8-2021

Comparative Study of Machine Learning Models on Solar Flare Prediction Problem

Nikhil Sai Kurivella
Utah State University

Recommended Citation

Kurivella, Nikhil Sai, "Comparative Study of Machine Learning Models on Solar Flare Prediction Problem" (2021). *All Graduate Theses and Dissertations*. 8140.
https://digitalcommons.usu.edu/etd/8140

COMPARATIVE STUDY OF MACHINE LEARNING MODELS ON SOLAR FLARE PREDICTION PROBLEM

by

Nikhil Sai Kurivella

A thesis submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Computer Science

Approved:

Soukaina Filali Boubrahimi, Ph.D. Major Professor	Nicholas Flann, Ph.D. Committee Member
Curtis Dyreson, Ph.D. Committee Member	D. Richard Cutler, Ph.D. Interim Vice Provost of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2021

ABSTRACT

Comparative Study of Machine Learning Models on Solar Flare Prediction Problem

by

Nikhil Sai Kurivella, Master of Science

Utah State University, 2021

Major Professor: Soukaina Filali Boubrahimi, Ph.D.
Department: Computer Science

Solar flare events are explosions of energy and radiation from the Sun's surface. These events occur due to the tangling and twisting of magnetic fields associated with sunspots. When Coronal Mass ejections accompany solar flares, solar storms could travel towards earth at very high speeds, disrupting all earthly technologies and posing radiation hazards to astronauts. For this reason, the prediction of solar flares has become a crucial aspect of forecasting space weather. Our thesis utilized the time-series data consisting of active solar region magnetic field parameters acquired from SDO that span more than eight years. The classification models take AR data from an observation period of 12 hours as input to predict the occurrence of flare in next 24 hours. We performed preprocessing and feature selection to find optimal feature space consisting of 28 active region parameters that made our multivariate time series dataset (MVTS). For the first time, we modeled the flare prediction task as a 4-class problem and explored a comprehensive set of machine learning models to identify the most suitable model. This research achieved a state-of-the-art true skill statistic (TSS) of 0.92 with a 99.9% recall of X-/M- class flares on our time series forest model. This was accomplished with the augmented dataset in which the minority class is over-sampled using synthetic samples generated by SMOTE and the majority classes are randomly under-sampled. This work has established a robust dataset and baseline models for future studies in this task, including experiments on remedies to tackle the class imbalance problem such as weighted cost functions and data augmentation. Also the time series classifiers implemented will enable shapelets mining that can provide interpreting ability to domain experts.

(78 pages)

PUBLIC ABSTRACT

Comparative Study of Machine Learning Models on Solar Flare Prediction Problem

Nikhil Sai Kurivella

Solar flare events are explosions of energy and radiation from the Sun's surface. These events occur due to the tangling and twisting of magnetic fields associated with sunspots. When Coronal Mass ejections accompany solar flares, solar storms could travel towards earth at very high speeds, disrupting all earthly technologies and posing radiation hazards to astronauts. For this reason, the prediction of solar flares has become a crucial aspect of forecasting space weather. Our thesis utilized the time-series data consisting of active solar region magnetic field parameters acquired from SDO that span more than eight years. The classification models take AR data from an observation period of 12 hours as input to predict the occurrence of flare in next 24 hours. We performed preprocessing and feature selection to find optimal feature space consisting of 28 active region parameters that made our multivariate time series dataset (MVTS). For the first time, we modeled the flare prediction task as a 4-class problem and explored a comprehensive set of machine learning models to identify the most suitable model. This research achieved a state-of-the-art true skill statistic (TSS) of 0.92 with a 99.9% recall of X-/M- class flares on our time series forest model. This was accomplished with the augmented dataset in which the minority class is over-sampled using synthetic samples generated by SMOTE and the majority classes are randomly under-sampled. This work has established a robust dataset and baseline models for future studies in this task, including experiments on remedies to tackle the class imbalance problem such as weighted cost functions and data augmentation. Also the time series classifiers implemented will enable shapelets mining that can provide interpreting ability to domain experts.

To my parents, for their endless love, support and encouragement.

ACKNOWLEDGMENTS

First and foremost, I am extremely grateful to my major professor, Dr. Soukaina Filali Boubrahimi for her valuable advice, encouragement, and patience during my Masters study. I would also like to thank Dr. Nicholas Flann and Dr. Curtis Dyreson for taking the responsibility for being in my committee and providing support. Finally, I would like to express my gratitude to my parents for their invaluable support and encouragement that made it possible for me to complete my study.

Nikhil Sai Kurivella

CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

ACRONYMS

MVTS	Multivariate Time Series
AR	Active Region
SDO	Solar Dynamics Laboratory
NASA	National Aeronautics and Space Administration
CME	Coronal Mass Ejections
GOES	Geostationary Operational Environmental Satellite
NOAA	National Oceanic and Atmospheric Administration
SHARP	Space weather HMI active region patch
HMI	Helioseismic Magnetic Imager
PCA	Principal component Analysis
TSF	Time Series Forest
RF	Random Forest
SVM	Support Vector Machine
LSTM	Long Short Term Memory
RNN	Recurrent Neural Network
SAXVSM	Symbolic Aggregate Approximation and Vector Space Model
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
SMOTE	Synthetic minority oversampling technique

CHAPTER 1

INTRODUCTION

The Sun has a massive impact on our planet and space behavior. Many changes, disturbances, and phenomena keep occurring in and around the Sun's atmosphere, generally referred to as Solar activity. It has several variables that fluctuate even on a fraction of a millisecond. There are four primary forms of solar activity: solar flares, coronal mass ejections, high-speed solar winds, and solar energetic particles. While each of these forms can negatively impact the earth, X-, M- solar flares and coronal mass ejections can disrupt all earthly technologies, space architectures, destroy power grids, and pose radiation hazards [2]. While posing such a significant threat, these extreme events are very rare, making the prediction task very challenging.

With the goal of understanding the influence of sun on earth's behavior they launched a Solar dynamics observatory mission in 2010. It continuously observes the sun's magnetic field and structure and tries to unfold how such huge amounts of energy is accumulated and released in the form of solar winds into heliosphere and geospace. Although several research communities are utilizing this information to explain the occurrence of flares, no theory was able to certainly explain the mechanism behind them; which is hampering the efforts to forecast. Meanwhile, with the growth in artificial data intelligence, space weather strategies are drifting towards data driven approaches to find an efficient way.

Our thesis attempts to help space weather forecasters predict solar flares occurring in next 24 hours using machine learning techniques. We trained and evaluated nine machine learning models to create a baseline for flare prediction tasks using a multivariate time series (MVTS) dataset. Following are the classification models implemented in this thesis: fully connected neural networks, convolutional neural networks [3], long short term memory networks, symbolic aggregate approXimation vector space model, time series forests, random forests, shapelet transforms, support vector machines and logistic regression.

1.1 What are Solar flares?

Solar flares are massive eruptions of electromagnetic radiation from the sun's surface. They originate from the release of high energy accumulated in the magnetic field of solar active regions. When such areas suddenly explode they travel straight out from the flare sight with incredible velocity in the form of solar winds. If the flare sight is facing the earth, then there is a possibility that it can reach the earth's atmosphere or human-interested space. Nevertheless, when these solar flares are accompanied by large clouds of plasma and magnetic field explosions called coronal mass ejections (CME) [4], they get splattered. They can travel in any direction causing high-speed solar storms. An image of an evident solar flare captured on August 24, 2014, by NASA's solar dynamics observatory is displayed in figure 1.1

Solar flares are classified into five different levels based on their peak x-ray flux. The figure 1.2 is the snapshot of 2 days from the GOES satellite observation where there is an instance of X and M class flares. They are A, B, C, M, X classes, starting from A, each class increases ten-fold in intensity. Most intense X and M class flares cause significant risk to humans. In contrast, other flares can hardly impact human interests but can play a major role in studying behavior of sun's atmosphere.

1.2 Why do we need to predict Solar Flares?

Intense enough Solar flares and CMEs, when they reach the earth's atmosphere or human interested space regions, can derange telecommunications, machinery, computers, electrical appliances, etc., and can even blackout the earth. The airline passengers and crew, astronauts in the space, space infrastructure, and satellites are at huge risk to such flares [5].

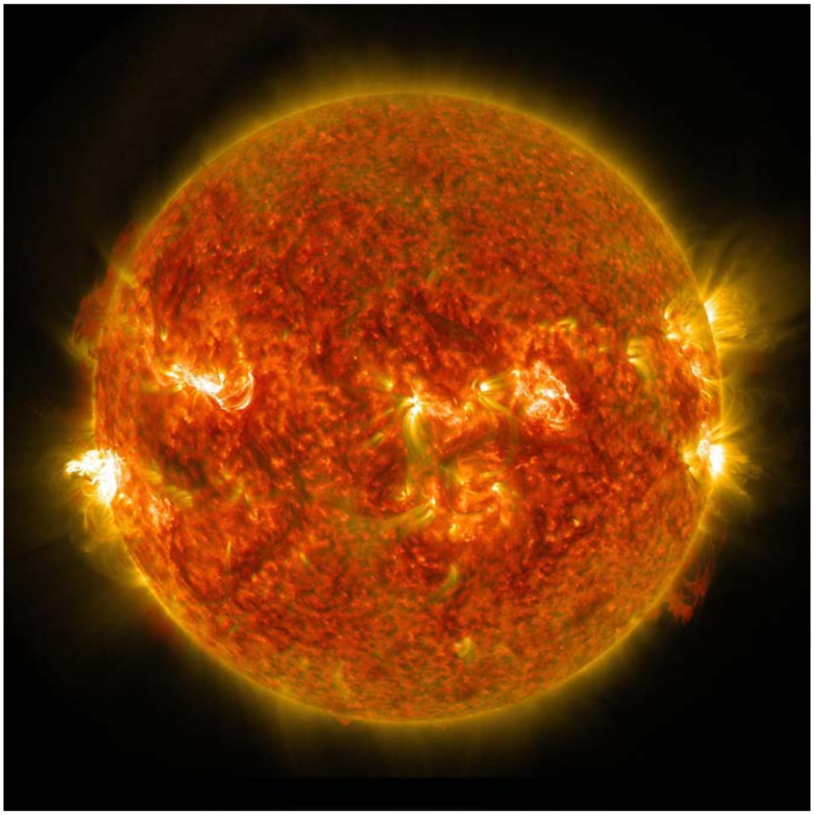


Fig. 1.1: M class solar flare captured by NASA's SDO. Image Credit: NASA/SDO

The sun goes through a solar cycle of 11 years, during which its activity decreases and increases. Usually, when the activity ramps up, frequent massive solar flares and CMEs occur, and chances that our planet can come in its way increases. However, flares hitting the earth's atmosphere are sporadic, but there are several instances in the last decade where the earth has faced severe consequences. There were also cases where the earth could have been wreak havoc if not for mere luck.

1.3 Few Instances when Solar flares hit the earth

- July 23, 2012, when earth made a narrow escape

  An unusually large solar flare(CME) has hit NASA's solar-terrestrial relations Observatory satellite (STEREO) spacecraft that orbits the sun. As it was placed in orbit to observe and withstand such storms and flares, it survived the encounter. However,

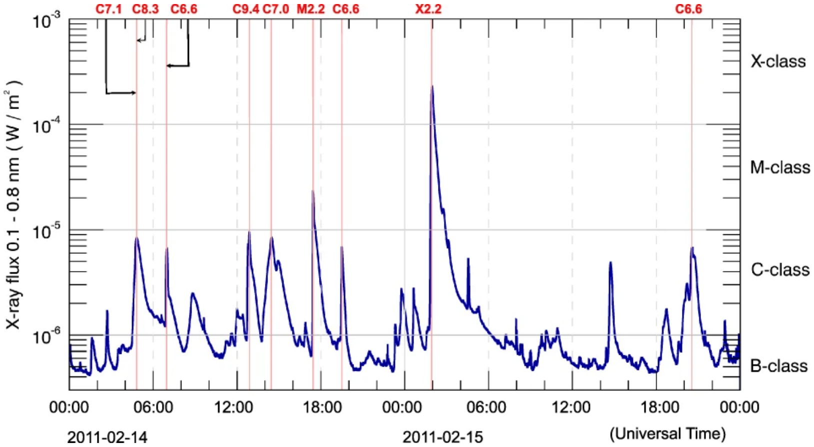


Fig. 1.2: Image Credit: [1] Levels of solar flares as measured by GOES satellite near Earth

when scientists calculated its occurrence, they realized that the earth had missed it by a margin of 9 days. Later in 2014, they have estimated that if a similar flare had occurred at that time, it would have caused cataclysmic damage worth $2 trillion, cost years to repair, and left people with the void of electricity.

- March 10, 1989, when North America faced a blackout.

  A powerful CME has caused electric currents on the grounds of North America. It has created chaos by affecting around 200 power grids in Quebec, Canada, which resulted in 12 hr blackout. It interfered with sensors on the space shuttle Discovery and disrupted communication with satellites in space. In 1989, we encountered two other storms, one of which crashed computer hard drives halting Toronto's stock market for few hours, and the other posed a life risk to astronauts causing burning in their eyes.

- May 23, 1967, when United States thought it's Russia.

  A large Solar flare has jammed the U.S. and United Kingdom's radar and radio communications. The U.S. military has begun preparing for war, suspecting the soviets for the failure in communication systems; Just in time, space weather forecasters have intervened.

Predicting the solar flares upfront may not help us to avoid the storms but could help us in mitigating the damage by taking necessary precautions. As humans are now are exploring space more than ever and becoming more and more technology and electricity-dependent, the risk the flares could pose to humans is shooting up [2]. Therefore, continuous monitoring of solar events is crucial for accurately forecasting space weather. NOAA has a constellation of geostationary operational environmental Satellites which send continuous stream of data to support the weather forecasting, solar events tracking and other research. However, the mechanism behind their occurrence is still not completely known.

As the understanding of theoretical models of solar flare events such as the relationship between the photospheric and coronal magnetic field is limited, the heliophysics community is showing an inclination towards data-driven approaches to replace the expensive setup [5]. **Note:** Information and facts about solar flares presented in this thesis have been taken from the web sources of NASA, NOAA, SDO.

1.4 What is the existing knowledge?

This section summarizes our study of the already established knowledge of flare prediction studies. Significantly, the prediction pipelines that were built using machine learning models. Most of the approaches were data-driven except for the rule-based expert system called THEO by Mcintosh [6] in the earlier days of flare prediction in 1990. Also, most of the earlier studies did not consider the data's time-series nature.

Initial data-driven approaches were based on linear statistical modeling. In the work of Jing [7], they tried to find the correlations between the magnetic parameters and a flare production based on line of sight vector magnetogram data. Later on, several machine learning approaches were implemented using the vector magnetogram data. Below is the list of Machine learning techniques that have been experimented on in this task.

- Time series Forest(TSF) (2020) [8] They took a contrasting approach; they tried to predict if the forecasts flare-free or not while trying to minimize the number of false negatives and false alarms. They trained the interval based model called TSF, which uses a random forest classifier to make the prediction. They achieved able to detect non-flaring activity with a recall of 88.5%.

- Solar flare prediction using LSTM network (2020) [9] They reduced the flare prediction problem to binary classification and used 20 SHARP parameters to train an LSTM network. A TSS of 0.79 has been achieved in identifying flares $\geq$ M. They concluded that SHARP parameters have limited information.

- Gramian Angular Fields Images using Neural Networks (2018) [10]

  This research converted the time series X-ray data acquired from the GOES satellite into Gramian angular field images to train a convolutional neural network after compressing the data using a piece-wise approximation. However, their results were not impressive.

