the number of True Positives (TP) by the sum of the number of True Positives and False Positives (FP). It can also be called the Positive Predictive Value (PPV) and its mathematical expression is given as:

$$Precision\ (P) = \frac{TP}{TP+FP} \quad (1)$$

Precision can be regarded as an indicator of how exactly a classification algorithm will classify a true class as a true. From the equation, it can easily be seen that a low precision value means a large number of False Positives.

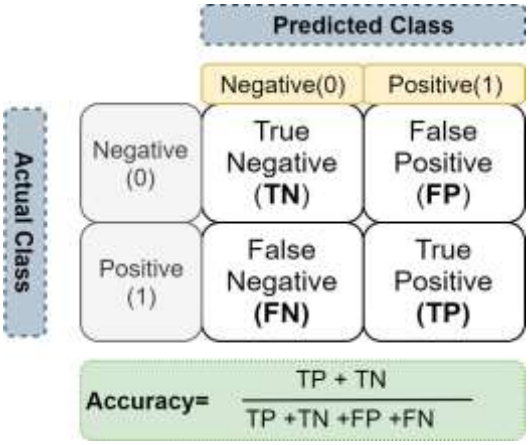

Fig. 2. Confusion Matrix. '1' Represents FB Event '0' Non-FB.

2) Recall (R): Recall is the ratio between the number of True Positives and the sum of the number of True Positives and the number of False Negatives (FN). In the literature, we can see it also as Sensitivity or the True Positive Rate. Its formula is given by:

$$Recall\ (R) = \frac{TP}{TP+FN} \tag{2}$$

Recall shows the completeness of a classification algorithm, whereas precision shows the exactness of a classification algorithm. A low value of recall testifies that there are many False Negatives.

3) F1-score (F1): The F1 Score is obtained when we divide the product of precision and recall with the sum of the same and again multiplied by 2. It is also termed as the F Score or the F Measure and it shows the balance between the precision and the recall.

$$F1 = 2\frac{P\,R}{P+R} \tag{3}$$

4) ROC Curve: It stands for Receiver Operating Characteristics (ROC) curve and is also called AUROC (Area Under the Receiver Operating Characteristics). It is used as one of the most important performance evaluation metrics for the classification model. Using the ROC curve, the performance of any classification model can be measured by setting the thresholds at various points. Mathematically, the ROC is a probability curve and the area under the curve indicates the degree or the measure of separability between classes. We can interpret the ROC as; the higher the AUC, the better the model. That is, the model is predicting True as True and False as False. We can plot the ROC curve putting True Positive Rate (TPR) on the y-axis and False Positive Rate (FPR) is on the x-axis. By analogy, we can say that the model performs better if the ROC is about to touch the left-top corner of the plot.

D. Feature Importance

Machine learning models are largely dependent on high-quality features. The inclusion of irrelevant or less correlated features not only degrades the model performance but also wastes the computational resources, training time and cost. Therefore, the selection of highly important features contributes towards better performance of the machine learning model. From the perspective of seismic data processing in which a large volume of data is overwhelming due to the availability of high data acquisition technology, the training time of the model for real-time deployment is a crucial factor that is drawing serious attention from business owners and researchers alike. Thus, prioritizing the features or removing the less contributing features is the best acceptable choice among the groups. As such, we deploy the Recursive Feature Elimination (RFE) method in combination with three powerful ensemble estimators such as Random Forest, Extra Trees and Gradient Boosting, one at a time.

IV. Experiments and Results

A. Dataset and Data Preparation

The dataset used in this study is from the published work of Mezyk and Malinowski [12] and is available in Github repository[1]. The data preparation steps and extracted features are shown in Fig. 3. For the machine learning model to generalize well, a sufficient number of input samples are required to be trained. Original raw traces of 1195 are perturbed using the Gaussian method to generate more noisy traces. Afterwards, feature matrices are constructed by extracting a total of nine features and appending label '1' for FB and '0' for Non-FB. The extracted features are: (1) raw trace amplitudes, (2) gradient of the absolute trace amplitudes, (3) trace entropy, (4) gradient of the trace entropy, (5) fractal dimension of the trace, (6) gradient of fractal dimension, (7) STA/LTA of the trace, (8) sum of the amplitude spectra of the trace, (9) gradient of the sum of the amplitude spectra. And finally we have a training dataset of 289190 instances that is balanced with true and false first-break events. From the whole noisy dataset, 25% is allotted for validation to measure the learning validity of the model while we use original traces of 5300 for testing purposes in order to check the generalization capability of the learned model.

