Internship Report

Early Classification of Time Series

Saurabh Kumar Singh
Tezpur University
saurabh.official.address@gmail.com

May 13, 2022

Abstract

Early Classification of Time Series involves classification of Time series early without observing complete timestamps, focusing on earliness while not severely compromising accuracy. Early classification of time series has multiple applications in Medical Diagnostic, Quality Monitoring, Human Activity Classification and Industrial process Monitoring. In recent years, ECTS problem have been researched extensively using different approaches like prefix based, shapelet based, model based etc. We propose a novel ECTS method that handles the Early classification problem in two parts, part one extracts features from the data using a LSTM and a CNN block, part two makes use of a Reinforcement Learning Agent to determine when to stop and produce a prediction. We evaluate LCR model using Image, Sensor, ECG, Motion, Device, Spectro type datasets from UCR time-series dataset archive. Results show that our method Consistently outperforms state-of-the-art alternatives in both accuracy and earliness. *Keywords*: Long Short Term Memory, Convolutional Neural Network, Recurrent Neural Network, Reinforcement Learning.

1 Introduction

Classification of time series has been a prominent problem in temporal data analysis. A large number of TSC algorithms have emerged offering cutting-edge efficient solutions. TSC problem has been approaches in various ways, Authors in [2, 5] have approached it using deep learning models, recently published State-of-the-art algorithm(HIVE-COTE 2021) [6] forms its ensemble from classifiers of multiple domains, including phase-independent shapelets, bag-of-words based dictionaries and phase-dependent intervals.

In year 2002, Rodr´ıguez Diez and Alonso Gonz´alez [4] first introduced the problem of Early Classification of time series, since then ETSC has been gaining popularity. Time sensitive applications like medical diagnostic get benefited by early classification. Accuracy and Earliness are inversely related, therefore managing trade-off between them is one of the primary problem that needs to be addressed. In time sensitive application early classification plays a key role, managing the appropriate earliness of classification differs in different applications. Medical diagnostics often are time critical and also require highly accurate classification, there is a huge amount of medical data available for research, which can be used to train models for early classification.

1.1 Problem Statement

The ECTS (Early classification of time series) problem is classification of time series at early stages of observing the time series, i.e. to select a time stamp in the time series at which a accurate prediction can be made based the information observed till that time stamp without having to observe complete timestamps of the time series. The earliness of prediction is often inversely related to accuracy, balancing earliness and accuracy trade-off makes this problem challenging. While a balanced trade-off could work for one application, it may not work for others, hence a robust solution is required which can adapt in accord with different applications.

1.2 Motivation for ECTS

In data mining and machine learning, early classification of time series has received significant attention as it can solve time-critical problems in many areas including medical, industry, and transportation. Literature indicates numerous applications of early classification of time series. Primary motivation of the work is to develop the early classification approaches for medical diagnosis of the diseases such as asthma, viral infection, abnormal Electrocardiogram (ECG), etc. Early diagnosis of these diseases can significantly minimize the consequences on patient health and assist the doctors in treatment.

1.3 Proposed Method

We proposed a novel ECTS method that handles the Early classification problem in two parts, Part one extracts features and another part makes use of a Reinforcement Learning Agent to determine whether enough information have been seen to output a predictions. The proposed LCR (LSTM-CNN-RL) model uses a LSTM block and a FCN block to extract features from the input data at each time stamps, the extracted features are then passed through a RL Agent which decides based on these input features whether or not to halt and predict a label. The Agent solves a Partially-Observable Markov Decision Process (POMDP) in which the features extracted by the LSTM-FCN block is recieved at each timestamp, depending on the agent's learned policy an action is sampled and based on the sampled action's quality a reward is observed. Agent's job is to maximize its reward based on accuracy and earliness of prediction.

1.4 Contributions

• We design a novel ECTS method that handles the Early classification problem in two parts, Part one extracts features from the data using LSTM and CNN, another part makes use of a Reinforcement Learning Agent to determine whether enough information have been seen to output a predictions.
• We propose the first solution that combines LSTM,CNN and RL to achieve earliness and accuracy goals by combining them into the objective function for one model.
• We evaluate LCR model using 7 different datasets from UCR time-series dataset archive. Results show that our method significantly outperforms state-of-the-art alternatives in both accuracy and earliness

2 Related Work

Early classification of time series is important when data becomes available over time and decisions need to be taken as early as possible. It addresses two conflicting goals: maximizing accuracy typically reduces earliness and visa-versa. Early Classification on Time Series (ECTS) (Xing et al. 2012)[10] is one of the first papers to introduce the problem. The authors adopt a 1- nearest neighbour (1-NN) approach and introduce the concept of minimum prediction length (MPL) in combination with clustering. Time series with the same 1-NN are clustered. The optimal prefix length for each cluster is obtained by analysing the stability of the 1-NN decision for increasing time stamps. Only those clusters with stable and accurate offsets are kept. To give a prediction for an unlabelled TS, the 1-NN is searched among clusters. Reliable Early Classification (RelClass) (Parrish et al. 2013)[8] presents a method based on quadratic discriminant analysis (QDA). A reliability score is defined as the probability that the predicted class for the truncated and the whole time series will be the same. At each time stamp, RelClass then checks if the reliability is higher than a user-defined threshold. In Early Classification of Time Series based on Discriminating Classes Over Time (ECDIRE) (Mori et al. 2017b)[7] classifiers are trained at certain time stamps, i.e. at percentages of the full time series length. It learns a safe time stamp (the start time) as the fraction of the time series which states that a prediction is safe. Furthermore, a reliability threshold is learned using the difference between the two highest class probabilities. Only predictions passing this threshold after the safe time stamp are chosen. The idea of EDSC (Xing et al. 2011)[11] is to learn Shapelets that appear early in the time series, and that discriminate between classes as early as possible. In (Mori et al. 2017a) early classification is approached as an optimization problem. The authors combine a set of probabilistic classifiers with a stopping rule that is optimized using a cost function on earliness and accuracy.