- k-NN based on Time series data (2017) [5]

  This study performed their experiments using a k-NN classifier with the vector magnetogram time-series data. They compared each of the 16 univariate feature's performance before concluding that "TOTUSJH" is the best AR parameter with a flaring region recall of 90% .

- Multivariate Time Series Decision Trees (2017) [11] This research has approached this problem using multivariate time series analysis perspective to cluster potential flaring active regions by applying distance clustering [12–15] to each feature and then organizing them back together into a multivariate time series decision tree. They achieved an accuracy of 85% in predicting the flaring activity.

- k-NN and ERT (Extremely randomized tree) (2017) [16]

  This research combined the line of sight data, vector magnetogram data, GOES X-ray data and UV continuum of the lower chromosphere from atmospheric imaging assembly to calculate 65 features. They trained three machine learning algorithms: SVM, k-NN, and ERT, to achieve the true skill score of 0.9. (Skill scores will be explained in the later section)

- Support Vector Machines (2015) [17] Their attempt to forecast the X- and M- flares relied on the 25 AR vector magnetogram features. They calculated the linear correlation between each feature and target variable to eliminate 12 features from their forecasting algorithm. They achieved a TSS skill score of 0.61 with the SVM algorithm.

Apart from the above-mentioned notable research on Flare prediction, more research is performed on this task in a data-driven approach. Right from 2001 to 2015 several machine learning algorithms were experimented to improve flare prediction; such as pattern recognition using relevance vector machine by Al-Ghraibah [18], cascade correlation Neural Networks by Ahmed [19], logistic regression by Y.Yuan [20], decision trees by D.Yu [21] and artificial neural networks by R.A.F. Borda [22].

Based on our literature study, we observed that the time-series nature of vector magnetogram data is not yet well studied except for the works of Angryk [11], [8], where they used time-series forest and multivariate decision trees. Secondly, the flare prediction problem is modeled chiefly as a binary classification problem. They combined X and M class flares as positive classes and C, B, Q as the negative class. Though binary modeling fulfills the main objective of identifying the most harmful flares, they ignore the importance of C and B class flares, completely treating them as non-flaring events. According to a study [23], C class flares are more valuable than X and M class flares in studying the evolution of sunspots. Furthermore, B class flares are more synchronous with sunspot group numbers than X and M when the number of sunspot groups is around 100. This study concludes C and B as low magnetic activity flares that could be crucial in studying the sunspot groups, which are the main cause of solar flares. Though there are a different variety of machine learning models experimented there is no common dataset that has been investigated on all of them to find the classifier that is the most suitable for this prediction task.

Considering the above-discussed points, our thesis aims to make an extensive comparison of different types of Neural Network architectures and time-series classifiers and the baseline comparison with traditional classifiers such as Random Forests, SVM, logistic regression. We modeled the problem as a 4 class problem to classify the flares between X, M, C/B, and Q classes. We hope that this contribution will be a valuable asset to the data-driven flare prediction community and fills a few gaps in this area's existing knowledge.

1.5 Thesis Organization

- Chapter 2: Introduction to the dataset, explains where the data is coming from and what are the features present, how much data we have and the challenges it brings and the feature selection pipeline used to arrive at the final dataset.
- Chapter 3: In this chapter, we discuss machine learning methodologies used in different experiments and the evaluation techniques used in our thesis including the domain specific metrics.
- Chapter 4: Here, we describe and discuss the results achieved in the experiments and compare the performance of classifiers.
- Chapter 5: In the final chapter, we discuss the outcomes, conclusions, challenges and future scope of this thesis.

CHAPTER 2
DATA ANALYSIS AND FEATURE SELECTION

2.1 Introduction to the Dataset

A Time series is a progression of a variable feature captured in continuous time intervals. The spacing between each time instance (τ) captured is known as cadence. Our dataset is time-series data with a cadence of 12 minutes through 60 time intervals (60τ, 12 hours long). As our data has multiple variable features (33), it is called as Multivariate Time series data. The below matrix represents the dimension of our sample.

Each sample of our data consists of 33 active region magnetic field parameters acquired from HMI instrument on solar dynamics observatory. Each feature is a time-series of mathematically represented magnetic field parameter. The table 2.2 below is the list of features and their description. More information about each feature can found at: "http://jsoc.stanford.edu/doc/data/hmi/sharp/sharp.htm".

$$MVTS_{(33,60)} = \begin{bmatrix} f_{1,1} & f_{1,2} & f_{1,3} & f_{1,4} & \cdots & f_{1,60} \\ f_{2,1} & f_{2,2} & f_{2,3} & f_{2,4} & \cdots & f_{2,60} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ f_{33,1} & f_{33,2} & f_{33,3} & f_{33,4} & \cdots & f_{33,60} \end{bmatrix}$$

2.2 Data Acquisition

Solar flares and CMEs occur in the Sun areas where there is a strong magnetic field. Such areas are called flaring Active regions(AR). Though the underlying mechanism of how solar flares and CMEs occur is still not well understood, there are several studies that show solar flares arise because of sudden release of energy developed by photospheric magnetic reconnection in the coronal field [24]. This is well known as CSHKP model; there are several other variations of these models explaining the initiation mechanisms of the flares and

CMEs. Since the photospheric magnetic field has a huge role in driving the coronal field around the Sun, it is highly possible that evolution patterns of the photospheric magnetic field may serve as indicators for predicting solar flares and CMEs.

The solar dynamic observatory satellite that has been launched by NASA a few years ago has been observing the Sun from 2010. It has three instruments set up on board for different purposes: Extreme Ultraviolet Variability Experiment (EVE), Helioseismic and Magnetic Imager (HMI) and Atmospheric Imaging Assembly (AIA). HMI takes the high-resolution measurement of photospheric vector magnetic field parameters and extracts space weather HMI AR patches (SHARPS). [25] AIA is responsible for imaging the outer surface(corona) of the sun, it captures full sun images every 10 seconds. EVE keeps track of extreme ultraviolet rays and helps us in calculating the amount of UV reaching the earth.

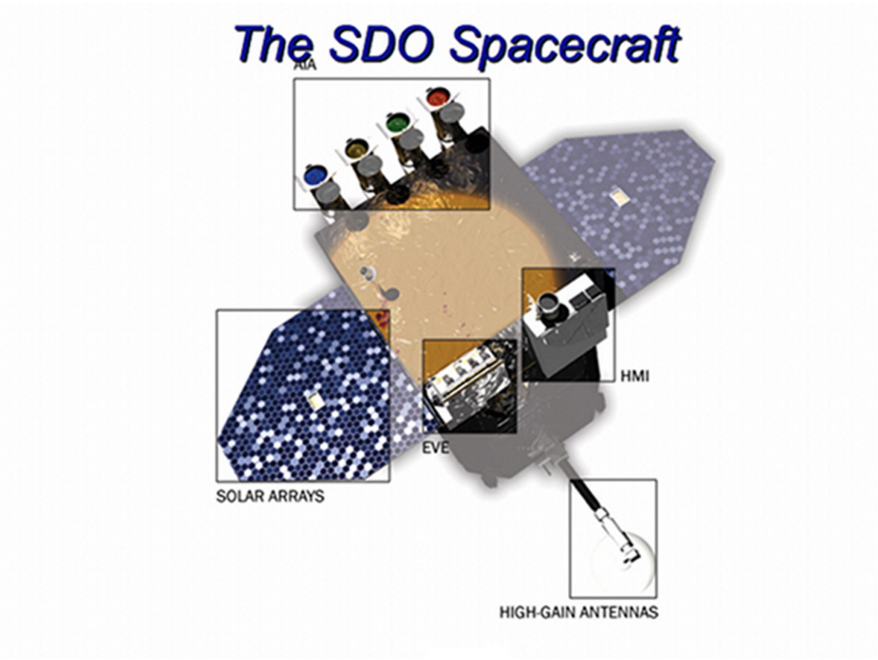


Fig. 2.1: This image shows the three main SDO instruments on-board: AIA, HMI, EVE. HMI is responsible for capturing solar magnetic field information and the primary source of AR parameters that we feed to the classifiers to train and test. Image credits: NASA

In our thesis, the data we are using is coming from the HMI called SHARPS. They have several pre-calculated time series quantities(AR parameters) captured by HMI every 12 minutes. These quantities called as active region (AR) parameters, and are our features in the each sample 2.2. Each time series is a look back of flare or no flare occurrence. Similar to the work of Bobra and Couvidat [17] we are using these past observations of AR parameters as preflare signatures to train the AI models to predict future flaring activity. These long SHARP time series are broken into chunks of 60 instances of span 12 hours, making our multivariate time series dataset. The label of each MVTS is the largest flare that occurred in next 12 hours (prediction period).

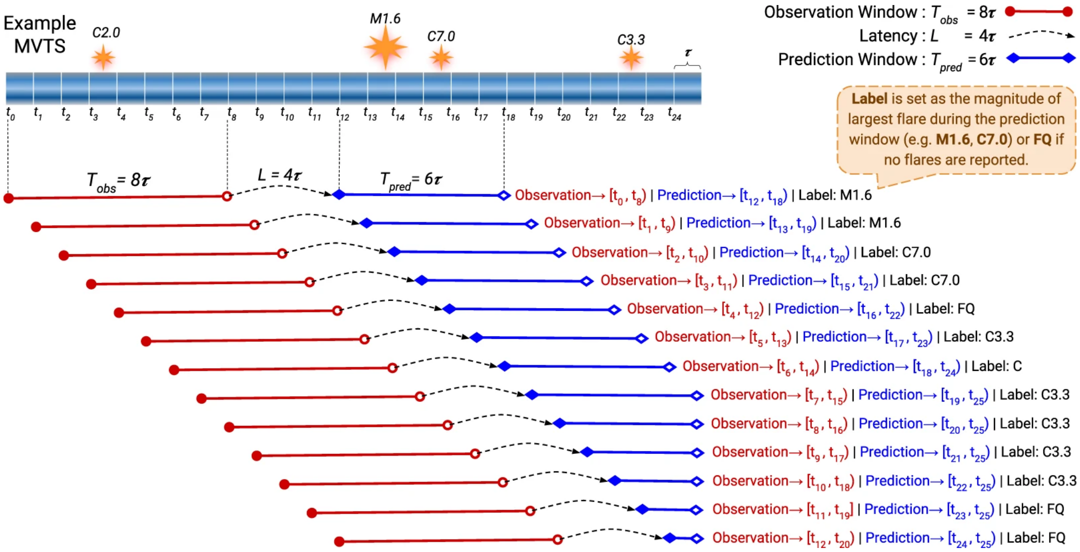

Example MVTS
C2.0
M1.6
C7.0
C3.3
τ
Observation Window : T_{obs} = 8τ
Latency : L = 4τ
Prediction Window : T_{pred} = 6τ
t_0 t_1 t_2 t_3 t_4 t_5 t_6 t_7 t_8 t_9 t_{10} t_{11} t_{12} t_{13} t_{14} t_{15} t_{16} t_{17} t_{18} t_{19} t_{20} t_{21} t_{22} t_{23} t_{24}
Label is set as the magnitude of largest flare during the prediction window (e.g. M1.6, C7.0) or FQ if no flares are reported.
T_{obs}= 8τ
L = 4τ
T_{pred}= 6τ
Observation→ $[t_0, t_8)$ | Prediction→ $[t_{12}, t_{18})$ | Label: M1.6
Observation→ $[t_1, t_9)$ | Prediction→ $[t_{13}, t_{19})$ | Label: M1.6
Observation→ $[t_2, t_{10})$ | Prediction→ $[t_{14}, t_{20})$ | Label: C7.0
Observation→ $[t_3, t_{11})$ | Prediction→ $[t_{15}, t_{21})$ | Label: C7.0
Observation→ $[t_4, t_{12})$ | Prediction→ $[t_{16}, t_{22})$ | Label: FQ
Observation→ $[t_5, t_{13})$ | Prediction→ $[t_{17}, t_{23})$ | Label: C3.3
Observation→ $[t_6, t_{14})$ | Prediction→ $[t_{18}, t_{24})$ | Label: C
Observation→ $[t_7, t_{15})$ | Prediction→ $[t_{19}, t_{25})$ | Label: C3.3
Observation→ $[t_8, t_{16})$ | Prediction→ $[t_{20}, t_{25})$ | Label: C3.3
Observation→ $[t_9, t_{17})$ | Prediction→ $[t_{21}, t_{25})$ | Label: C3.3
Observation→ $[t_{10}, t_{18})$ | Prediction→ $[t_{22}, t_{25})$ | Label: C3.3
Observation→ $[t_{11}, t_{19}]$ | Prediction→ $[t_{23}, t_{25})$ | Label: FQ
Observation→ $[t_{12}, t_{20})$ | Prediction→ $[t_{24}, t_{25})$ | Label: FQ

Fig. 2.2: The picture shows the observation window, which we feed to our classification models to predict the flare occurrence in the prediction period. In our case, the span of the observation window and the prediction window is 12 hours and 24 hours, respectively. Latency is minimized to the smallest possible time interval (the last point of the observation period is closest to the first point of the prediction period). Image Credits: [1]

The MVTS dataset was made available by a research team from Georgia State University. This dataset was made with an intention to facilitate machine learning models. The labels for each time series are also marked with utmost care after several phases of cleaning, pre-processing and cross checking with secondary sources. The block diagram briefly mentions the cleaning phases and secondary data sources 2.3. Much detailed information about AR parameters and the dataset can be found at [1].