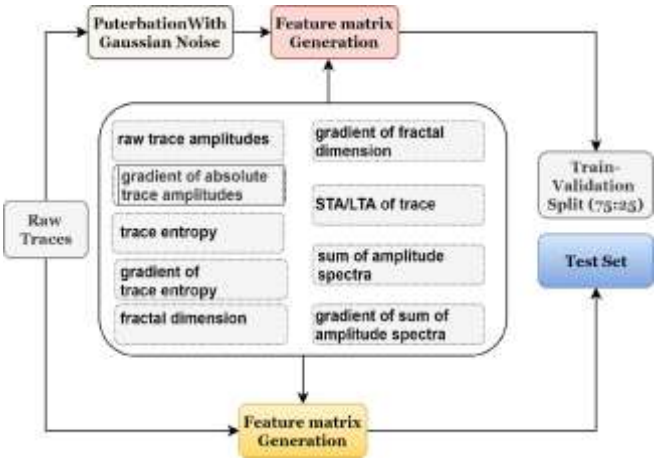

Fig. 3. Data Preparation Pipeline. Train and Validation Sets are Noisy Data. The Test Set Contains Purely Original Traces to Challenge the Generalization Capability of the Models.

[1] https://github.com/mmezyk/fbpicker/tree/master/data

B. Experiment Setup

We use Python 3.7.3 (Open source programming with the largest community) and Scikit-learn (Sklearn) machine learning library to implement this work on Keras [41] that use Tensor flow as the back-end. To realize this, we used all-in-one machine learning software package Anaconda 1.9.12 configured in an NVIDIA GEOFORCE GTX 950 GPU and four-core i7 6th generation CPU machine with total graphic memory of 16GB and RAM of 16GB. For all experiments, we used Adam and binary cross-entropy as optimizer and loss function, respectively. The experiment results are reported using five best and commonly used metrics such as accuracy, precision, recall, F1-score and ROC score as mentioned in Section III-C. Besides, we also take into account the training time required for each model so that comparative and feasibility analysis can be done.

C. Experiment Results

1) FNN: The model is compiled using binary cross-entropy [33] as loss function and Adam [34] (Kingma and Ba, 2014) as the optimizer. The model is trained for 200 epochs with a batch size of 32 on 25% of validation data. The loss, and ROC curves are shown in Fig. 4. From Table II, it is seen that it has a score of accuracy 92.64%, precision 89.18%, recall 97.06% and F1-score 92.95%. From the precision and recall scores it is observed that the neural network model predicts more non-FB events as FB events than first breaks as non-first breaks. This can be clearer we if we analyze the confusion matrix, where there are a total of 312 non-FB events that are predicted as FB (False Positive) in comparison to 78 FB that are predicted as non-FB events (False Negative).

2) KNN, LR and DT: We trained KNN keeping the number of neighbors as 3 and, the performance scores as shown in Table II are accuracy 91.68%, F1-score 91.95% and ROC AUC score 96.17%. LR has a precision of 80.55%, recall of 92.38% and accuracy of 85.04%. The rate at which it classifies true first break and false break is significantly lower than KNN, and hence the higher numbers of False Positives and False Negatives. By the DT model, the scores achieved are 89.79%, 92.00% and 90.01, for accuracy, recall and F1-score, respectively. Moreover, False Negatives and False Positives are higher than KNN with lower False Positives than LR . In DT, the quality of the node split is monitored using the Gini impurity measure. Branching of the node is allowed till all leaves becomes pure.

3) RF, ET, GBT and XGBT: The hyper parameter tuning using cross-validation random search recommends the deployment of random samples while building each tree and entropy as information gain. It is also observed that the best result is achieved when nodes are split with a minimum number of samples of 9 based on 7 features maximum with having a minimum of 2 samples at the leaf node. The F1-score obtained is 93.39%, with precision and recall 91.19% and 95.70%, respectively. All scores can be seen in Table II.