3 Methodology

The proposed methodology for early classification of time series are of two types first, we tried to tackle the problem using fixed approaches, we preset the size of timestamps at which the classification will be made. Later we developed a online approach with help of a reinforcement learning agent which has no preset classification, the agent decide based the information available at each timestamp whether or not to halt and predict a classification.

We divide the problem in two smaller problems, classification and earliness. Classification of time series is a problem which has been researched for decades, many effective and efficient works have been published by different authors, Fazle Karim [5] proposed a multivariate time series classification model by augmenting the fully convolutional block with a squeeze-and-excitation block, we used this model for classification. Earliness refers to classification of Time series early without observing complete timestamps, we used different methods and produced effective models that can classify a time series early while also focusing on accuracy. Four different model have been proposed by us, first three models are fixed approaches, while the last model introduces an online approach.

3.1 MLSTM-FCN 20-Point Search

Model uses a fixed approach, presetting the input dataset timestamp length by dividing it into twenty parts, such that 20 different timestamps size datasets may be selected to output classification before observing complete timestamps size. The proposed method uses a LSTM block and a FCN block to extract features from the input data at each time stamps, the extracted features from both the blocks are concatenated and then passed through a SoftMax layer to produce final prediction of labels. Classification Accuracy of 20 different timestamp size models are obtained.

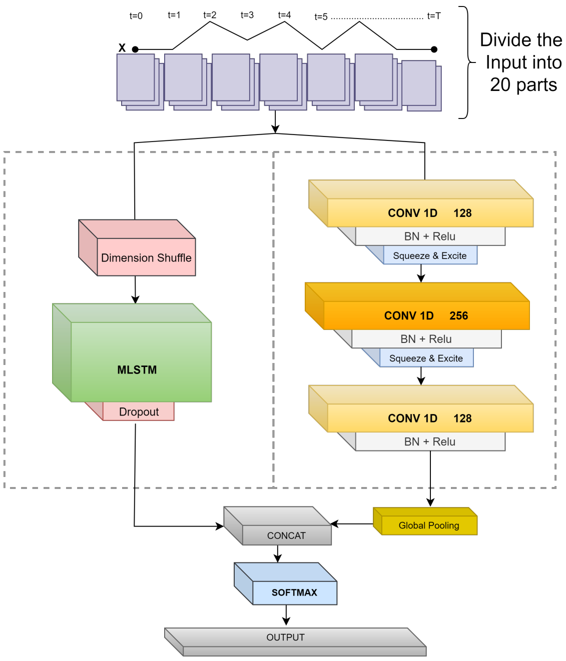

Figure 1: MLSTM-FCN 20-point search Architecture

For evaluation purpose, we used a uniform distribution of time-series size by only increasing the size by ten percent for each of 10 models that were trained. These models provide the classification accuracy for every ten percent increase in earliness.

3.1.1 Loss and Accuracy

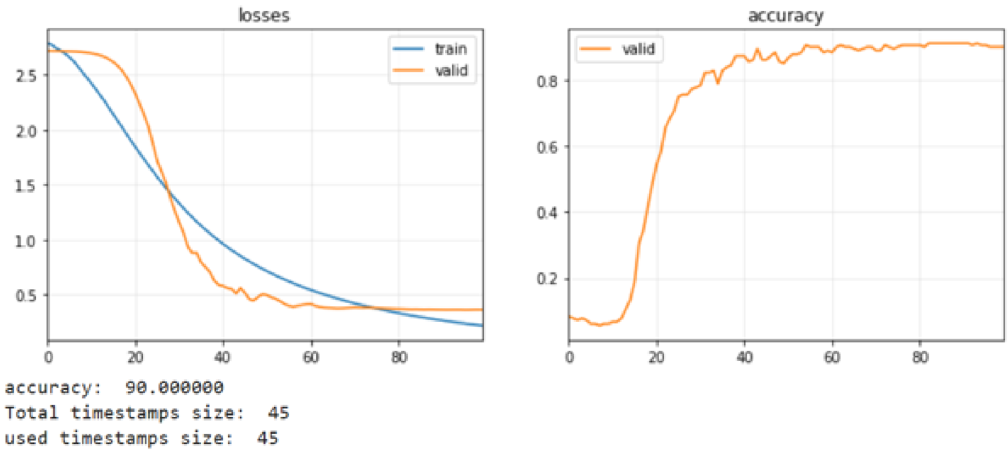

Figure 2: a)Epoch vs Loss b)Epoch vs Accuracy

Classification Accuracy of univariate and multivariate time series dataset at different timeseries size are shown in Table 1.