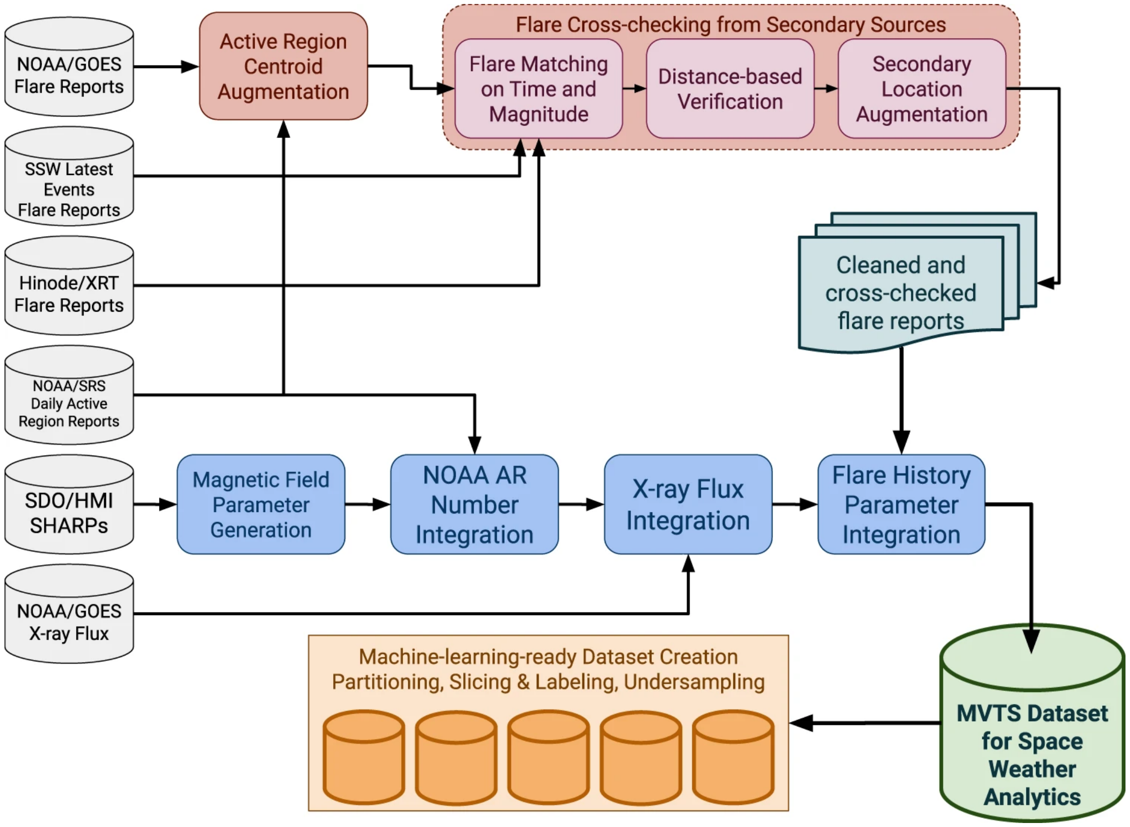

NOAA/GOES Flare Reports
SSW Latest Events Flare Reports
Hinode/XRT Flare Reports
NOAA/SRS Daily Active Region Reports
SDO/HMI SHARPs
NOAA/GOES X-ray Flux
Active Region Centroid Augmentation
Flare Cross-checking from Secondary Sources
Flare Matching on Time and Magnitude
Distance-based Verification
Secondary Location Augmentation
Cleaned and cross-checked flare reports
Magnetic Field Parameter Generation
NOAA AR Number Integration
X-ray Flux Integration
Flare History Parameter Integration
MVTS Dataset for Space Weather Analytics
Machine-learning-ready Dataset Creation
Partitioning, Slicing & Labeling, Undersampling

Fig. 2.3: The flow of information from several sources and cross checking before arriving at the final MVTS dataset ready for machine learning. Image Credits: [1]

2.3 Datasets

On the Georgia state university public server, the latest flare dataset is available. They have three partitions of data that are labeled. There are a total of 214023 MVTS, out of which only 1.98% of them are from X- and M- preflare signatures and 82.64% are flaring quiet regions (Q class), and the remaining are C and B class preflare signatures 2.1. The dataset excluded A class flare data as they do not pose any risk and are have almost negligible peak flux.

Datasource: https://dmlab.cs.gsu.edu/solar/data/data-comp-2020/

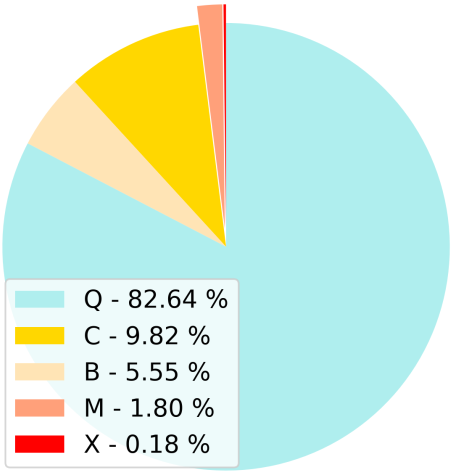


Fig. 2.4: Flare class distribution showing the huge class imbalance

Classes/Partition	X	M	C	B	Q
P1	172	1157	6531	6010	63400
P2	72	1393	9130	5159	78013
P3	141	1313	5364	709	35459
Class %	**0.179**	**1.80**	**9.82**	**5.54**	**82.64**

Table 2.1: Class Distribution across different data partitions

2.3.1 Class imbalance problem

From the pie chart in figure 2.4 it's explicit that we have fewer samples of the critical X- and M- class flares. On the other hand, we have a ton of samples for non-flaring instances, leaving us with class imbalance problem as high as 1 is to 800 between the minority X class and majority Q class. Class imbalance problems can skew the models drastically, making them biased to the classes with majority samples. On top of that, our majority class is the negative flare class (Q) which implies that the model could end up with too many false negatives (Increased chances of missing the flare detection) [26] .

For the scope of this thesis, we decided to generate two data sets to experiment with different machine learning models; one based on random under sampling and the other with a combination of oversampling and under-sampling to tackle the class imbalance problem. Note that we merged classes C and B as one class based on the discussion in the literature study. Hence our two final data sets will be having X, M, C/B, and Q classes.

2.3.2 Random under-sampling

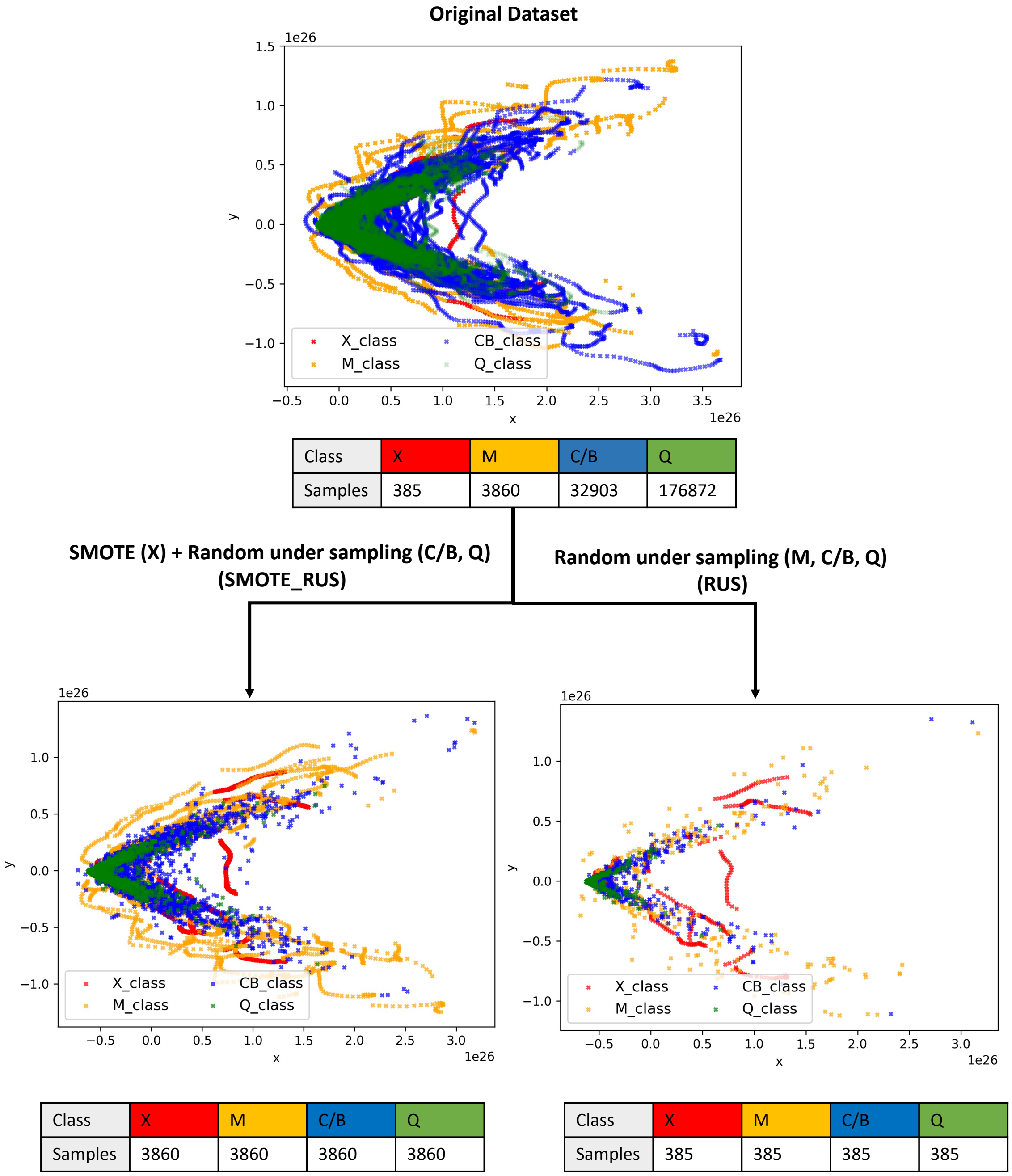
We took the classic random under-sampling approach on the classes with more samples to tackle this problem. In this method, we down-sample the majority classes to the minority classes by randomly selecting the required number of samples to be included. While this method looks very simple, we can't overlook its success in addressing the class imbalance in several applications [27]. Though random under-sampling might lead to the loss of unique data, in most cases, it is proven to be advantageous and generalized better than oversampling. In the study of Satwik Mishra [28], they compared SMOTE approach with under-sampling on randomly generated data sets and found out that random under-sampling is performing better with random forests and XGboost. Our under-sampled dataset has 1540 samples, with each of the majority classes down-sampled to 385 samples to be on par with the minority X- class. This dataset will be referred as RUS dataset in the rest of the thesis.

2.3.3 SMOTE along with random under-sampling

However, we can not say that one specific approach would work with this application. Hence, we did not wholly overlook the oversampling approach, as it is advantageous to accommodate more data for data-hungry models like neural networks. We arrived at a hybrid method that includes oversampling the minority class by SMOTE and random under-sampling of majority classes.

Synthetic minority oversampling (SMOTE) over samples the minority class by generating synthetic samples, unlike oversampling by replicating data. SMOTE operates on feature space instead of data space; it uses the kNN approach [29] and creates new samples on the line segment, joins all or any of the k nearest neighbors based on the chosen oversampling rate. [30] In our case, we over-sampled the X- class and down-sampled C/B and Q classes to the level of M- class. This brings our dataset to the size of 15440 with 3860 samples in each category. This dataset will be referred as SMOTE_RUS dataset in rest of the thesis.

To get a general idea about how well the data of each class is separated, we used principal component analysis (PCA) to project the original dataset in a two-dimensional plane. PCA computes principal components by performing eigen decomposition on the covariance matrix of the features from the dataset while minimizing the loss of information. Looking at the figure 2.5, we can undoubtedly say that the data is not linearly separable. The figure shows the derived data sets their class distribution and projection in the 2-D feature space.



Original Dataset

Class	X	M	C/B	Q
Samples	385	3860	32903	176872

SMOTE (X) + Random under sampling (C/B, Q) (SMOTE_RUS)

Random under sampling (M, C/B, Q) (RUS)

Class	X	M	C/B	Q
Samples	3860	3860	3860	3860

Class	X	M	C/B	Q
Samples	385	385	385	385

Fig. 2.5: 2-D projections of the original and derived data sets to visualize the separability between the data samples of each class. It looks quite obvious that classes are not linearly separable. The SMOTE plus random under sampling dataset has over-sampled X class and under-sampled C/B and Q classes while the random under sampling dataset has under-sampled M, C/B and Q classes. Also shows that the derived data sets have done decent job in maintaining the original distribution.

ID	AR Parameter	Description
0	TOTUSJH	Total unsigned current helicity
1	TOTBSQ	Total magnitude of Lorentz force
2	TOTPOT	Total photospheric magnetic energy density
3	TOTUSJZ	Total unsigned vertical current
4	ABSNJZH	Absolute value of the net current helicity in G2/m
5	SAVNCPP	Sum of the absolute value of the net current per polarity
6	USFLUX	Total unsigned flux in Maxwells
7	TOTFZ	Sum of Z-component of Lorentz force
8	MEANPOT	Mean photospheric excess magnetic energy density
9	EPSZ	Sum of Z-component of normalized Lorentz force
10	SHRGT45	Area with shear angle greater than 45 degrees
11	MEANSHR	Mean shear angle
12	MEANGAM	Mean inclination angle
13	MEANGBT	Mean value of the total field gradient
14	MEANGBZ	Mean value of the vertical field gradient
15	MEANGBH	Mean value of the horizontal field gradient
16	MEANJZH	Mean current helicity
17	TOTFY	Sum of Y-component of Lorentz force
18	MEANJZD	Mean vertical current density
19	MEANALP	Mean twist parameter
20	TOTFX	Sum of X-component of Lorentz force
21	EPSY	Sum of Y-component of normalized Lorentz force
22	EPSX	Sum of X-component of normalized Lorentz force
23	R_VALUE	Total unsigned flux around high gradient polarity inversion lines using the Br component
24	RBZ_VALUE	Derived
25	RBT_VALUE	Derived
26	RBP_VALUE	Derived
27	FDIM	Derived
28	BZ_FDIM	Derived
29	BT_FDIM	Derived
30	BP_FDIM	Derived
31	PIL_LEN	Derived
32	XR_MAX	Derived

Table 2.2: 33 Active region magnetic field parameters which are our features in multivariate time series dataset. Each of these parameter is a time-series captured with a cadence of 12 minutes. Our MVTS dataset has 60 instances of each feature in every sample

2.4 Data preprocessing

As the dataset is thoroughly validated and made machine learning ready [1], we only made the basic checks, such as imputing the missing values and removing the outliers. We filled them with the corresponding feature's mean in the time series with missing values. We got rid of the XR_MAX feature as it consisted of many outliers and its univariate performance was also very poor. Univariate performance evaluation is discussed in the next section.

The AR features have values ranging in different scales and huge magnitudes. Hence, each feature has been z-normalized (standard score) across the dataset before sending them as input to the machine learning models. Standard score normalization allows us to compare raw scores from two different normal distributions while considering both mean values and variance. In our experiments, Z-normalization helped neural networks avoid exploding gradient problems and speed up the computation by reducing rounding errors and expensive operations.

2.5 Feature Selection

Now that we have our data samples ready and classes are finalized we can now jump into feature selection. In this section we aimed to quantify the importance of features using few statistical and performance based tests. Based on our tests few features turned out to be useless and only added noise to the ML systems. To calculate each feature's contribution, we performed a statistical test, which is independent of the machine learning models and two machine learning-based performance tests using Random Forests and Convolutional Neural Networks. Our goal here was to find the best-performing minimal feature set.

Our focus in this feature selection pipeline was to quantify the univariate importance of derived parameters that introduced each parameter's weighting based on their distance from the polarity inversion line on the active region the parameter were collected. As these parameters were experimental, their contribution to the models is skeptical.

- Mutual Information Analysis: Mutual Information between two variables captures both linear and non-linear relationships. For example, if X and Y are two variables, let us take X as one of the 33 features and Y as our target class (X- flare). Suppose X and Y are independent variables, then the mutual information score will be 0. If there exists a deterministic function between X and Y, then the mutual information score will be one. For other cases, mutual Information will be between 0 and 1. The below bar chart shows us where each feature stands based on the mutual information 2.6.