The maximum features considered to branch a given node in ET is calculated based on the square root value of the total features. In our case, we have nine features and each node is split with 3 features maximum. The branching of the node will stop when all leaves have a number of samples less than 2. The extra tree classifier achieved a precision of 93.07%, accuracy of 95.26% and F1-score of 95.38%.

GBT is trained on 100 trees with a maximum depth of 3 for each tree. Different from other classifiers above, we use Friedman's Mean Squared Error (MSE) to monitor the quality of a split. We have a ROC AUC score of 98.47%, an accuracy of 92.75% and F1-score of 92.94%. XGBT is trained using the maximum depth of each tree with 6 and the precision score is 91.16%. The accuracy and F1-scores are 93.23% and 93.29%, respectively. The ROC curves for all shallow machine learning models are shown in Fig. 5.

4) Feature Importance: For feature ranking, we experimented with three estimators using the RFE method. We use an advanced method of REF where the RF estimator is trained using the cross-validation method of StratifiedKFold with a 10 split. We use 100 trees with the replacement of samples while building trees and Giniimpurity is used to calculate information value. For the ET estimator, we use the same parameter as Random Forest except bootstrap equals to false, meaning the subset of samples used for building one tree are not replaced while building subsequent trees. GB Estimator is trained with 100 trees as in previous sections on stratified 10-fold cross-validation of RFE. For calculating the information value, the Friedman Mean Square Error (FMSE) is used. We keep the learning rate 0.1 with the maximum depth of the tree 3, and loss function as deviance.

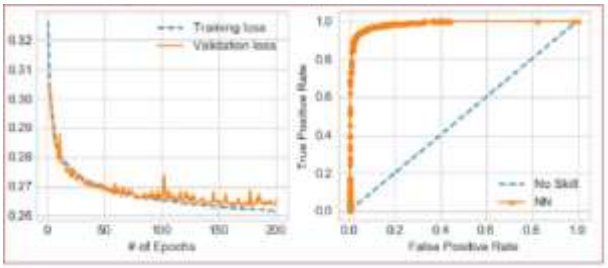

Fig. 4. The Loss and ROC Curves of FNN. The Learning behavior of the Model is good with Little Peak Fluctuation throughout the Epochs.

TABLE II. PERFORMANCE SCORES OF ALL MODELS

Model	Acc.	Pr	Re	F1	ROC	Time (sec.)
FNN	92.64	89.18	97.06	92.95	98.64	1661
DT	89.79	88.11	92.00	90.01	89.79	29
KNN	91.68	89.07	95.02	91.95	96.17	22
LR	85.04	80.55	92.38	86.06	95.23	10
RF	93.23	91.19	95.70	93.39	98.41	160
ET	95.26	93.07	97.81	95.38	99.23	192
GBT	92.75	90.58	95.43	92.94	98.47	534
XGBT	93.23	91.16	95.74	93.39	98.53	163

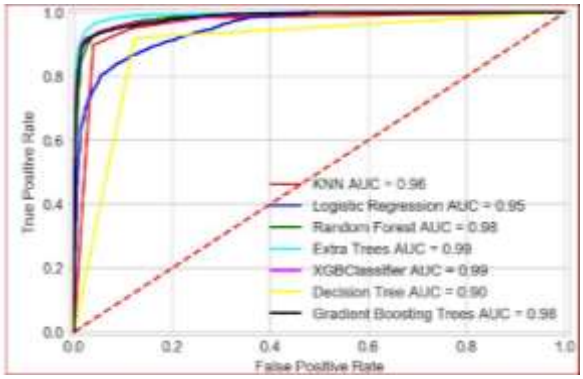

Fig. 5. ROCs of Machine Learning Models. The ET has the Highest Score.