Table 1: Univariate and Multivariate Accuracy

MTS	Accuracy(20 point search)			UTS	Accuracy(20 point search)		
% T Used	Pendigits	libras	UwaveG	% T Used	wafer	Mote	ElectricD.
100	98.71	91.66	92.18	100	99.93	90.41	89.66
90	98.28	88.33	90.31	90	99.74	89.29	88.57
80	98.11	86.66	89.06	80	99.82	88.33	87.67
70	95.25	88.33	90.62	70	99.91	87.77	86.26
60	89.5	81.66	88.75	60	99.78	88.09	83.58
50	90.08	81.11	86.56	50	99.87	88.09	81.81
40	80.7	79.44	83.12	40	99.83	86.1	77.81
30	63.57	72.77	78.125	30	99.72	82.98	68.62
20	46.51	66.11	65.93	20	99.22	83.54	66.3
10	0	43.88	47.81	10	98.86	83.7	63.33

3.2 EDM (Early Decision Maker)

The MLSTM-FCN 20-Point Search model exhaustively searches for appropriate accuracy and earliness by training twenty different size time-series models, to overcome this drawback we proposed new model EDM which uses binary search for searching a particular classification accuracy. A hyper-parameter ‘A’ can be defined to search for accuracy with minimum timestamps used.

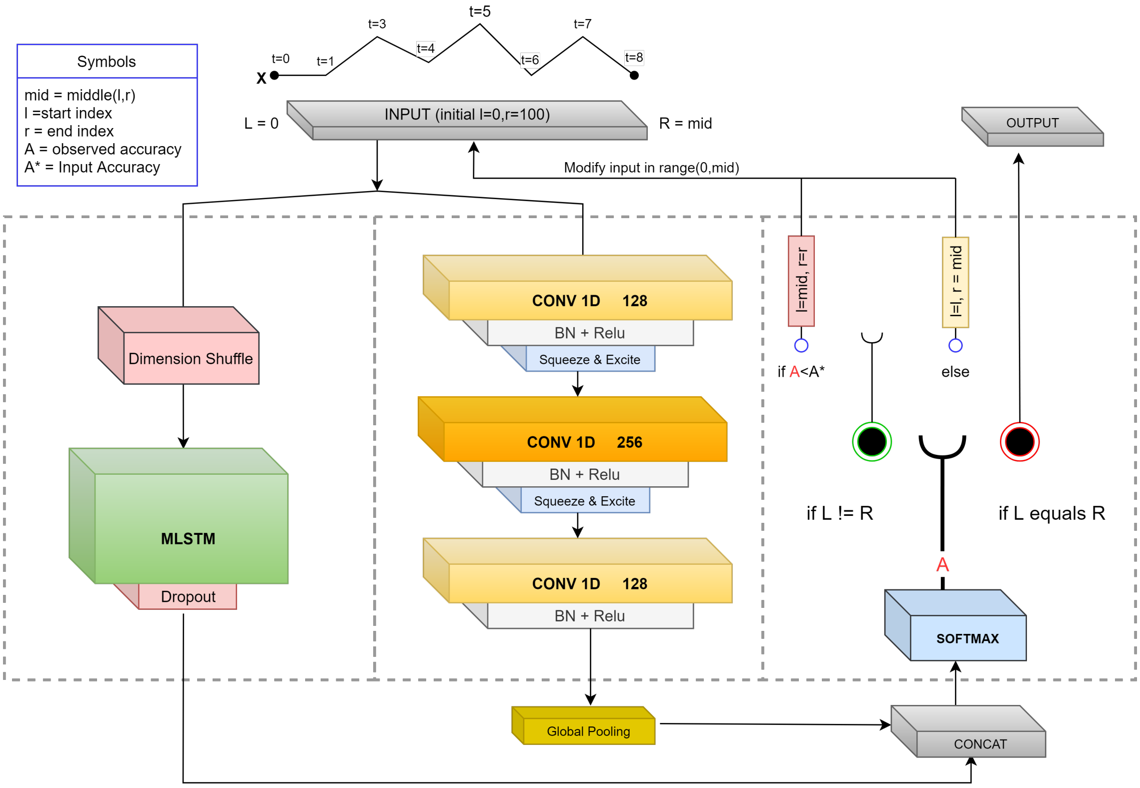

Figure 3: EDM Architecture

The percentage representation of timeseries can be treated as an index which maps to the accuracy of classification achieved by the model using that percentage of time series. The mapping enables us to use binary search on the index, to receive the lowest possible timeseries size which can produce the desired accuracy. After the Binary search block is executed, a minimum timestamp size trained model is obtained which produces accuracy(A) closest to the input accuracy(A*).