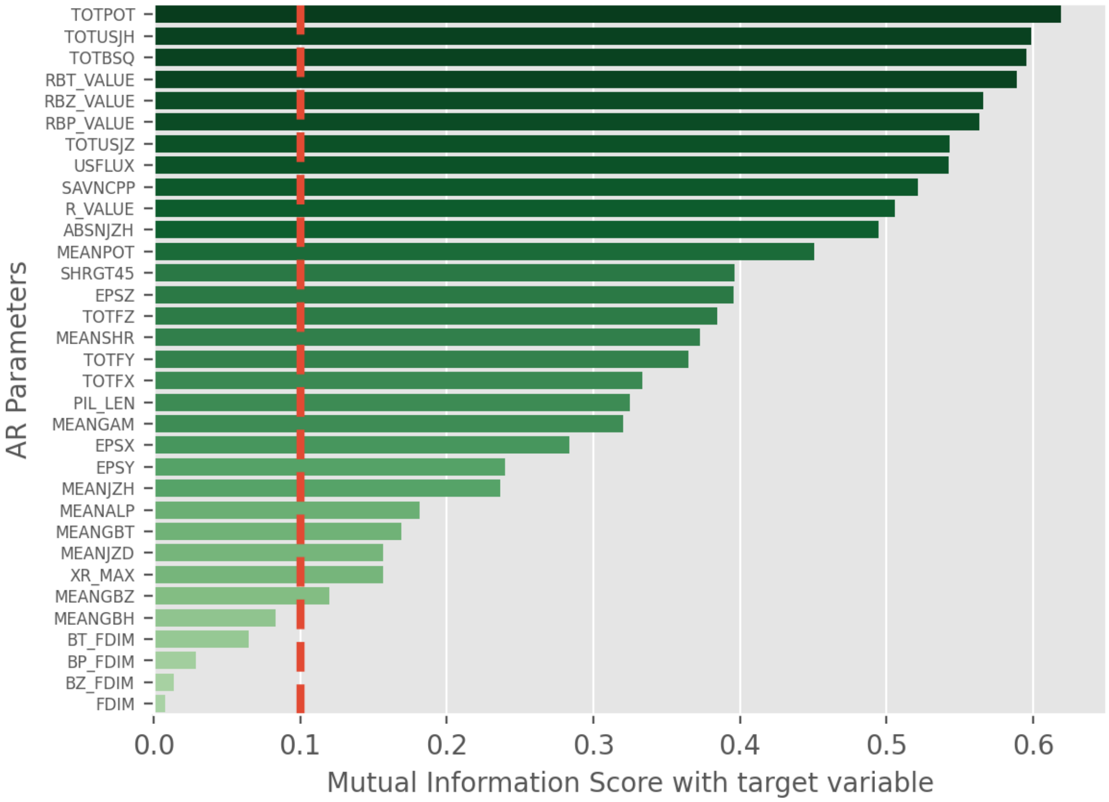


Fig. 2.6: Mutual Information scores of individual AR parameters, the red line is our threshold below which we doubt the contribution of the feature.

The following five features were on the radar to be removed as their mutual information contribution is limited and under the lenient threshold of 0.1

- FDIM
- BZ_FDIM
- BP_FDIM
- BT_FDIM
- MEANGBH

However, before removing these features to get more evidence and confirm that these features are underperforming or not adding any value to the remaining features, we have run experiments on each univariate feature with random forest and convolutional neural networks to rank the features based on classification performance.

A simple convolutional neural network with 4 and 8 filters and a random forest with 50 decision trees has been trained with each univariate feature individually. Performance results are captured by training and testing the models using 10-fold cross-validation. The results in figures 2.8 and 2.7 show that four of the five features sidelined based on mutual information scores also appeared here at the table's bottom except for the MEANGBH feature.

As a final step in our feature selection process, we performed a backward recursive feature elimination technique to ensure that the removal of the features is not affecting the classification performance. In this procedure, we remove the low-priority features one by one based on a priority order until there is only the highest priority feature left. Ideally, this test plots an elbow curve showing the peak value at an optimal number of features.

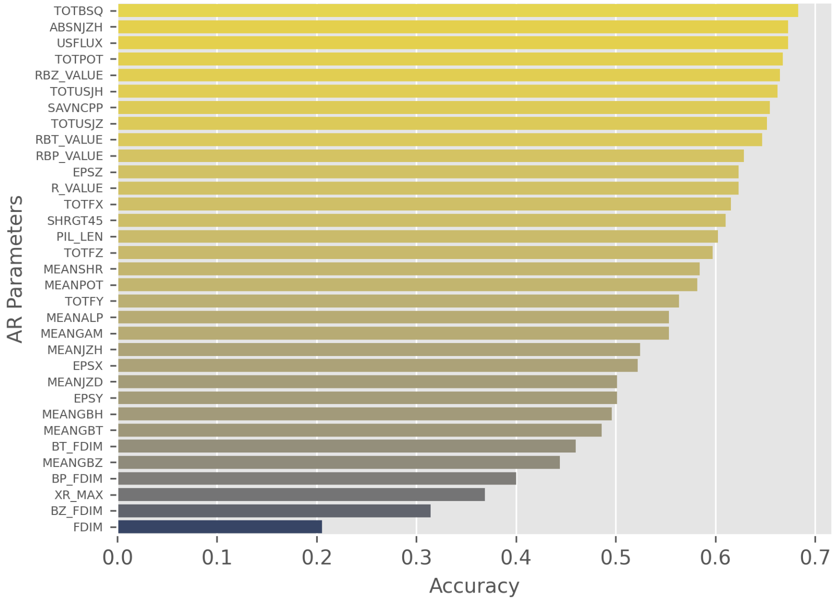


Fig. 2.7: Random Forest Performance on individual AR parameters, it shows the same features as mutual information test as low performing except for MEANGBH.

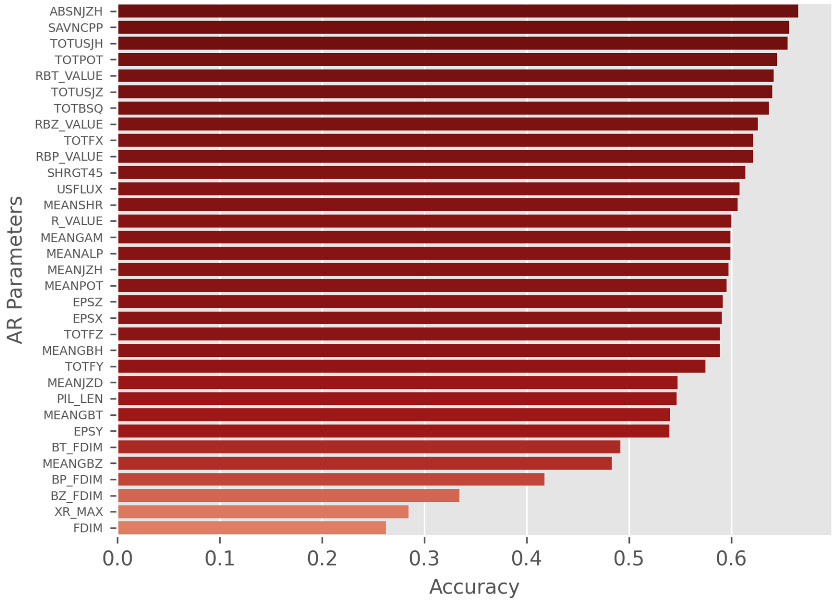


Fig. 2.8: Neural Network Performance on individual AR parameters, it shows the same features as mutual information test as low performing except for MEANGBH.

Stating that our hypothesis of removing the four features (FDIM, BZ_FDIM, BP_FDIM, BT_FDIM) is true, the elbow curve has peaked when these four features are removed from the dataset. We have implemented this backward recursive feature elimination on random forests and neural networks.The fig 2.9 and table 2.3 shows the results of these tests.

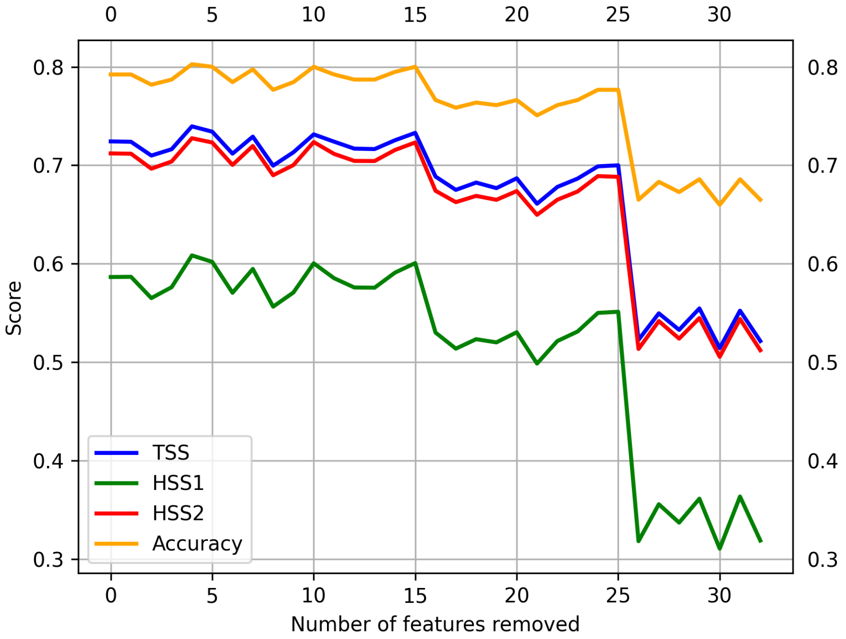


Fig. 2.9: Backward recursive feature elimination showing that the Random Forest model achieved the peak accuracy of 80.1% when four features are removed

Metric\Number of Features	**32**	**28**	**27**	**20**	**14**
Accuracy	0.785	0.795	0.770	0.735	0.729
HSS1	0.570	0.591	0.541	0.471	0.458
HSS2	0.701	0.717	0.681	0.627	0.618
TSS	0.703	0.718	0.682	0.629	0.621

Table 2.3: Performance metrics of neural networks with number of features as variable. Left to right features are removed based on the mutual information priority. With 28 features the neural networks gave ts peak performance.

Based on the above experiments, we can say that model performance has slightly increased with removing features; adding those four features does not benefit. Hence, we decided to remove the four features FDIM, BZ_FDIM, BP_FDIM, BT_FDIM, along with the XR_MAX, which is causing outliers. The final datasets (RUS and SMOTE_RUS) we are going to use for implementing machine learning models will have a reduced feature space of 28 features instead of 33.

CHAPTER 3

METHODS AND EXPERIMENTS

As we now developed a thorough understanding of the solar flare prediction problem and the data sets. This chapter will discuss machine learning algorithms that were used in our experiments along with training and testing strategies and performance evaluation techniques. All the experiments in thesis can be classified as RUS based and SMOTE_RUS based. First we experimented on different machine learning models using the RUS dataset and next the top six performing architectures/models were experimented using the SMOTE_RUS dataset. The chapter will be structured as mentioned below.

- Performance Evaluation Metrics
- Training and Testing Strategy
- Traditional Machine learning Algorithms:
  - Logistic Regression
  - Support Vector Machines
  - Random Forests
- Neural Network Architectures:
  - Multi layer Perceptron (MCP)
  - Recurrent Neural Networks with Long short term memory(LSTM) units
  - Convolutional Neural Networks
- Time Series Classifiers:
  - Time Series Forest
  - Symbolic Aggregate approXimation (SAX) Vector space model (VSM)
  - Shapelet Transforms

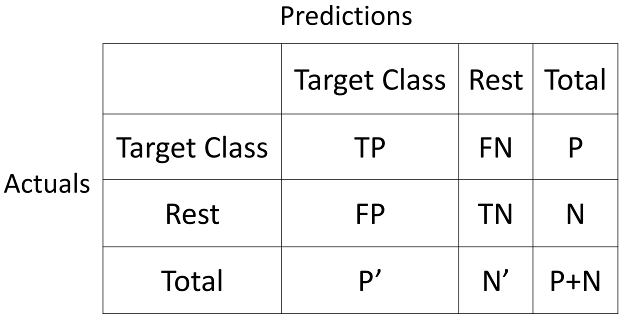
3.1 Performance Evaluation Metrics

Metrics help analyze the performance of the model. After data cleaning, Feature selection, and training the model, we get the results as a probability or class. It's essential that we choose the right metrics to evaluate our models based on the nature of the data, such as the consequences of false negatives or false positives. For example, In a model that tells us whether to buy a specific stock, we can't afford a false positive because investing in an unprofitable venture leads you to lose. Hence, in this case, precision is more important than recall. Similarly, in COVID detection, false negatives can't be tolerated as they pose a massive risk to the patients and other people in contact with the person. Hence, in this case, the recall will be our chosen metric.

In Solar Flare prediction, the false negatives can cause crucial damage to the earth and space resources; consequences such as satellite destruction or collapse of power grids may occur, while false positives can also trigger satellite maneuvering or expensive preventive measures. However, not detecting the flares will definitely because more damage than false alarms. Hence recall of flaring activity will be crucial in deciding the suitable model. Below are the standard metrics we are using to evaluate our models.

As our prediction has more than two outcomes, we followed the one vs. rest procedure to find performance measures for each class and then calculated the weighted average to find the metric for the model as a whole. The following quantities required to calculate metrics are captured from confusion matrix 3.1.

- True Positives (TP) and False Positives (FP)
- True Negatives (TN) and False Negatives (FN)
- Positives (P) and Negatives (N)
- Predicted Positives (P') and Predicted Negatives (N')



Predictions

		Target Class	Rest	Total
Actuals	Target Class	TP	FN	P
	Rest	FP	TN	N
	Total	P’	N’	P+N

Fig. 3.1: Confusion Matrix

- The Recall represents the models ability to classify a positive class as positive.

$$\frac{TP}{TP+FN}$$

- The Precision represents the model’s ability to not classify a negative class as positive.

$$\frac{TP}{TP+FP}$$

- The F1-score is the harmonic mean of precision and recall of a classifier. This metric comes into picture when you can’t decide upon choosing a model based on recall or precision.

$$\frac{2 * Precision * Recall}{Precision + Recall}$$

- The Accuracy allows us to quantify the total number of correct predictions made by a classifier. This metric often fails to handle the class imbalance problem as it gets biased to the class with more samples.

$$\frac{TP+TN}{TP+TN+FP+FN}$$

Except for the recall of other metrics, the metrics mentioned above are not very suitable for this task. Hence, we decided to calculate skill scores [31] which are being widely used by the solar flare prediction community [17] [16] [5]. They are the following:

- Heidke Skill Scores (HSS) is present in two versions; HSS1 is defined by Leka and Barnes [32]; it ranges from $-\infty$ to 1. A score closer to 1 represents a perfect forecast, while 0 illustrates that the model predicted everything as a negative class. While negative scores denote that performance is worse than anticipating everything under negative class. It measures forecast over predicting that no flare will ever occur. This measure makes sense, as solar flares are rare events.

  $$HSS_1 = \frac{TP + TN - N}{P}$$

  HSS2 is another skill score which measures the improvement of prediction over random predictions [5] [33]. This metric is defined by the Space weather prediction center. [17] Score ranges from 0 to 1; closer to one better the forecast.

  $$HSS_2 = \frac{2 * [(TP * TN) - (FN * FP)]}{P * N' + N * P'}$$

- Gilbert Skill Score (GS) [5] [33] takes True positives obtained by chance into consideration. This helps one to arrive at model which is not predicting the flares by chance.

  $$GS = \frac{TP * (P + N) - P * P'}{FN * (P + N) - N * P'}$$

- True Skill Statistic (TSS) is a metric independent of the class imbalance ratio. This is the most desired metric in the solar flare prediction community, as it converges to 1 when the false alarm rate and the false-negative rate are minimized. Flare prediction demands a very similar metric; we neither want to create chaos unnecessarily or miss a flare and put humans at risk. [17] [34] [5].

  $$TSS = \frac{TP}{P} - \frac{FP}{N}$$

Every skill score has its characteristic which can be helpful to analyze. However, HSS1, HSS2, and GS are still somewhat prone to class imbalance ratio [5]. Hence, in the later sections of results and discussion, we will be analyzing the performance more in terms of TSS and Recall.

3.2 Training and testing strategies

To handle the class imbalance problem, we have under-sampled our dataset to 1540 MVTS. As most machine learning models are data-hungry, standard train-test splits such as 60-40 or 70-30 would leave the model with a shortage of data either for validation or training. Hence, we decided to perform stratified 10-fold cross-validation.