V. Discussion and Analysis

A. Machine Learning Models

We have evaluated eight different deep machine learning models for picking FB and the performance behavior is monitored using five evaluation metrics. Understanding the real-time deployment and resource requirements for training machine learning models, the training times for each models are also recorded. The FNN has a recall of 97.06%, which is the second-highest in the same category and indicates that it predicts the actual first break event as a non-first break more than the other model does. The training time of machine learning depends on many factors such as the size of the dataset, the number of features used, number of epochs, and many other factors. Furthermore, choosing a larger batch size can make the training faster but with comparatively poor performance.

In general, compared to deep learning models, shallow machine learning models are computationally less sophisticated with less training time and memory requirements than deep learning models. Moreover, traditional machine learning models have limited performance when the data volumes are extremely large. Nonetheless, they can be effectively deployed for real-world problems with the careful granular tuning of the model parameters. In this work, as seen in Table II and Fig. 6, LR and DT have the lowest and second-lowest classification performance in terms of all metrics, respectively. In terms of recall and ROC, LR performs better than DT with scores of 92.38% and 95.23%, respectively. If we scan the scores carefully, we observe that the Extra Tree (ET) performs best in terms of all metrics with an accuracy of 95.25% and ROC 99.23%. GBT has the longest training time with 534 seconds followed by Extra Trees with 192 seconds. LR takes the least amount of training time of 10 seconds among all classifiers. This trend is noticed from the values given in the last column of Table II and Fig. 7.

From the different classifiers we evaluated in this work, Extra Tree clearly outperformed other models in terms of all evaluation metrics. The second highest performers are RF and XGBT with very similar scores if we consider the accuracy, precision and F1-score as evaluation criteria. In terms of recall and ROC, FNN is the second-best performer. If, from all classifiers, the suitable models with less training time and better accuracy are to choose for real-time deployment, ET, RF and XGBT are the perfect choice because they all need reasonable training time, below 200 seconds. Therefore, for FB arrival classification, traditional machine learning is enough for real-time deployment if ready-made pre-calculated features are to be used, that is if only trace amplitude-based features instead of the seismic image are to be deployed as features.

B. Feature Importance

1) RFE using Random Forest Estimator: REF removes less important features in an iterative fashion. Feature importance is calculated as the coefficient of some estimators such as Random Forest. From the experimental results it is noticed that the highest performance is obtained when all nine features are used. But this method gives STA/LTA, the traditional method, the highest importance and the gradient of trace amplitude (g trc amp) the least importance. The other best five features with decreasing importance are the trace amplitude (trc amp), fractal dimension of the trace (fdm), the gradient of fractal dimension (g fdm), the sum of the amplitude spectra (sum amp spec) of the trace and trace entropy (trc ent). This trend can be seen in Fig. 8.

2) RFE using Extra Trees Estimator: As in RF, the best six features according to their importance are the STA/LTA, the fractal dimension of the trace, the trace amplitude, gradient of fractal dimension, trace entropy, and the sum of the amplitude spectra of the trace. As seen in Fig. 9, the fractal dimension of the trace, which is in third place in RF, has now become the second important feature in the Extra Trees estimator. This is due to their handling of input samples while training the different trees.

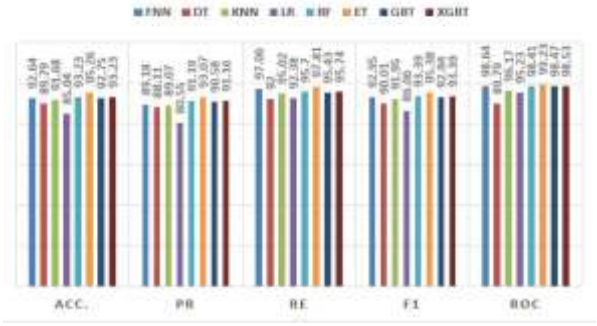

Fig. 6. Performance Scores of Machine Learning Models.

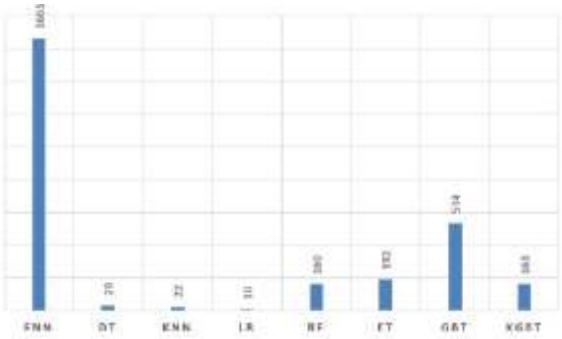

Fig. 7. Training Time (in Seconds) for Machine Learning Models.