3.2.1 Loss and Accuracy

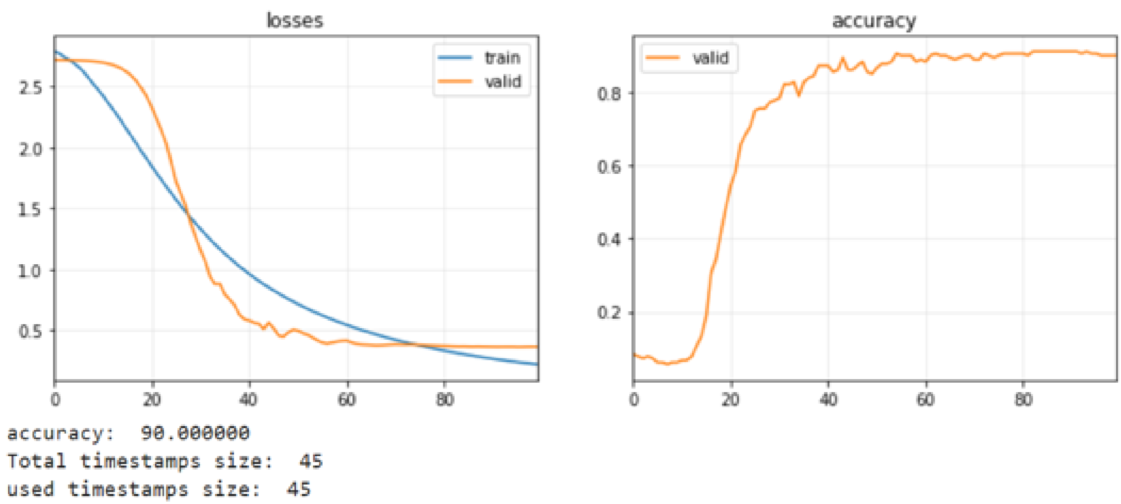

Figure 4: a)Epoch vs Loss b)Epoch vs Accuracy

Classification Accuracy of univariate and Multivariate time series dataset at different timeseries size are shown in Table 2 and 3.

Table 2: Univariate dataset Accuracy

UTS	Accuracy(EDM)					
% T Used	wafer	Mote	ElectricD.	Epilepsy	NATOPS	Articular Word Recognition
100	99.93	90.41	89.66	96.37	94.44	98.66
90	99.74	89.29	88.57	96.37	93.33	98.66
80	99.82	88.33	87.67	97.1	94.44	98.33
70	99.91	87.77	86.26	97.1	93.33	98.33
60	99.78	88.09	83.58	96.37	88.88	98.01
50	99.87	88.09	81.81	95.65	82.77	97.33
40	99.83	86.1	77.81	94.92	72.77	93.66
30	99.72	82.98	68.62	94.92	52.22	92.01
20	99.22	83.54	66.3	93.47	30	90.66
10	98.86	83.7	63.33	88.4	17.77	88.01

Table 3: Mutivariate dataset Accuracy

MTS	Accuracy(EDM)						
% T Used	Pendigits	libras	UwaveG	ACSF1	Adiac	ShapesAll	SKAppliances
100	98.71	91.66	92.18	93	86.95	90	78.66
90	98.28	88.33	90.31	91	84.39	91.33	79.47
80	98.11	86.66	89.06	91	84.91	90.83	79.2
70	95.25	88.33	90.62	90	83.37	88.16	81.87
60	89.5	81.66	88.75	90	81.58	86.66	79.2
50	90.08	81.11	86.56	88	80.81	85.66	75.47
40	80.7	79.44	83.12	88	80.05	80.5	80.53
30	63.57	72.77	78.125	89	78.77	78.16	74.4
20	46.51	66.11	65.93	87	70.33	73	77.07
10	0	43.88	47.81	82	60.61	59.83	76.53

3.3 G-LSTM (General LSTM)

The MLSTM-FCN classification model requies a fixed input size for classification, previous model had a fixed input size which was inconvenient for training as for every input size of the same dataset a new model has to be trained with that specific input size. A general model for all input Sizes could solve this problem and provide room for adaptive early classification approaches. To overcome this we used Used interpolation method to convert different TS sizes into one Size. This allowed us to train a single model for all timeseries sizes. The trained model gives us the accuracy of classification when halted at input timestamp T.