Stratified Cross-validation uses a stratified sampling technique to divide the data into k folds. Stratified sampling ensures the same proportion of samples are present in each fold. This method helps us to approximate the generalization error using cross-validation closely. Also, no sample will be over or underrepresented in the training and testing sets. It follows the below procedure:

- Randomly Shuffles the entire dataset.
- Splits the dataset into k groups with each group having same class concentration.
- Every group is once held out as a testing group (Test Set).
- Remaining groups are used to train the model (Train set).
- Train the model with train set and evaluate with test set.
- Re-initialize the model to train with all of the k combinations.
- Performance of model is retained after evaluating model with each test set.

3.3 Traditional Classifiers

We performed experiments on the traditional classifiers to establish a baseline for Neural Network models and Time series classifiers. These classifiers can handle only univariate time series data. Hence, we flattened the 28 features into one extended feature vector, partly sacrificing the time-series nature of the data following the work of Bobra [17] where he used a similar kind of vector magnetogram data.

Our task is a 4 class classification problem, and the class data is non-overlapping. The train and Test Splits are z-score normalized for all the classifiers before flattening and sending them as input. All the machine learning model implementations in this section are using the Scikit-Learn library.

3.3.1 Support Vector Machines

SVM is a supervised machine learning algorithm known for classification tasks. SVM plots the N-dimensional data and aims to find the optimal (N-1) hyperplane possible that can split the dataset into different groups [35]. If there are only two features, then the hyperplane becomes a line. The two hyper-parameters we need to adjust are C, Gamma γ and the kernel type. C is the penalty imposed for each misclassification; the lower the C, the more tolerant to misclassifications and the better it generalizes. Gamma is the curvature amount that we need to tune for only the RBF kernel.

In our experiments with SVM we performed a grid search with C ranging from 0.0001 to 10 and gamma from 0.0001 to 1. We used this search space for rbf and poly except for linear kernel which doesn't need gamma to be adjusted. The radial basis function is our best-performing kernel with a C value of 0.1 and Gamma of 0.01.

3.3.2 Multinomial Logistic Regression

Logistic regression gives an estimated probability as output; it tells the likelihood of the predicted class. It fits the below sigmoid curve to solve this optimization problem. Logistic regression has a set of N weights for N features which it optimizes using a logarithmic cost function via gradient descent algorithm. We experimented with the following solvers: newton-cg, sag, saga, and lbfgs; with C ranging from 0.0001 to 1. Newton-cg solver resulted in the most generalized model with a C value of 0.1. We used the L2 penalty for misclassifications. Below are the equations of hypothesis function and the Cost function of Logistic regression.

$$h_\Theta(X) = \frac{1}{1 + e^{-(a_0 + a_1 X)}}$$

$$J(\Theta) = -\frac{1}{m}\Sigma_{i=1}^{m}[y^{(i)}log(h_\Theta(x(i)) + (1 - y^{(i)})log(1 - h_\Theta(x(i)))]$$

3.3.3 Random Forests

This is one of the machine learning methods that hasn't been explored much in this solar flare prediction domain. Random Forests operate using an ensemble of decision trees for categorical and continuous response variables. Random Forests is likely the most interpretable machine learning method [36–38] and also performs on par with the Neural Networks in most cases, especially with numerical data; they are relatively speedy to train and also demand less computational power [39]. With only two hyper parameters, it's easy to reach the optimal model. They can be used directly with high-dimensional data and be easily implemented in parallel.

In our thesis, we used random forests as a baseline approach and calculated feature importance (discussed in the previous chapter). We experimented with different decision trees from 10 to 300 to find the optimal model.

3.4 Neural Networks

Neural networks(NN) or Deep learning is essentially a subset of machine learning. They comprise an input layer that accepts input features, an output layer, which gives prediction results, and an arbitrary number of hidden layers, making the core computational part of NN. Layers consist of nodes; each node can be treated as a single computational unit with a threshold over which it is activated. All the layers are fully connected, and every connection between nodes is associated with a weight.

To train the NN, we pass the data through the network. Based on the error between the ground truth and the predicted result, the back-propagation algorithm performs a backward pass to adjust the network's weights and biases. This back and forth process is continued until NN gradually learns the correlations and dependencies between input and output variables to minimize the error.

Neural networks are primarily used in a supervised learning setting. There are several variations of NN used for different purposes. Our thesis has used three types to perform this 4-class classification task.

3.4.1 Training strategies implemented for the neural networks:

Below mentioned methods are applied throughout the training, and each model is trained for a maximum of 1000 epochs.

1) Model Checkpoint: During training, we continuously monitored the validation accuracy to save the model with the best performance ultimately.

2) Early Stopping: Though we set the maximum epochs to 1000. we stopped the training of models if the validation loss doesn't reach a new minimum for 50 epochs. This helps us to avoid over-fitting the model and training time.

3) Reduce learning rate on the plateau: Usually, when the learning of the model stagnates, they get benefited by reducing the learning rate a little bit. We reduced the learning rate

if the validation loss didn't decrease for 30 epochs.

4) Dropout: We implemented a dropout of 0.5 to the fully connected layers when we observed the over-fitting model. In ANN and LSTM this came to an excellent use to generalize.

5) Batch Normalization: Applying batch normalization helped us save some crucial amount of training time as it helped the model converge quickly. It also adds a bit of regularization. Especially in ANN, it seemed to have improved the performance slightly.

3.4.2 Multi-layer Perceptron

It is a basic feed-forward Neural Network that accepts only vector data as input. Hence, we flattened the feature space(28*60) into a vector(1680) to train it. We started with one hidden layer while experimenting with the learning rate, batch size, and layer size. In our search space, below architecture mentioned was our best performing one.

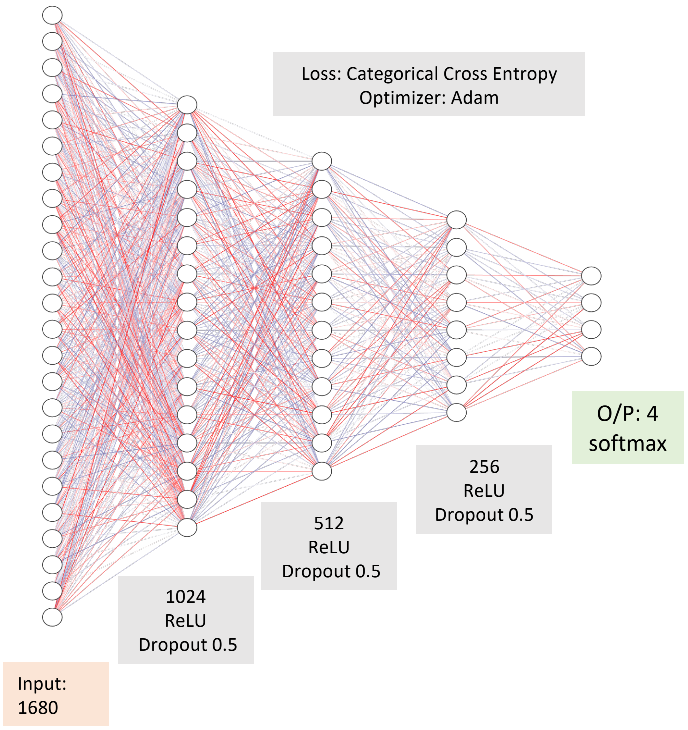


Fig. 3.2: ANN architecture that achieved the best performance

3.4.3 Convolutional Neural Network

CNN's are known for learning visual Imagery data. They can accept n channel 2-D data as input. It uses convolutional layers to learn several features from the input data such as edges, color, etc. also captures several high-level features as we increase the number of layers. It uses pooling layers to reduce the dimensionality of convoluted features. We used three convolutional layers with different kernel sizes and a max-pooling strategy.

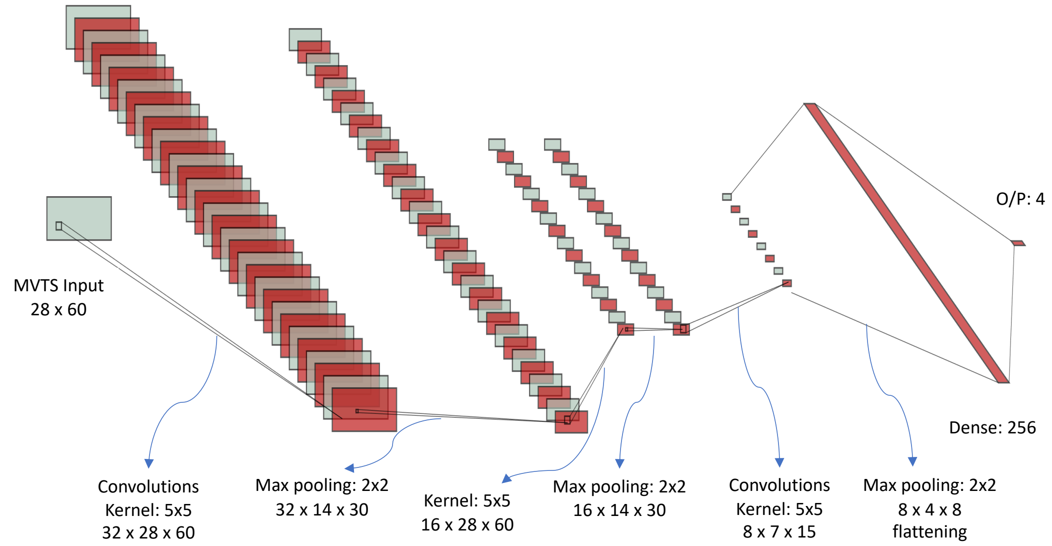


Fig. 3.3: CNN architecture that achieved the best performance

3.4.4 Recurrent Neural Network LSTM

Recurrent Neural Networks can accept input of arbitrary lengths. They have hidden states which remember the past that influences the decision-making of RNN. We can also configure the RNN as bidirectional, where it also remembers the future states. RNNs are especially helpful with time-series data. Unlike vanilla RNNs, LSTM networks use additional gates and hidden units to maintain the information for longer periods allowing them to learn long-term dependencies. Our best LSTM architecture used 2 LSTM layers with 100 hidden units each. Below figure 3.4 is the architecture used.

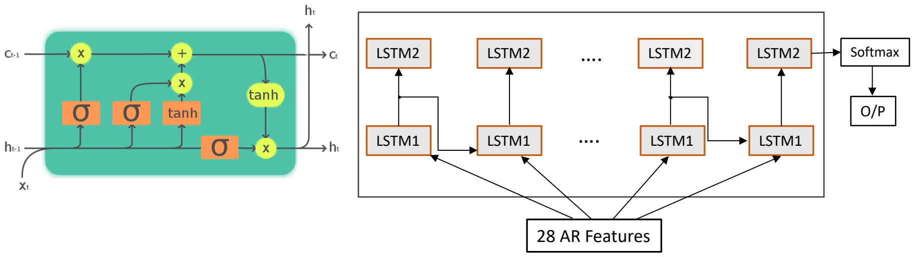


Fig. 3.4: LSTM architecture that achieved the best performance, image on the left side is the internal structure of LSTM unit.

3.5 Time Series Classifiers

Time series data is found everywhere globally, ranging from health care, finance, marketing, stocks, disease detection, etc. As the need for handling temporal data for several applications increases, many new algorithms have been explicitly proposed to deal with time-series data. Among them, interval-based, dictionary-based, and shapelet-based are the three primary classifiers. In our thesis, we decided to explore one technique from each category.

These time series classifiers have a feature engineering layer before performing classification, unlike deep learning and most of the other machine learning algorithms which internally do feature engineering. This extra layer increases the development time. However, they are more interpretable and sometimes achieve better results. They are known to surpass the classical nearest neighbors with dynamic time warping techniques in several applications. [40]

3.5.1 Time-series Forests (TSF)

This is an interval-based classifier that adds a statistical feature extraction scheme on top of random forest implementation. It reduces the dimension of time-series data by randomly extracting intervals and replacing them with mean, standard deviation, and slope. They combine the entropy used by random forests and distance measures obtained from statistical measures to take decisions [41]. The figure3.5 below describes how the input data is transformed into a reduced feature space and trained on random forests. In our experiments, TSF has been experimented with using different number of decision trees. As it adapts to the random forests, hyperparameter tuning became more straightforward with fewer parameters, and we were able to train them quickly. TSF with 150 decision trees has reported the best accuracy with the slightest variance in folds.

3.5.2 Symbolic Aggregate ApproXimation and Vector Space Model (SAXVSM)

This is a dictionary-based classifier, as the name suggests, it converts input time series data into a set of words by using a bag of words concept on top of symbolic aggregate approximation (SAX). SAX converts the time series of length l to a string of length s, where s << l. It requires the input time series to be normalized and transformed using a piece-wise aggregate approximation. As the data is in Gaussian distribution, it now divides the data magnitude into equal-sized areas and assigns an alphabet to each bin.

Once the SAX sentences are extracted from the multiple binning approaches, SAXVSM generates word frequency matrices for each time-series class and classifies the new time-series based on the cosine similarity. Figure 3.6 visualizes the process in detail. In our experiment we varied the number of bins, window size and word size to find the optimal parameters. However, SAXVSM's heavy preprocessing restricts us from widening the hyper parameters search space. Window size of 0.25%, word size of 0.075% and 24 bins have given us the best accuracy in our search space.