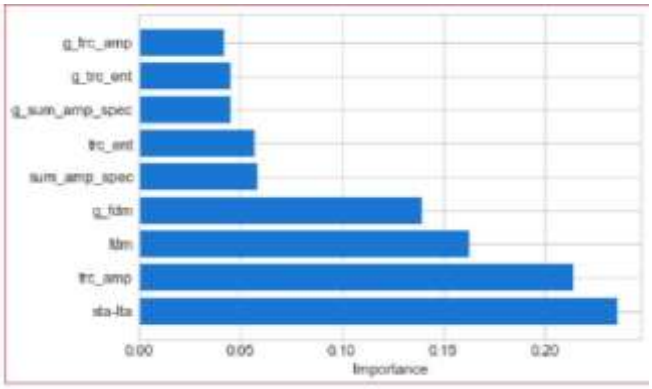

Fig. 8. Feature Ranking using RF Estimator.

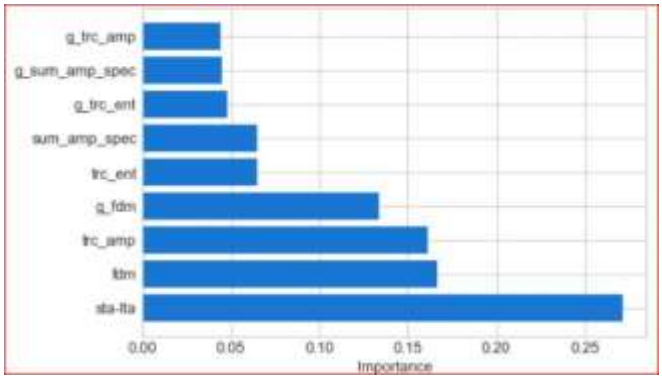

Fig. 9. Feature Ranking using ET Estimator.

3) RFE using Gradient Boosting Estimator: Almost like earlier estimators, this estimator selects STA/LTA, the amplitude of the trace, the fractal dimension of the trace, and gradient of fractal dimension as four most significant features as seen in Fig. 10.

From the above experiment results, it is apparent that the STA/LTA, the amplitude of the trace, the fractal dimension of the trace, and gradient of fractal dimension are the most important common features for the classification of FB picking.

C. Prediction Correlation

The machine learning models used in this study provide a prediction for both class labels (i.e. 0 or 1) and probabilities by using a prediction function. Direct class labels are used to calculate accuracy, precision, recall and F1 score, and probabilities are used for calculating AUC, ROCAUC, MSE, MAE, RMSE and many others. In terms of class label prediction, Gradient Boosting Trees and Extreme Gradient Boosting trees have the highest positive correlation with 0.95. The second highest correlation pairs with 0.91 and 0.90 are ET and RF, ET and XGBT, respectively. This can be seen in Fig. 11. If we compare the correlation in terms of prediction probability, we see that almost all classifiers are correlated with values greater than 0.90. This is the reason that ROC scores are higher than those of accuracy and F1 measures. Moreover, we also observed that the correlation of the decision tree and logistic regression with other classifiers are less than that of others in both cases, confirming that both models are not suitable enough to handle first-break arrival picking efficiently, compared to other models. Furthermore, as seen in Fig. 12, XGB and GBT have a strong correlation between them with a coefficient of 1.

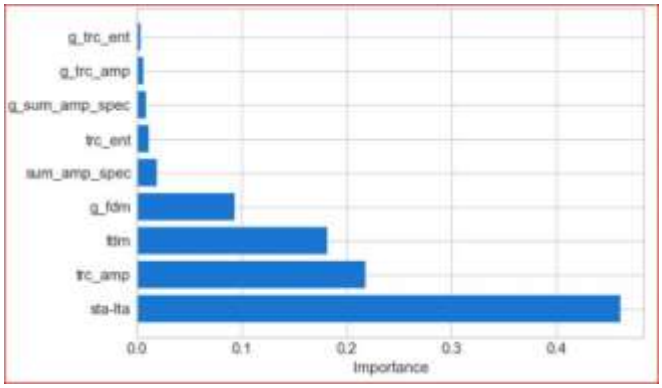

Fig. 10. Feature Ranking using Gradient Boosting Estimator.