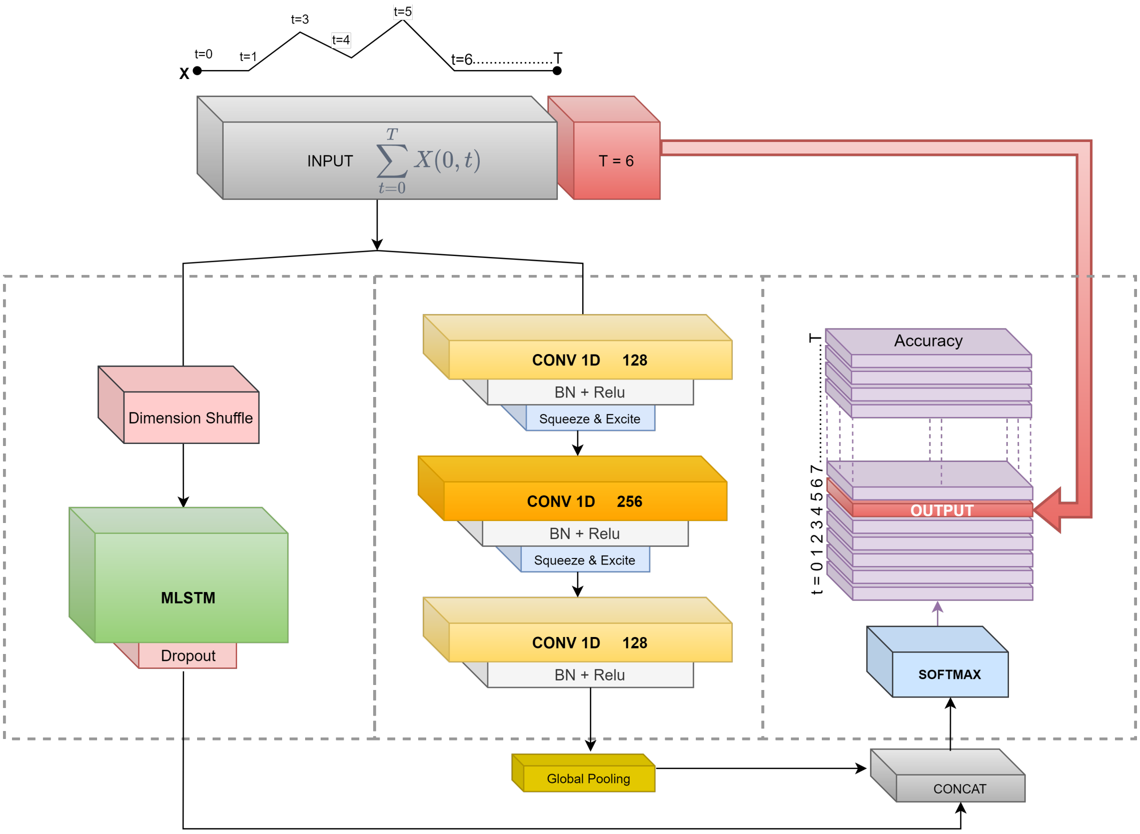

Figure 5: G-LSTM Architecture

3.3.1 Loss and Accuracy

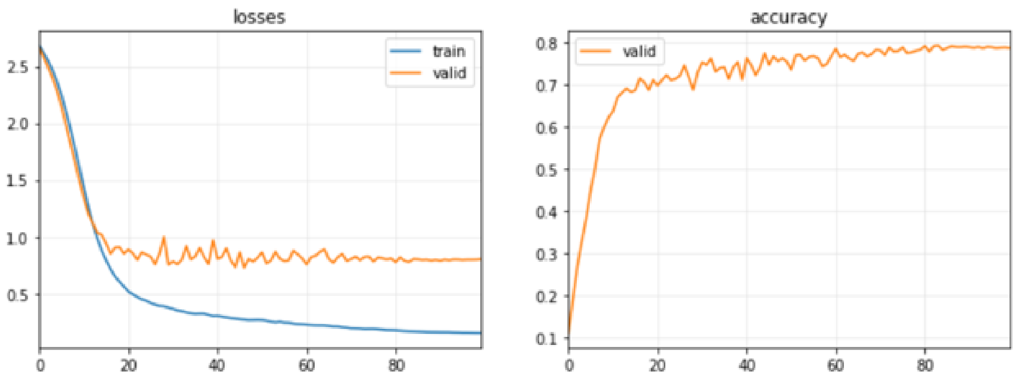

Figure 6: a)Epoch vs Loss b)Epoch vs Accuracy

Classification Accuracy of G-LSTM with univariate timeseries datasets (Pendigits,Libras) and multivariate time series datasets (Mote,wafer)at different timeseries size ("% T Used" represent percentage of timeseries used for classification) are shown in Table 4.

Table 4: Accuracy vs Earliness G-LSTM

MTS	Accuracy(G-LSTM)		UTS	Accuracy(G-LSTM)	
% T Used	Pendigits	libras	% T Used	Mote	wafer
100	98.68	90	100	91.69	99.7
90	98.42	90.55	90	93.13	99.7
80	97.45	90	80	91.69	99.69
70	96.14	91.11	70	89.29	99.77
60	90.65	89.44	60	90.25	99.8
50	90.65	83.33	50	88.73	99.83
40	81.96	81.11	40	88.33	99.87
30	64.92	76.11	30	87.06	99.88
20	46.76	69.44	20	86.1	99.85
10	46.76	45	10	81.74	99.62

3.4 LCR (LSTM-CNN-RL)

Previous models tried to incorporate earliness by fixed approaches, we propose a LCR (LSTM-CNN-RL) model that uses a LSTM block and a FCN block to extract features from the input data at each time stamps, the extracted features are then passed through a RL Agent which decides based on these input features whether or not to halt and predict a label. The Agent solves a Partially-Observable Markov Decision Process (POMDP) in which the features extracted by the LSTM-FCN block is recieved at each timestamp, depending on the agent's learned policy an action is sampled and based on the sampled action's quality a reward is observed. Agent's job is to maximize its reward based on accuracy and earliness of prediction.