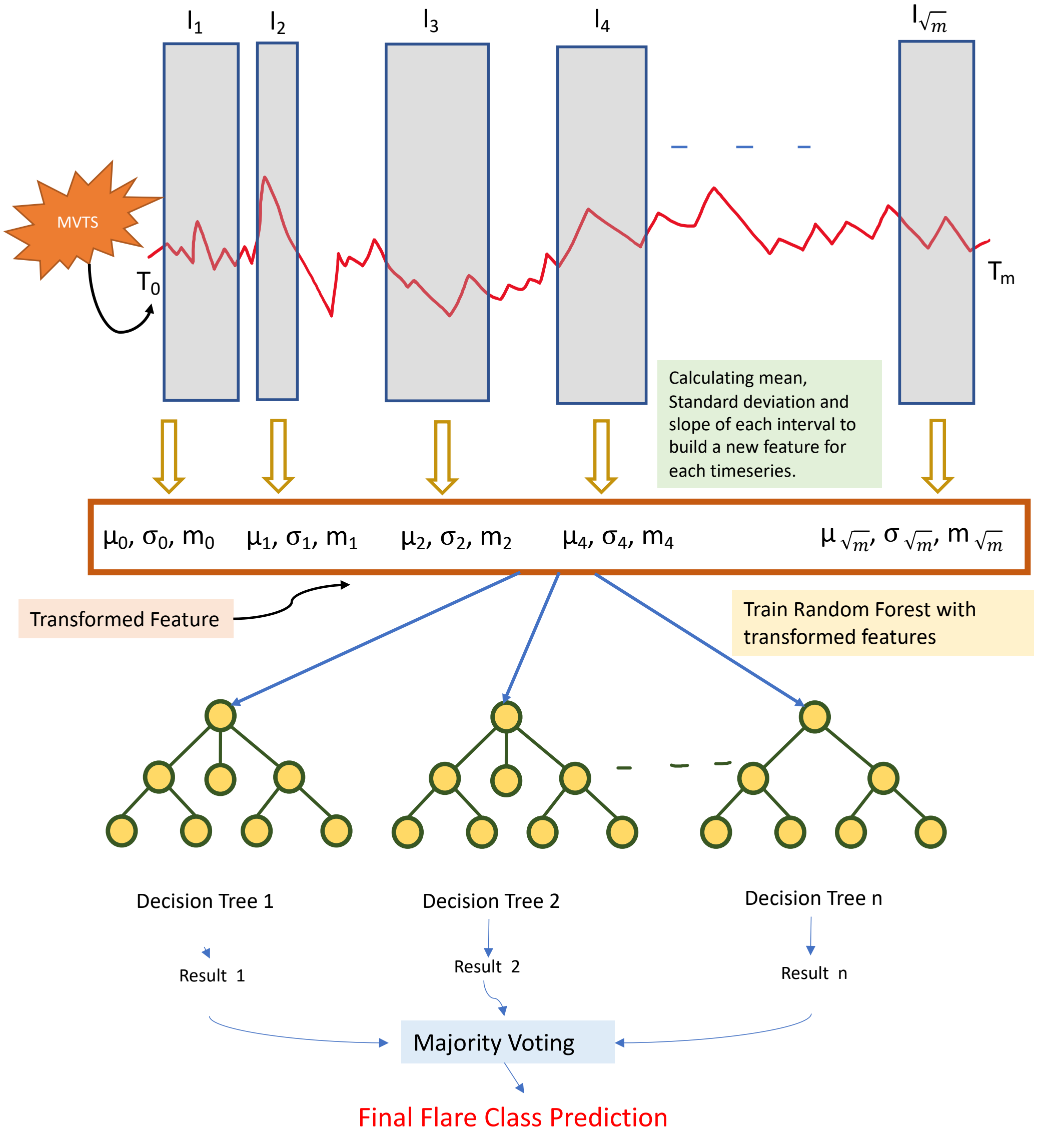
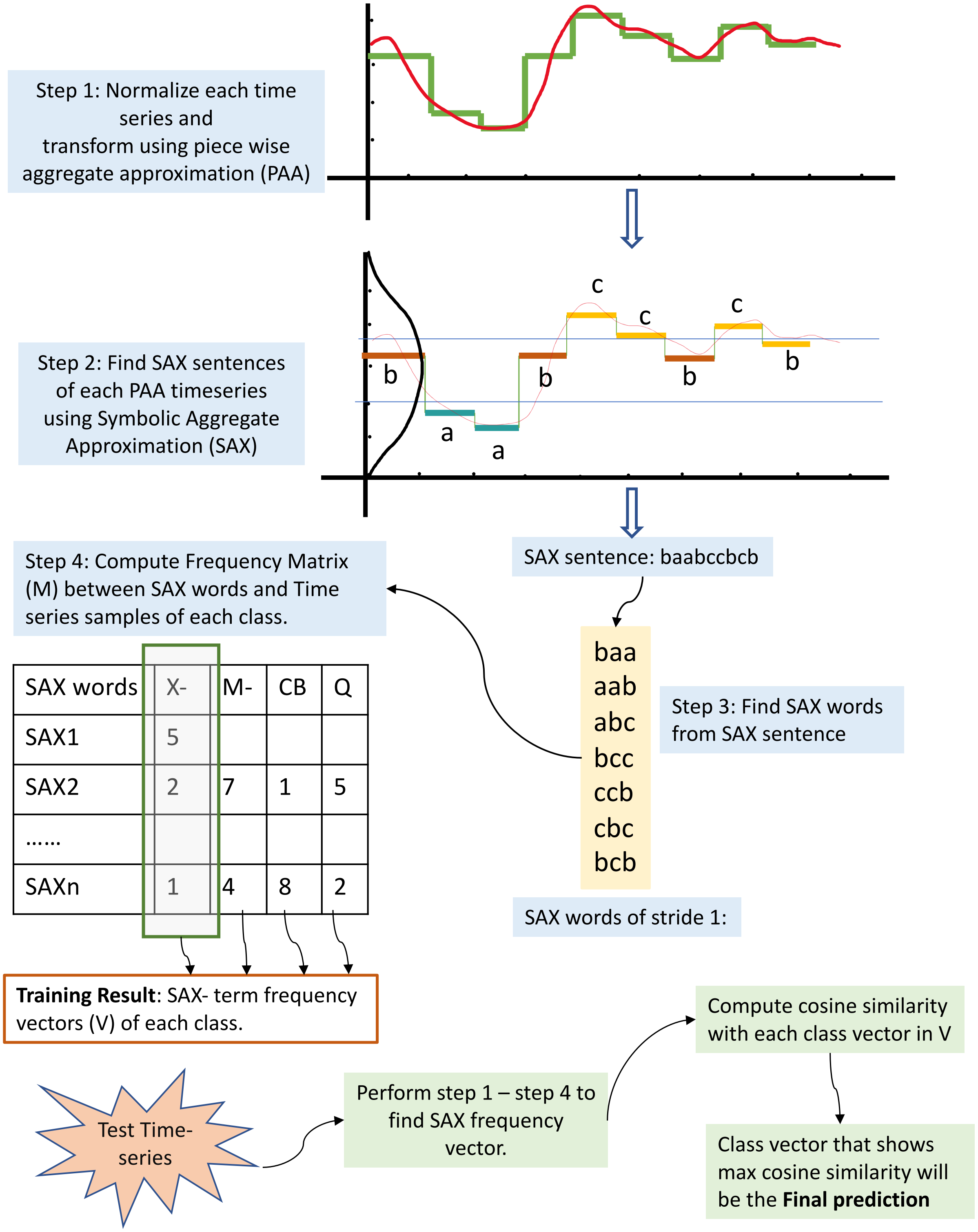

I_1
I_2
I_3
I_4
$I_{\sqrt{m}}$
MVTS
T_0
T_m
Calculating mean, Standard deviation and slope of each interval to build a new feature for each timeseries.
μ_0, σ_0, m_0 μ_1, σ_1, m_1 μ_2, σ_2, m_2 μ_4, σ_4, m_4 $\mu_{\sqrt{m}}, \sigma_{\sqrt{m}}, m_{\sqrt{m}}$
Transformed Feature
Train Random Forest with transformed features
Decision Tree 1
Decision Tree 2
Decision Tree n
Result 1
Result 2
Result n
Majority Voting
Final Flare Class Prediction

Fig. 3.5: Time series Forest algorithm; explains how each time series is transformed into a compressed feature space using a interval based approach and fed to the random forest


Step 1: Normalize each time series and transform using piece wise aggregate approximation (PAA)
Step 2: Find SAX sentences of each PAA timeseries using Symbolic Aggregate Approximation (SAX)
c
c
c
b
b
b
b
a
a
SAX sentence: baabccbcb
baa
aab
abc
bcc
ccb
cbc
bcb
Step 3: Find SAX words from SAX sentence
SAX words of stride 1:
Step 4: Compute Frequency Matrix (M) between SAX words and Time series samples of each class.
Training Result: SAX- term frequency vectors (V) of each class.
Test Time-series
Perform step 1 – step 4 to find SAX frequency vector.
Compute cosine similarity with each class vector in V
Class vector that shows max cosine similarity will be the Final prediction

SAX words	X-	M-	CB	Q
SAX1	5			
SAX2	2	7	1	5
……				
SAXn	1	4	8	2

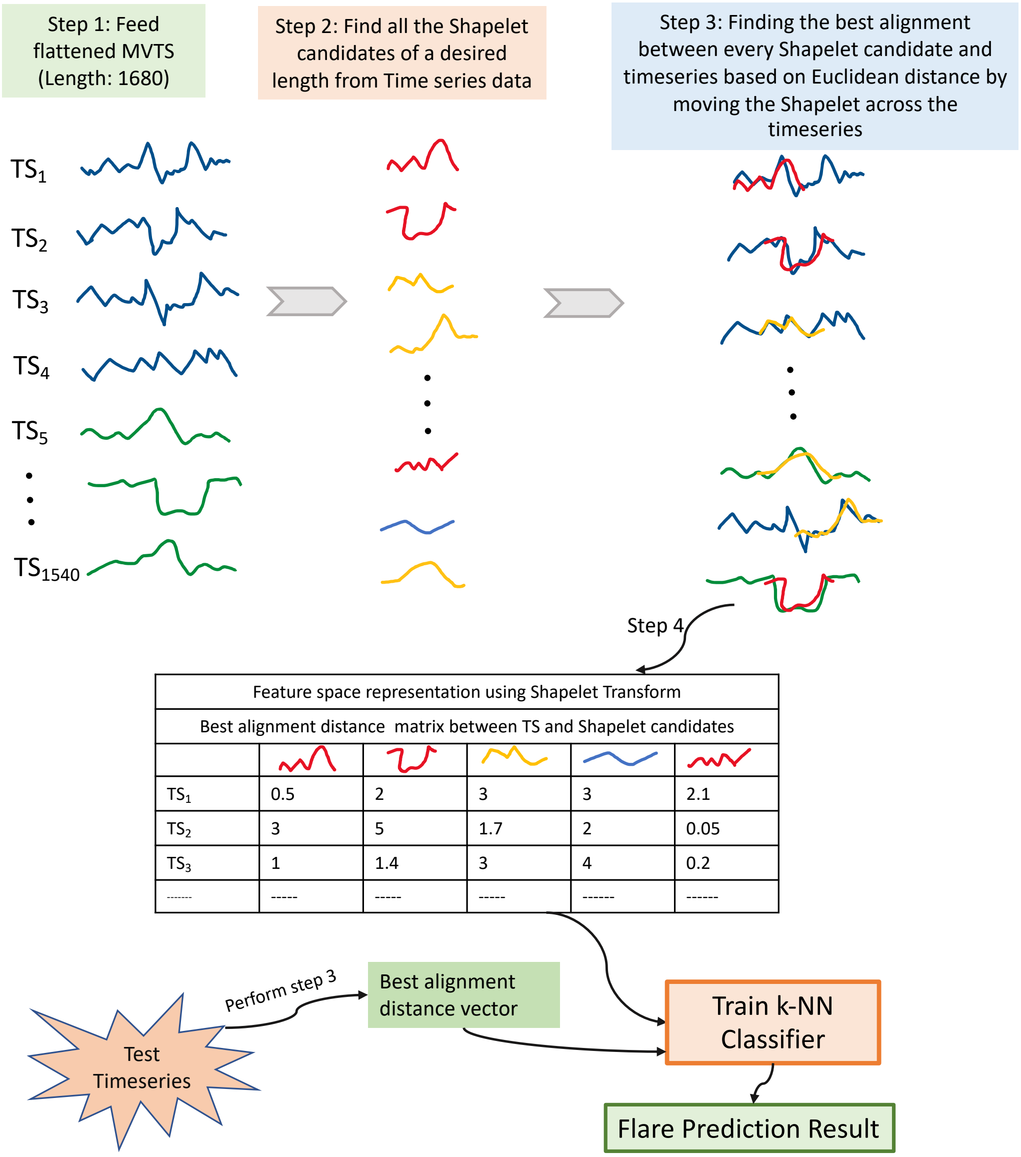
Fig. 3.6: Symbolic Aggregate Approximation and Vector Space Model; explains how the SAX dictionary of dataset in obtained based on which SAX term frequency vector of each class is derived.

3.5.3 Shapelet Transforms

This classifier comes under the family of shapelet-based classifiers. Shapelets are nothing but snapshots of a time series that represent a specific class. This algorithm aims to find the user-defined number of shapelets with the highest discriminatory power based on the alignment distance with each time series. It tries to identify specific shapelets whose presence or absence could help classify a time series into a particular class.

However, this is a greedy algorithm, and it's hard to find out the correct size and amount of shapelets required to achieve desired discriminatory power. The search space for window size and stride is directly proportional to the size of the time series, and in applications like ours, where feature space is vast, it demands heavy computational power. Shapelets extracted are used to build distance vectors with each time series and trained on a choice of the distance-based classifier. In our case, we used a kNN to teach a shapelet-based best alignment distance matrix. Figure 3.7 visualizes the classification procedure implemented clearly.

In our experiments with shapelet transforms we have varied the shapelets size from 10 to 100 and number of shapelets from 100 to 520 with an interval of 60 to construct a set of the best alignment distance matrices. Matrix calculated with 280 shapelets of length 30 have resulted us the highest accuracy with k-NN classifier with a k value of 5.


Step 1: Feed flattened MVTS (Length: 1680)
Step 2: Find all the Shapelet candidates of a desired length from Time series data
Step 3: Finding the best alignment between every Shapelet candidate and timeseries based on Euclidean distance by moving the Shapelet across the timeseries
TS1
TS2
TS3
TS4
TS5
TS1540
Step 4
Perform step 3
Best alignment distance vector
Test Timeseries
Train k-NN Classifier
Flare Prediction Result

Feature space representation using Shapelet Transform

Best alignment distance matrix between TS and Shapelet candidates

TS_1	0.5	2	3	3	2.1
TS_2	3	5	1.7	2	0.05
TS_3	1	1.4	3	4	0.2
-------	-----	-----	-----	------	------

Fig. 3.7: Shapelet based classification Algorithm; explains how the shapelets are extracted from the time series dataset and best alignment distance matrix is obtained to train a kNN dataset.

CHAPTER 4

RESULTS AND DISCUSSION

We applied a wide range of supervised machine learning algorithms to predict the flares occurring in the next 24 hours using the 28 AR features of the vector magnetogram data acquired from the GOES satellite. Classifiers are modeled to classify each data point (MVTS) into one of the four flaring levels. X- and M- flares pose a threat to humans, CB flares are essential for heliophysics studies, and Q represents that there is no flaring activity.

All the models are trained and tested in a stratified 10-fold cross-validation setting and evaluated using standard measures such as precision, recall, f1 score, and side domain-specific measures such as TSS, HSS1, and HSS2, and GS.

4.1 Prediction results on RUS dataset

This section discusses the results achieved on the dataset which is prepared using random under sampling. Among the nine classifiers we modeled in our thesis, seven of them have shown decent performance in predicting solar flares. Time series forest (TSF) is the best individual classifier, while Random Forests and Neural Networks have crossed the mark of 80% accuracy. The table 4.1 reveals the performance of each classifiers.

Classifier\Metric	**Accuracy**	**F1-score**	**Recall**	**Precision**	**HSS1**	**HSS2**	**GS**	**TSS**
ANN	0.811 (0.026)	0.808 (0.028)	0.811 (0.026)	0.81 (0.029)	0.622 (0.053)	0.739 (0.038)	0.605 (0.047)	0.74 (0.038)
CNN	0.831 (0.021)	0.83 (0.021)	0.831 (0.021)	0.836 (0.022)	0.661 (0.041)	0.768 (0.029)	0.641 (0.036)	0.769 (0.029)
LSTM	0.818 (0.023)	0.818 (0.024)	0.818 (0.023)	0.826 (0.025)	0.636 (0.047)	0.752 (0.033)	0.624 (0.04)	0.752 (0.033)
TSF	0.855 (0.023)	0.855 (0.023)	0.855 (0.023)	0.859 (0.022)	0.71 (0.047)	0.804 (0.032)	0.694 (0.043)	0.804 (0.032)
SAXVSM	0.558 (0.04)	0.563 (0.042)	0.558 (0.04)	0.64 (0.051)	0.117 (0.08)	0.377 (0.063)	0.253 (0.047)	0.372 (0.063)
ST	0.625 (0.034)	0.622 (0.03)	0.625 (0.034)	0.629 (0.029)	0.25 (0.068)	0.461 (0.049)	0.317 (0.04)	0.462 (0.051)
RF	0.825 (0.024)	0.824 (0.025)	0.825 (0.024)	0.83 (0.024)	0.651 (0.048)	0.762 (0.034)	0.637 (0.042)	0.762 (0.034)
SVM	0.773 (0.03)	0.773 (0.031)	0.773 (0.03)	0.799 (0.03)	0.545 (0.06)	0.689 (0.043)	0.547 (0.05)	0.688 (0.044)
LG	0.723 (0.028)	0.722 (0.026)	0.723 (0.028)	0.73 (0.028)	0.447 (0.056)	0.612 (0.04)	0.456 (0.042)	0.612 (0.041)

Table 4.1: Classifier Performance Analysis on the RUS dataset. Reported values are the mean of the 10-fold stratified cross-validation test accuracies. Value in the parenthesis denote the standard error of each measure across the folds.