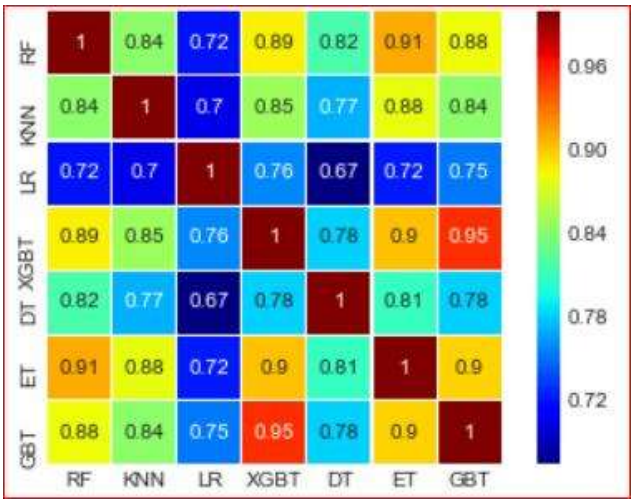

Fig. 11. Pearson Correlation of Machine Learning Classifiers used in this Paper. The Correlation between each Pair of the Classifier is based on Predicted Class (1 for FB Event, 0 for non-FB Event).

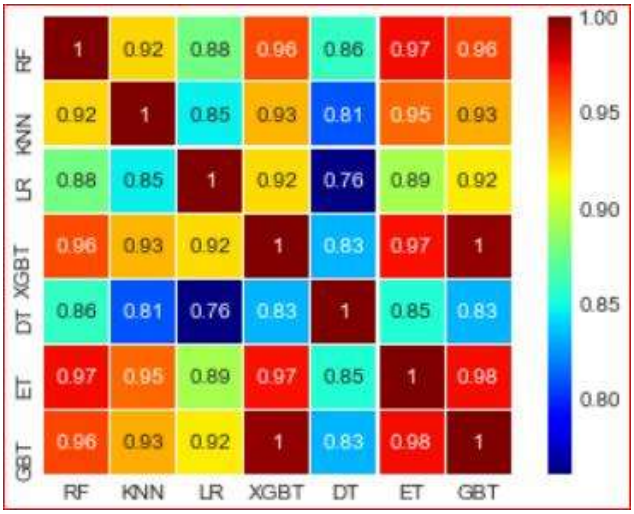

Fig. 12. Pearson Correlation of Machine Learning Classifiers used in this Paper. The Correlation between each Pair of the Classifier is based on Predicted Probability. (1 for FB Event, 0 for Non-FB Event).

VI. Conclusion and Future Work

A. Conclusion

In this study, we have explored and investigated FB arrival picking by formulating it as a binary classification problem. In that vein, we deploy eight machine learning models. The models are trained on noisy data generated applying Gaussian perturbation using nine features and tested on original data for generalization capability. The models architectures are fine-tuned in line with the first break arrival picking problem by undergoing rigorous experiments. Extra Tree has the highest accuracy of 95.26% and F1-score of 95.38%. All the top performers have acceptable training time and show suitability for the real-time automatic deployment of first break picking. Our deployment of an RFE on nine features using three ensemble classifiers suggests four common important features: the STA/LTA, the amplitude of the trace, the fractal dimension of the trace and gradient of the fractal dimension. Careful investigation of the performance scores proves that a single metric alone is not sufficient to evaluate the FB picking models. As such, other types of measures such as precision, recall and F1-scores are required to further validate the performance of the model. In line with this, we noticed that the use of precision and recall can help experts in obtaining deeper insight into the classification behavior thereby allowing better real-time decisions.