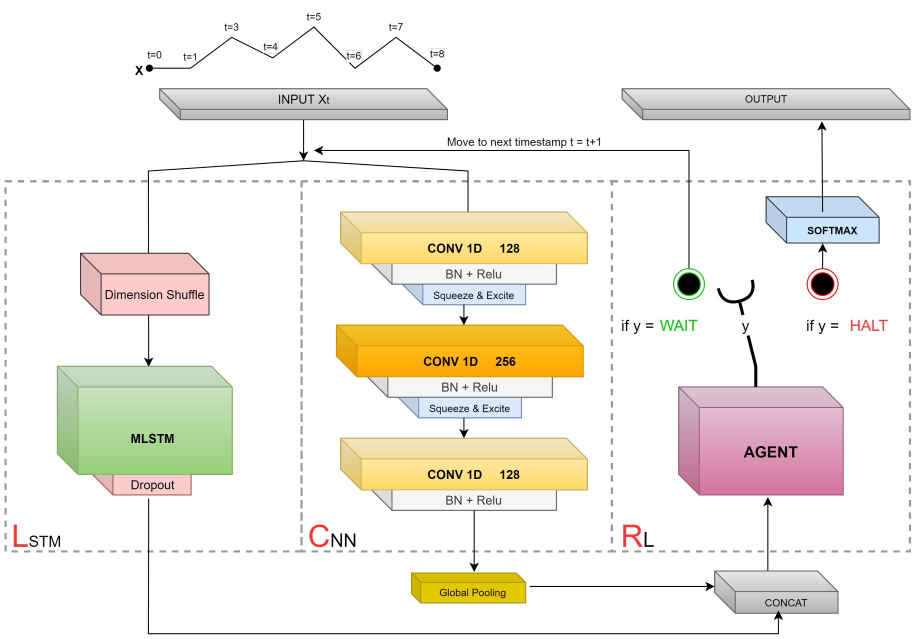

Figure 7: LCR Architecture

The accuracy and earliness of the classification work against each other, while focusing too much on earliness, accuracy gets severely compromised. To address this problem of accuracy-earliness trade off we introduced a hyperparameter Lambda λ which control the earliness of model by penalising for loss in accuracy at that halting point.

The LCR model uses a general model for all input Sizes for adaptive early classification. we used interpolation method to convert different TS sizes into one size. This allowed us to train a single model for all timeseries sizes. Other than interpolation we also used mean values for vacant places to convert timeseries into required size. In general the method of interpolation worked better for early classificaion for most of the datasets as show in Table 5.

We Compared our model with another deep learning model used for early classification like Earliest [3] our model performed better in terms of extracting features from the timeseries and gave better accuracy and earliness on different datasets. Comparison between LCR and Earliest are shown in Table 5.

Table 5: LCR and Earliest comparison

Dataset	**ACCURACY**			**EARLINESS**		
	EARLIEST	LCRv2	LCR	EARLIEST	LCRv2	LCR
ItalyPowerDemand	61%	63%	67%	24.81	35.6	19.63
SonyAIBORobotSurface1	72%	75%	85%	12.58	9.5	8.7
ECG200	65%	84%	88%	15.28	9.4	9.7
TwoLeadECG	64%	59%	80%	18.44	17.79	21
SonyAIBORobotSurface2	73%		78%	46.62		15.9
Seizures	91%		91%	17.82		17.778

3.4.1 Hyperparameter lambda λ

The RL agent maximizes its reward by selecting the optimal halting point for maximum accuracy, at some point in training the agent will learn that halting at the end fetches the maximum reward, to incorporate earliness into the RL agent an additional term can be added to the loss function which promotes adaptive early halting based on the hyperparameter lambda λ. Magnitude of lambda determines the weightage on earliness for RL agent, by increasing the lambda, the RL agent emphasises on earliness more.

we used the "NATOPS" dataset from UCR archives to analyse the effect of different values of lambda on accuracy and earliness, the results are show in figure 8.