With a mean TSS of 0.804 (SE=0.029) over a rigorous ten-fold cross-validation, the time series forest takes first place. CNN has the highest skill score of 0.769 (SE = 0.029) among the Neural Networks, followed by the LSTM network and ANN. Random Forest (TSS=0.762, SE=0.024) has clearly shown its superiority in the baseline approaches, while SVM (TSS=0.688, SE=0.044) and Logistic Regression (TSS=0.612, SE=0.041) are far behind.

Performance in terms of skill scores

TSS can be thought of as a stringent version of accuracy. It subtracts the false alarm rate and false-negative rate from one to reflect the model's worth. HSS1 shows how much better the model is than always predicting a negative class, while HSS2 shows how much better the model is than random prediction. HSS2 will be zero for a four-class problem if the accuracy is at least 25%. The Gilbert score (GS) takes into account true positives that happened by chance. HSS1 isn't appropriate because our issue isn't binary classification, and GS isn't as good as TSS or HSS2 because reducing false negatives is more relevant in our situation. Hence, our primary metrics will be TSS and HSS2. However, other metrics are also mentioned to enable comparison with the literature.

The performance pattern of models has remained consistent across various metrics, indicating that models have established the balance in multiple aspects, such as true positive rate, false alarm rate, precision, and recall. The HSS metrics that are vulnerable to class imbalance benefited from removing class imbalance. The box plots in 4.1 provide a broad overview of where models stand. Despite modeling the prediction task as a 4-class problem, we have achieved relatively better output with our TSF, CNN, ANN, and LSTM in comparison to Ji's [8] time-series forests and Nagem's [10] prediction using gramian angular fields.

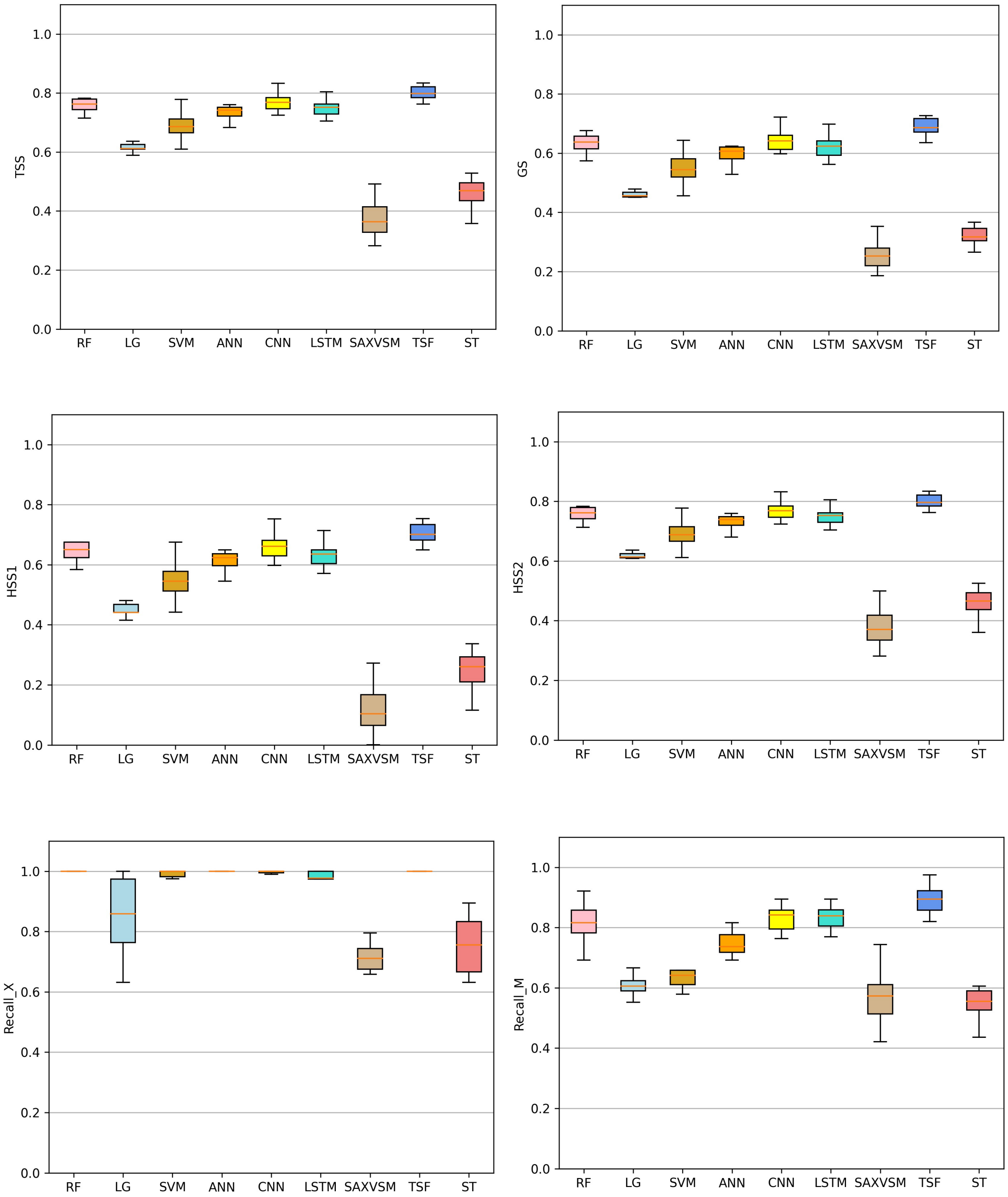


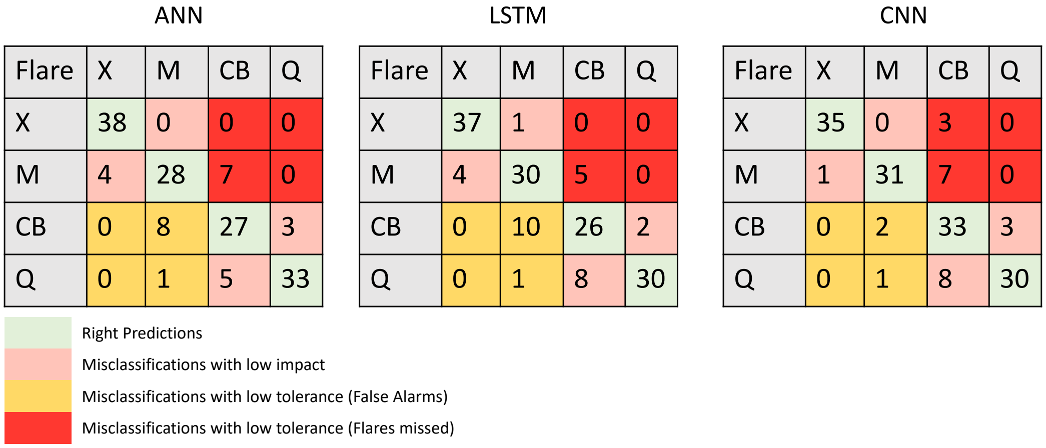
Fig. 4.1: Classifiers across stratified 10-fold cross validation setting. Smaller box plots represent more consistent performance of classifier and larger ones represent the inconsistency in performance in different folds.

How consistent are the models across the folds?

The low standard error and variance across the ten folds, TSF, RF, ANN, CNN, and LSTM, appear pretty robust. When looking at the box plots, it's clear that logistic regression has the smallest box plot but a relatively high standard error, suggesting outliers. Its prediction accuracy, on the other hand, is not particularly impressive. SVMs performed admirably but were unreliable because of their significant variance. On the contrasting side, SAXVSM and Shapelet Transform are nowhere close to the optimum behavior making it plain that they are under performers.

4.1.1 Performance of Neural Networks

The accuracy of all three Neural Network architectures was greater than 81 percent, with CNN reaching 83 percent. Let's look at the models' confusion matrices to understand the models' strengths and weaknesses better.



ANN

Flare	X	M	CB	Q
X	38	0	0	0
M	4	28	7	0
CB	0	8	27	3
Q	0	1	5	33

LSTM

Flare	X	M	CB	Q
X	37	1	0	0
M	4	30	5	0
CB	0	10	26	2
Q	0	1	8	30

CNN

Flare	X	M	CB	Q
X	35	0	3	0
M	1	31	7	0
CB	0	2	33	3
Q	0	1	8	30

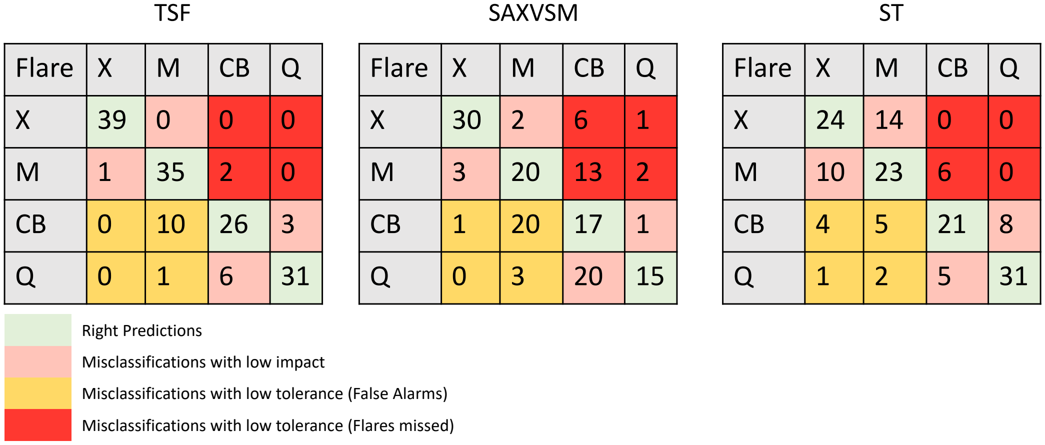
🟩 Right Predictions
▨ Misclassifications with low impact
🟨 Misclassifications with low tolerance (False Alarms)
🟥 Misclassifications with low tolerance (Flares missed)

Fig. 4.2: Class confusion matrix of the neural networks in median performing fold

The confusion matrices infer that all the neural networks classified the X-class flares to near perfection. The majority of the confusion is with CB flares; ANN and LSTM have identified triggered false alarms 8 and 10 times, respectively, by misclassifying CB flares as M class flares. They even classified Q class(Non-flaring) to M class a couple of times. On the other hand, they have missed identifying M class flares 7 and 5 times each. Their area of improvement is to learn M- and CB flares better. ANN has done a pretty good job identifying non-flaring time series data. Overall, ANN and LSTM seem to be biased towards X and M classes.

When it comes to CNN, they achieved a better balance between all of the classes. Notably, it did a better job of classifying CB flares than the other two networks. CNN is even more vigilant when it comes to false alarms. But on the other hand, it needs to boost its recall of M class flares. On the whole, CNN managed to keep the count lower in red and yellow regions of Confusion matrices to achieve a TSS of 0.769.

4.1.2 Performance of Time series classifiers



TSF

Flare	X	M	CB	Q
X	39	0	0	0
M	1	35	2	0
CB	0	10	26	3
Q	0	1	6	31

SAXVSM

Flare	X	M	CB	Q
X	30	2	6	1
M	3	20	13	2
CB	1	20	17	1
Q	0	3	20	15

ST

Flare	X	M	CB	Q
X	24	14	0	0
M	10	23	6	0
CB	4	5	21	8
Q	1	2	5	31

🟩 Right Predictions
🟧 Misclassifications with low impact
🟨 Misclassifications with low tolerance (False Alarms)
🟥 Misclassifications with low tolerance (Flares missed)

Fig. 4.3: Class confusion matrix of the time series classifiers in median performing fold

In this thesis, Time Series Forest is our best bet to perform the prediction task. It has clearly gained advantage with X and M class flares 4.3. It has only skipped reporting flares twice, indicating that it is our safest classifier. It has outperformed CNN in classifying non-flaring regions as well. However, on the downside, TSF has to reduce the false alarm rate to improve its TSS.

Below is the plot 4.4 showing the performance of TSF with a varied number of decision trees. The low time complexity of TSF has enabled us to find the optimal number of decision trees [11,42] by launching several experiments. We have chosen our TSF, which gave us the highest TSS while keeping the X- and M- flares recall at best; in the tie case, we opted for the model with the slightest variance across ten fold results.

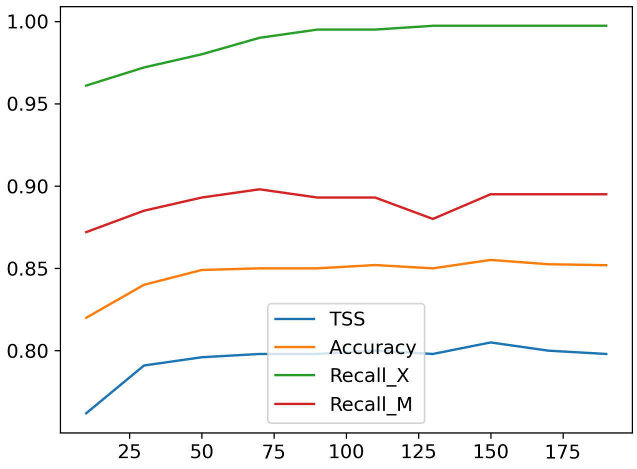


Fig. 4.4: Time series forests performance on varying the number of decision trees

Coming to SAXVSM, it has its factors for poor performance distributed all over the matrix. It has only managed to classify X- class flares slightly better. One of the reasons SAXVSM could not perform could be because some features are constant at most places. The bag of words(BOW) can't form distribution and create bins if the feature is constant in the sliding window, leading BOW to consider larger windows and word size. A large word size might cause it to miss minor variations in the time series, which could be crucial to distinguish between two flare classes.

Despite its poor results, Shapelet Transform has achieved classification accuracy on par with CNN for Q class flares. We believe that shapelet transforms being a greedy classifier; it needs vast number of shapelets with different window sizes to find the optimal subset for classification. This procedure seems to be quite computationally demanding, but exploring this technique would help us better interpret discriminative shapelets. Below fig 4.5

displays the shapelets that are considered most discriminative by the shapelet transforms. Tracing the shapelets back to the corresponding AR features reveals that RBT_VALUE, RBP_VALUE, RBZ_VALUE, TOTPOT could be crucial features to distinguish between X, M, CB and Q classes.