B. Limitations and Future Works

In this paper, we consider only single sample features derived from a single trace using machine learning techniques. Though traditional machine learning techniques are less complicated and require less training time, their performance suffers from degradation when a huge volume of data is involved. Another limitation is that the models are compared using a single dataset. Therefore, as future work, we want to investigate FB picking using features derived from multiple samples on different datasets by deploying hybrid deep learning models.

Acknowledgment

The authors gratefully acknowledge the support provided by King Fahd University of Petroleum and Minerals. This study was supported by the College of Petroleum Engineering and Geosciences start-up grant.

References

[1] J. L. Hardebeck and P. M. Shearer, “A new method for determining first-motion focal mechanisms,” Bulletin of the Seismological Society of America, vol. 92, no. 6, pp. 2264–2276, 2002.

[2] P. L. Nelson and S. P. Grand, “Lower-mantle plume beneath the yellowstone hotspot revealed by core waves,” Nature Geoscience, vol. 11, no. 4, pp. 280–284, 2018.

[3] J. Zhang and N. Toksoes, “Monte carlo sampling of solutions to inverse problems,” Geophysics, vol. 63, pp. 1726–1737, 1998.

[4] X. Zhu, D. P. Sixta, and B. G. Angstman, “Tomostatics: Turning-ray tomography+ static corrections,” The Leading Edge, vol. 11, no. 12, pp. 15–23, 1992.

[5] R. V. Allen, “Automatic earthquake recognition and timing from single traces,” Bulletin of the Seismological Society of America, vol. 68, no. 5, pp. 1521–1532, 1978.

[6] P. R. Stevenson, “Microearthquakes at flathead lake, montana: A study using automatic earthquake processing,” Bulletin of the Seismological Society of America, vol. 66, no. 1, pp. 61–80, 1976.

[7] R. Sleeman and T. Van Eck, “Robust automatic p-phase picking: an on-line implementation in the analysis of broadband seismogram recordings,” Physics of the earth and planetary interiors, vol. 113, no. 1-4, pp. 265–275, 1999.

[8] Z. Ross, M. White, F. Vernon, and Y. Ben-Zion, “An improved algorithm for real-time s-wave picking with application to the (augmented) anza network in southern california,” Bulletin of the Seismological Society of America, vol. 106, no. 5, pp. 2013–2022, 2016.

[9] C. D. Saragiotis, L. J. Hadjileontiadis, and S. M. Panas, “Pai-s/k: A robust automatic seismic p phase arrival identification scheme,” IEEE Transactions on Geoscience and Remote Sensing, vol. 40, no. 6, pp. 1395–1404, 2002.

[10] Y. Chen, G. Zhang, M. Bai, S. Zu, Z. Guan, and M. Zhang, “Automatic waveform classification and arrival picking based on convolutional neural network,” Earth and Space Science, vol. 6, pp. 1244–1261, 2019.

[11] Y. Chen, “Automatic microseismic event picking via unsupervised machine learning,” Geophysical Journal International, vol. 212, no. 1, pp. 88–102, 2017.

[12] M. Mezyk˙ and M. Malinowski, “Multi-pattern algorithm for first-break picking employing an open-source machine learning libraries,” Journal of Applied Geophysics, vol. 170, pp. 103 848—103 860, 2019.

[13] S. Yuan, J. Liu, S. Wang, T. Wang, and P. Shi, “Seismic waveform classification and first-break picking using convolution neural networks,” IEEE Geoscience and Remote Sensing Letters, vol. 15, no. 2, pp. 272–276, 2018.

[14] H. Wu, B. Zhang, F. Li, and N. Liu, “Semiautomatic first-arrival picking of microseismic events by using the pixel-wise convolutional image segmentation method,” Geophysics, vol. 84, no. 3, pp. V143–V155, 2019.

[15] J. Wang, Z. Xiao, C. Liu, D. Zhao, and Z. Yao, “Deep-learning for picking seismic arrival times,” Journal of Geophysical Research: Solid Earth, vol. 124, pp. 6612–6624, 2019.