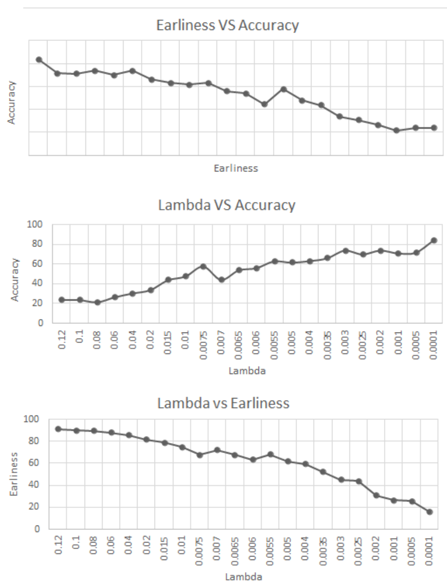

Figure 8: Effect of lambda

3.4.2 Loss, Accuracy and Earliness

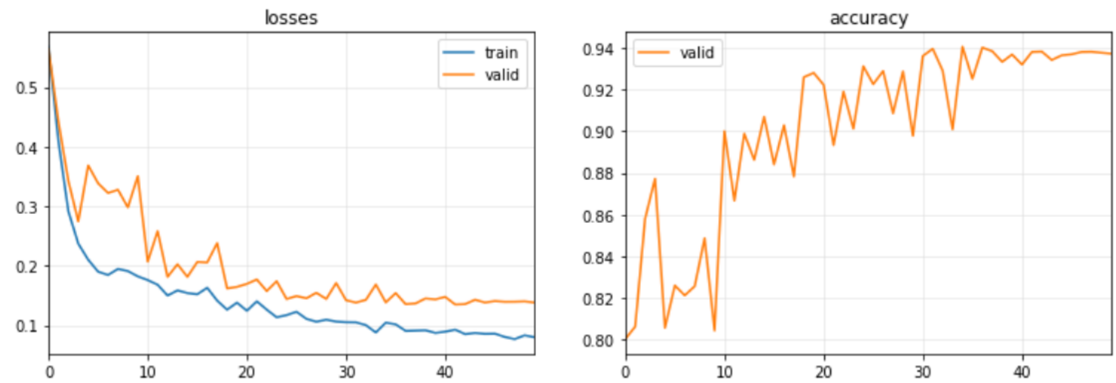

Figure 9: Classifier's Epoch vs loss/accuracy graph

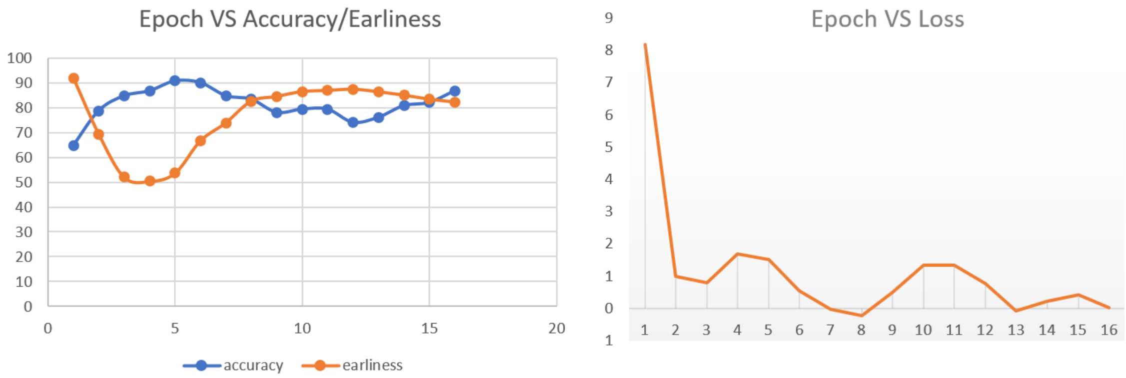

Figure 10: RL agent's Epoch vs loss/accuracy/earliness graph

Classification Accuracy of LCR along with parameter values are shown in Table 5.

Table 6: LCR Accuracy

Dataset	Type	Batch Size	epoch 1	epoch 2	lambda	accuracy	Earliness
SonyAIBORobotSurface1	SENSOR	5	50	200	0.0008	85%	8.7
SonyAIBORobotSurface2	SENSOR	6	6	200	0.0008	78%	15.9
GunPoint	MOTION	50	50	200	0.0008	87%	16.8
ECG200	ECG	25	50	200	0.0008	88%	9.7
MoteStrain	SENSOR	5	50	250	0.0008	65%	8.7
TwoLeadECG	ECG	5	50	250	0.00001*	80%	21
ItalyPowerDemand	SENSOR	16	12	250	0.02	67%	19.63

4 Experimental Evaluations

We evaluated LCR on 7 datasets from the UCR archive, these datasets have also been used in evaluation of proir work on early classification of timeseries. The datasets on UCR archive provide a train and test split, We compared our approach to the state-of-the-art methods, TEASER[9] ,ECTS (Xing et al. 2012)[10], RelClass (Parrish et al. 2013)[8], EDSC (Xing et al. 2011)[11], and ECDIRE (Mori et al. 2017b)[7]. To compare these methods with LCR, we used their published values of accuracy and earliness on UCR datasets.

4.1 Compared Methods

TEASER [9]
Teaser (early and accurate time series classification 2020) models eTSC as a two-tier classification problem: In the first tier, a classifier periodically assesses the incoming time series to compute class probabilities. However, these class probabilities are only used as output label if a second-tier classifier decides that the predicted label is reliable enough.
ECDIRE [7]
Early Classification of Time Series based on Discriminating Classes Over Time (ECDIRE) (Mori et al. 2017b) classifiers are trained at certain time stamps, i.e. at percentages of the full time series length. It learns a safe time stamp (the start time) as the fraction of the time series which states that a prediction is safe. Furthermore, a reliability threshold is learned using the difference between the two highest class probabilities. Only predictions passing this threshold after the safe time stamp are chosen.
RelClass [8]
Reliable Early Classification (Parrish et al. 2013) presents a method based on quadratic discriminant analysis (QDA). A reliability score is defined as the probability that the predicted class for the truncated and the whole time series will be the same. At each time stamp, RelClass then checks if the reliability is higher than a user-defined threshold.
ECTS [10]
In ECTS (Mori et al. 2017a) Early classification is approached as an optimization problem. The authors combine a set of probabilistic classifiers with a stopping rule that is optimized using a cost function on earliness and accuracy.
EDSC [11]
EDSC (Xing et al. 2011) learns Shapelets that appear early in the time series, and that discriminate between classes as early as possible.