[16] X. Duan, J. Zhang, Z. Liu, S. Liu, Z. Chen, and W. Li, “Integrating seismic first-break picking methods with a machine learning approach,” in SEG Technical Program Expanded Abstracts 2018. Society of Exploration Geophysicists, 2018, pp. 2186–2190.

[17] W. Zhu and G. C. Beroza, “Phasenet: a deep-neural-network-based seismic arrival-time picking method,” Geophysical Journal International, vol. 216, no. 1, pp. 261–273, 2018.

[18] Z. E. Ross, M.-A. Meier, and E. Hauksson, “P wave arrival picking and first-motion polarity determination with deep learning,” Journal of Geophysical Research: Solid Earth, vol. 123, no. 6, pp. 5120–5129, 2018.

[19] L. Gao, Z.-y. Jiang, and F. Min, “First-arrival travel times picking through sliding windows and fuzzy c-means,” Mathematics, vol. 7, no. 3, p. 221, 2019.

[20] K. C. Tsai, W. Hu, X. Wu, J. Chen, and Z. Han, “Automatic first arrival picking via deep learning with human interactive learning,” IEEE Transactions on Geoscience and Remote Sensing, 2019.

[21] Y. Ma, S. Cao, J. W. Rector, and Z. Zhang, “Automatic first arrival picking for borehole seismic data using a pixel-level network,” in SEG Technical Program Expanded Abstracts 2019. Society of Exploration Geophysicists, 2019, pp. 2463–2467.

[22] X. Duan and J. Zhang, “Multi-trace and multi-attribute analysis for first-break picking with the support vector machine,” in SEG Technical Program Expanded Abstracts 2019. Society of Exploration Geophysicists, 2019, pp. 2559–2563.

[23] Y. Hollander, A. Merouane, and O. Yilmaz, “Using a deep convolutional neural network to enhance the accuracy of first-break picking,” in SEG Technical Program Expanded Abstracts 2018. Society of Exploration Geophysicists, 2018, pp. 4628–4632.

[24] X. Liao, J. Cao, J. Hu, J. You, X. Jiang, and Z. Liu, “First arrival time identification using transfer learning with continuous wavelet transform feature images,” IEEE Geoscience and Remote Sensing Letters, 2019.

[25] X. Duan and J. Zhang, "Multitrace first-break picking using an integrated seismic and machine learning method," Geophysics, vol. 85, no. 4, pp. WA269–WA277, 2020.

[26] Y. Ma, S. Cao, J. W. Rector, and Z. Zhang, "Automated arrival-time picking using a pixel-level network," Geophysics, vol. 85, no. 5, pp. V415–V423, 2020.

[27] C. Fernhout, P. Zwartjes, and J. Yoo, "Automatic first break picking with deep learning," IOSR Journal of Applied Geology and Geophysics, vol. 8, no. 5, pp. 24–36, 2020.

[28] P. Yuan, S. Wang, W. Hu, X. Wu, J. Chen, and H. Van Nguyen, "A robust first-arrival picking workflow using convolutional and recurrent neural networks," Geophysics, vol. 85, no. 5, pp. U109–U119, 2020.

[29] J. Zheng, J. M. Harris, D. Li, and B. Al-Rumaih, "Sc-psnet: A deep neural network for automatic p-and s-phase detection and arrival-time picker using single component recordings," Geophysics, vol. 85, no. 4, pp. 1–64, 2020.

[30] C. Chai, M. Maceira, H. J. Santos-Villalobos, S. V. Venkatakrish-nan, M. Schoenball, W. Zhu, G. C. Beroza, C. Thurber, and E. C. Team, "Using a deep neural network and transfer learning to bridge scales for seismic phase picking," Geophysical Research Letters, p. e2020GL088651, 2020.

[31] Z. He, P. Peng, L. Wang, and Y. Jiang, "Enhancing seismic p-wave arrival picking by target-oriented detection of the local windows using faster-rcnn," IEEE Access, vol. 8, pp. 141 733–141 747, 2020.

[32] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," arXiv preprint arXiv:1505.00853, 2015.

[33] S. Mannor, D. Peleg, and R. Rubinstein, "The cross entropy method for classification," in Proceedings of the 22nd international conference on Machine learning, 2005, pp. 561–568.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.