Table 7: Accuracy Comparison

Dataset	ACCURACY					
	LCR	TEASER	ECDIRE	RelClass	ECTS	EDSC
MoteStrain	65%	89%	80%	58%	88%	78%
ItalyPowerDemand	67%	67%	93%	85%	94%	82%
SonyAIBORobotSurface2	78%	78%	74%	88%	85%	81%
TwoLeadECG	80%	77%	81%	72%	73%	88%
SonyAIBORobotSurface1	85%	85%	83%	79%	69%	80%
ECG200	88%	86%	91%	89%	89%	85%
GunPoint	87%	84%	87%	91%	87%	94%
Dataset	EARLINESS					
	LCR	TEASER	ECDIRE	RelClass	ECTS	EDSC
MoteStrain	8.7	21	12.1	90.94	79.06	38.08
ItalyPowerDemand	19.63	22	70.16	35.92	79.33	67.08
SonyAIBORobotSurface2	15.9	21	17.66	70.86	54.54	35.51
TwoLeadECG	21	21	69.38	83.63	64.43	46.85
SonyAIBORobotSurface1	8.7	28	62.26	57.7	68.49	47.03
ECG200	9.7	20	90.1	68.81	60.11	23.24
GunPoint	16.8	23	32.37	71.33	46.92	45.58

5 Observations and Result

The classification of time series problem had many effective solutions proposed by different authors, but We used MLSTM-FCN proposed by Fazle Karim [5] as our classifier, as we observed that implementing a reinforcement learning agent is most feasible on MLSTM-FCN. While we strictly used MLSTM-FCN for LCR model, other proposed models can change their classifier, we tested Inception Time[2] and ROCKET[1] on our models, which produced decent classification. MLSTM-FCN 20-point search and EDM had few major draw backs, these methods exhaustively train to find accuracy and earliness and are not adaptive, while G-LSTM model does not exhaustively train for accuracy and earliness, it still lacks in selecting adaptive halting point. LCR model addresses these drawbacks and delivers an efficient and effective solution which has a adaptive halting policy.

6 Conclusion

During the course of this Internship we developed a deep reinforcement learning model LCR which combines LSTM, CNN and RL for early classification of timeseries. Initial challenge was training different model for all timeseries sizes was computationally expensive, we used technique of interpolation to make a general model for all timeseries size of a dataset, which helped us to develop a advaptive halting stratagy with help of a reinforcement learning agent to avoid fixed apporaches for halting. Our model learns to adaptivily halt early without compromising on accuracy, We evaluated LCR model using 7 different datasets from UCR time-series dataset archive. Results show that our method significantly outperforms other methods in both accuracy and earliness

References

[1] Angus Dempster, François Petitjean, and Geoffrey I Webb. Rocket: exceptionally fast and accurate time series classification using random convolutional kernels. *Data Mining and Knowledge Discovery*, 34(5):1454–1495, 2020. 11

[2] Hassan Ismail Fawaz, Benjamin Lucas, Germain Forestier, Charlotte Pelletier, Daniel F Schmidt, Jonathan Weber, Geoffrey I Webb, Lhassane Idoumghar, Pierre-Alain Muller, and François Petitjean. Inceptiontime: Finding alexnet for time series classification. *Data Mining and Knowledge Discovery*, 34(6):1936–1962, 2020. 1, 11

[3] Thomas Hartvigsen, Cansu Sen, Xiangnan Kong, and Elke Rundensteiner. Adaptive-halting policy network for early classification. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining*, KDD '19, page 101–110, New York, NY, USA, 2019. Association for Computing Machinery. 9

[4] Magnus Lie Hetland. A survey of recent methods for efficient retrieval of similar time sequences. In *Data mining in time series databases*, pages 23–42. World Scientific, 2004. 1

[5] Fazle Karim, Somshubra Majumdar, Houshang Darabi, and Samuel Harford. Multivariate lstm-fcns for time series classification. *Neural Networks*, 116:237–245, 2019. 1, 3, 11

[6] Matthew Middlehurst, James Large, Michael Flynn, Jason Lines, Aaron Bostrom, and Anthony J. Bagnall. HIVE-COTE 2.0: a new meta ensemble for time series classification. *CoRR*, abs/2104.07551, 2021. 1

[7] Usue Mori, Alexander Mendiburu, Eamonn Keogh, and Jose A Lozano. Reliable early classification of time series based on discriminating the classes over time. *Data mining and knowledge discovery*, 31(1):233–263, 2017. 2, 10, 11

[8] Nathan Parrish, Hyrum S Anderson, Maya R Gupta, and Dun Yu Hsiao. Classifying with confidence from incomplete information. *The Journal of Machine Learning Research*, 14(1):3561–3589, 2013. 2, 10, 11

[9] Patrick Schäfer and Ulf Leser. Teaser: early and accurate time series classification. *Data Mining and Knowledge Discovery*, 34(5):1336–1362, 2020. 10, 11

[10] Zhengzheng Xing, Jian Pei, and S Yu Philip. Early classification on time series. *Knowledge and information systems*, 31(1):105–127, 2012. 2, 10, 11

[11] Zhengzheng Xing, Jian Pei, Philip S Yu, and Ke Wang. Extracting interpretable features for early classification on time series. In *Proceedings of the 2011 SIAM international conference on data mining*, pages 247–258. SIAM, 2011. 2, 10, 11