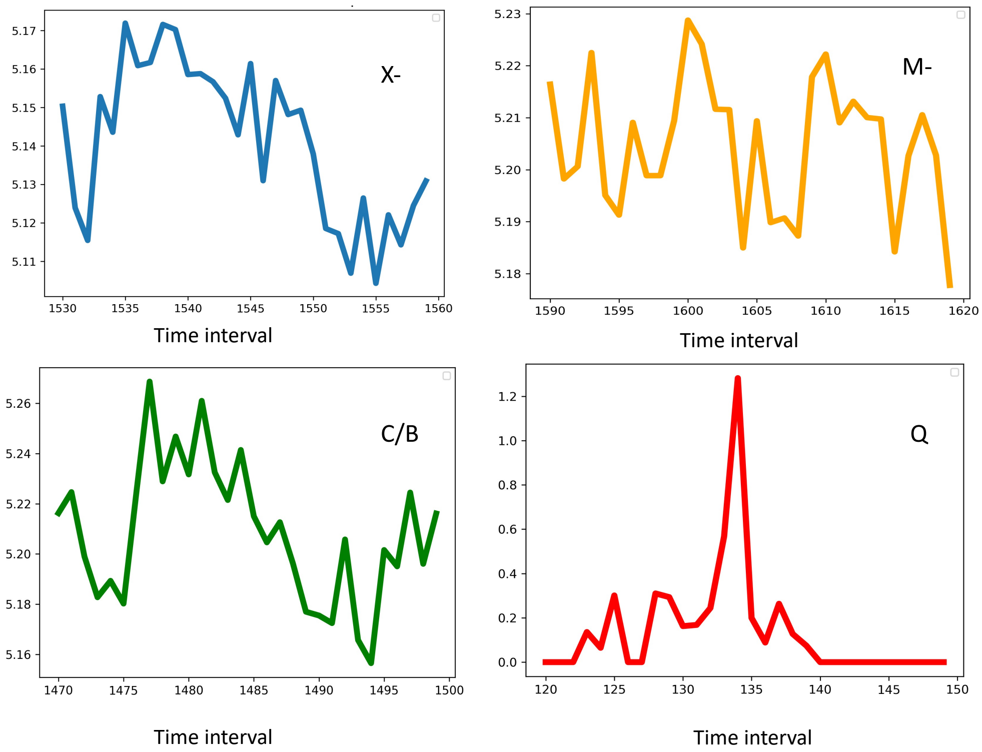
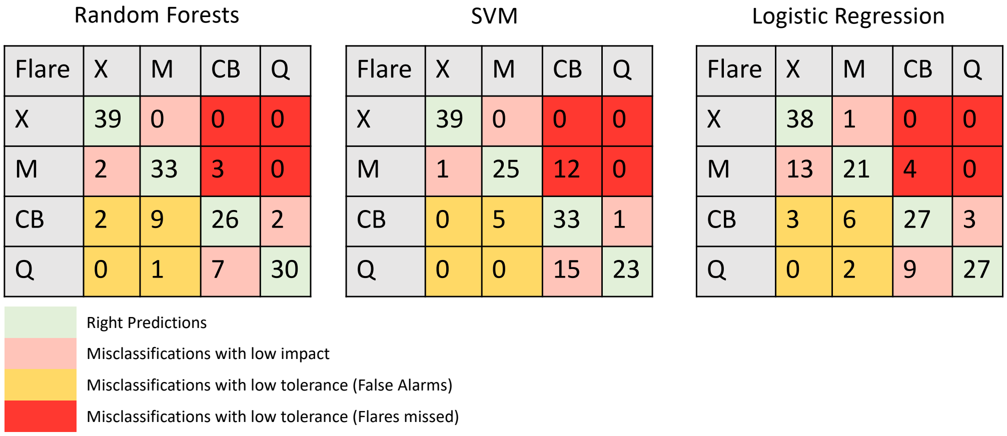


Fig. 4.5: Most discriminative shapelet of each class based on shapelet transforms. Corresponding AR features from which these shapelets are extracted are RBT_VALUE, RBP_VALUE, RBZ_VALUE, TOTPOT for X, M, CB and Q respectively. The graph represents value of the feature on y axis at a given time interval on x axis

4.1.3 Comparison with the Baselines

Support Vector Machines, Logistic Regression, and Random forest (RF) are the three baseline approaches we performed in this thesis. Random forests have stood right after TSF and CNN in terms of performance. Like most classifiers, Random forest has a perfect classification for X class flares and struggled with CB classification. The false alarms in



Random Forests

Flare	X	M	CB	Q
X	39	0	0	0
M	2	33	3	0
CB	2	9	26	2
Q	0	1	7	30

SVM

Flare	X	M	CB	Q
X	39	0	0	0
M	1	25	12	0
CB	0	5	33	1
Q	0	0	15	23

Logistic Regression

Flare	X	M	CB	Q
X	38	1	0	0
M	13	21	4	0
CB	3	6	27	3
Q	0	2	9	27

Right Predictions
Misclassifications with low impact
Misclassifications with low tolerance (False Alarms)
Misclassifications with low tolerance (Flares missed)

Fig. 4.6: Class confusion matrix of the baseline classifiers in median performing fold

yellow region of confusion matrix 4.6 has dragged down the TSS significantly for RF.

Though SVM is not among the top five classifiers, they did a compelling job in classifying CB class flares. SVM's high dimensionality projection would have helped discriminate the minor differences in the CB class. Along with CNN, SVM is the only classifier that performed well in this area. On the downside, SVM is poor at recalling M class flares which did not help its TSS measure. Logistic Regression also struggled to recall M class flares, but it confused with X class, making it safer than SVM, but its CB and Q class performance are only average.

TSF, CNN, LSTM, ANN has surpassed the performance of Logistic Regression and Support Vector Machines. At the same time, TSF crossed Random Forest performance by a comfortable margin, and CNN displayed a slight improvement over Random Forests.

4.1.4 Ensemble Performance

In the above sections, we have analyzed the performance of each classifier and identified the strengths and areas of improvement. Forming an optimal set of classifiers to overcome the challenges faced by TSF is our final attempt at improving the TSS of this flare prediction task.

We know that the major downside of Time Series Forest is producing False alarms; quite a few CB class flares are being classified as M- class flares. If we can improve the recall of CB class flares in TSF, we could arrive at a much robust prediction model. Here, we tried to take advantage of CNN and SVM in this ensemble, exhibiting a good recall of CB class flares. Simultaneously, we added Random Forest, LSTM, and ANN to maintain a low false-negative rate for X and M class flares relatively.

However, we followed a backward recursive approach similar to our feature selection process to identify the best ensemble [43, 44]. We can see in 4.15 that the ensemble with the top 6 classifiers has bumped up the accuracy to 88.05% from 85.5% and TSS from 0.804 to 0.840. If we look into the confusion matrices 4.7, we can observe that ensemble has improved its recall for CB and Q class flares consequently, reducing the false alarms for M-flares.

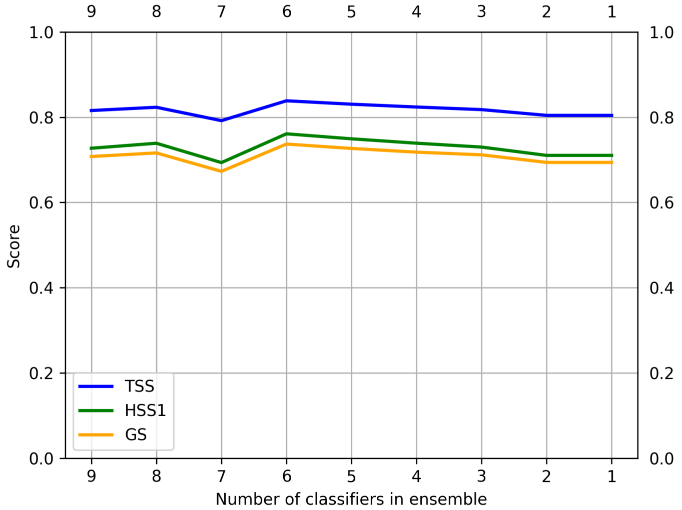
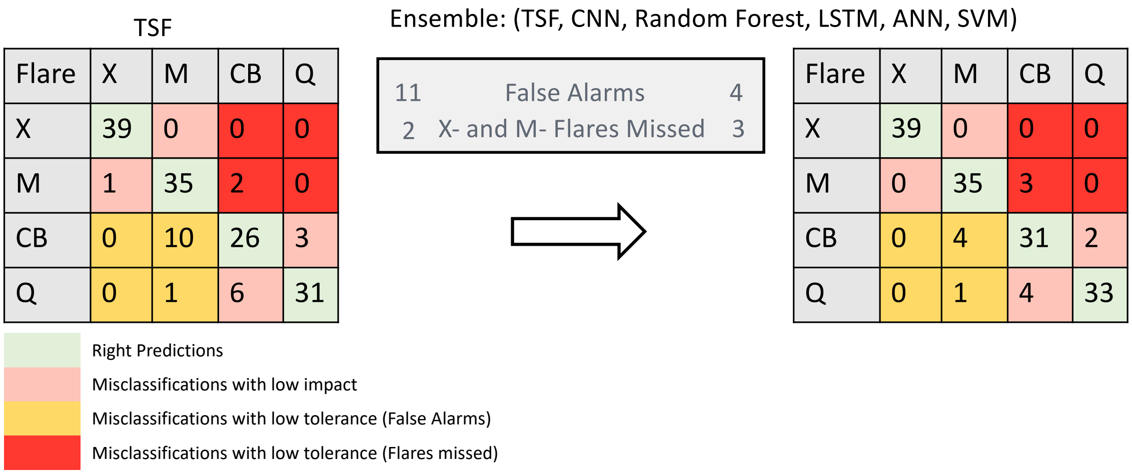


Fig. 4.7: Above line chart shows us the performance of ensemble with number of classifier in x-axis. Starting with all the models in ensemble, we removed one by one based on TSS as priority. We achieved the best TSS with top 6 classifiers in the ensemble.

The comparison made in fig 4.9 shows us the average gain of class-wise performance in



TSF

Flare	X	M	CB	Q
X	39	0	0	0
M	1	35	2	0
CB	0	10	26	3
Q	0	1	6	31

Ensemble: (TSF, CNN, Random Forest, LSTM, ANN, SVM)

11	False Alarms	4
2	X- and M- Flares Missed	3

Flare	X	M	CB	Q
X	39	0	0	0
M	0	35	3	0
CB	0	4	31	2
Q	0	1	4	33

🟩 Right Predictions
▨ Misclassifications with low impact
🟨 Misclassifications with low tolerance (False Alarms)
🟥 Misclassifications with low tolerance (Flares missed)

Fig. 4.8: Confusion matrix representing the reduced False alarm rate in the ensemble model classifier.

ensemble across tenfold evaluation. As the F1 score represents the harmonic mean of recall and precision, we can state that the ensemble's improvement in CB and Q classes has not come at the cost of precision. The below table 4.2 shows our top three models with RUS dataset in this thesis.

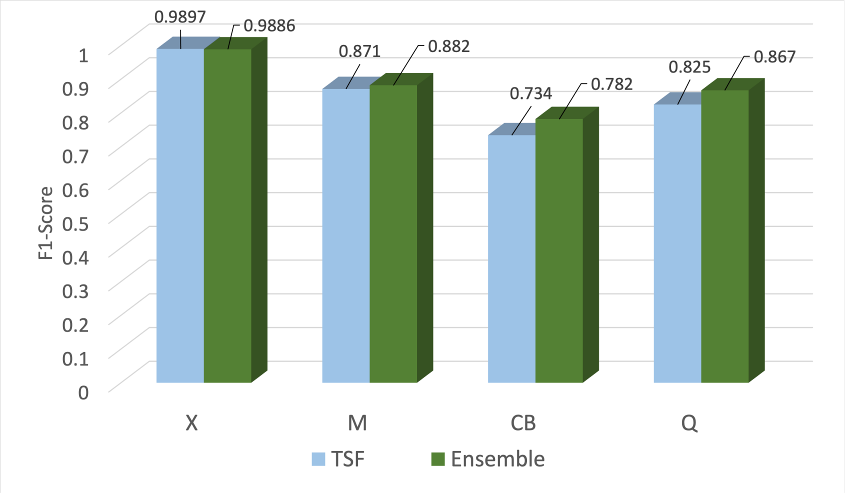


Fig. 4.9: The Bar chart clearly shows that ensemble has successfully helped TSF to improve its performance on CB and Q class flares while X and M class performance is unaffected.

Metric\Model	Ensemble	TSF	CNN
Accuracy	0.881	0.855	0.831
F1-score	0.880	0.854	0.829
Recall	0.881	0.855	0.831
Precision	0.884	0.859	0.836
HSS1	0.838	0.710	0.661
HSS2	0.761	0.803	0.768
GS	0.737	0.693	0.641
TSS	0.839	0.804	0.769

Table 4.2: Performance analysis of top three models achieved in this thesis using RUS dataset.

4.1.5 Comparison with previous work

In this final section of the results, let's compare our results with the latest publications. The comparison is made even though the previous works are modeled as binary classification problems, where X- and M- are merged as flaring classes while C/B and Q are considered as negative classes. Also, not all the works have utilized the same dataset. However, it gives us a good picture of where our models stand.

TSF model in this work with four classes has comfortably surpassed the latest work of Ji [8] that has achieved a TSS of 0.75 in detecting non-flaring activity; We think the reason for betterment is the use of 28 features instead of 2 in Ji's work. In the works of Wang [9], and Hamdi [5] they used LSTM and kNN to achieve an exceptional accuracy of 0.95 and 0.97, respectively; but their TSS indicating significant false positive and false negative rate; especially in the case of LSTM. Our ensemble approach with 0.881 accuracies and TSS of 0.749 looks better than the work of Zheng [35].

Metric\Model	Ensemble*	TSF*	CNN*	LSTM_Wang	kNN_Hamdi	CNN_Zheng	TSF_Ji
Accuracy	0.881	0.855	0.831	0.945	0.975	0.891	-
HSS2	0.839	0.804	0.769	0.382	0.852	0.759	0.57
TSS	0.838	0.804	0.768	0.681	0.885	0.749	0.75

Table 4.3: Comparison of our top models with literature. Models marked with * are from current work

4.1.6 Critical difference diagram

This diagram shows the statistical difference between models tested across different data sets. Our train and test sets from ten folds of stratified cross-validation are used as ten different data sets. Two tests are performed to generate this significance diagram. First, a Friedman test is performed to reject the null hypothesis, and then a post-hoc analysis is based on the Wilcoxon-Holm method. The model with the lowest score is the most statistically significant model, and models falling under the same thick black line can be considered not significantly different. Based on the diagram in 4.10 we can say ensemble of six classifiers (RUS_ensemble) and time series forest are the most statistically significant models.

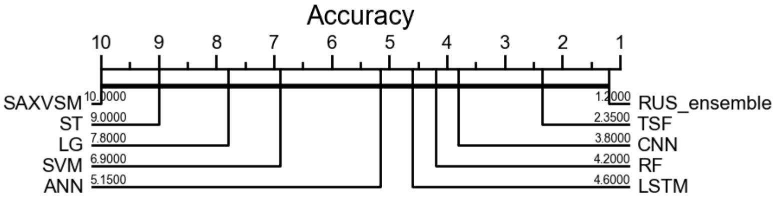


Fig. 4.10: Statistical significance diagram of the solar flare prediction classifiers

4.2 Prediction results on SMOTE_RUS dataset

Experiments on the SMOTE_RUS dataset are trained and evaluated using the top 6 models achieved on the RUS dataset: time series forests, CNN, ANN, LSTM, random forests, and SVM. In this section, we will analyze the results of these models compared to models trained using the RUS dataset. This dataset has resulted in significant improvements in every aspect of the model's performance. The biggest drawback of not having a good recall of the M class has been solved with this augmented dataset. The standings of models are unchanged to the RUS dataset. TSF is still our top performer with a TSS score of 0.92, far better than the RUS_ensemble's TSS of 0.84, while CNN (TSS: 0.88) and random forests (TSS: 0.88) stand in the following two places. The table 4.4 displays the performance metrics.

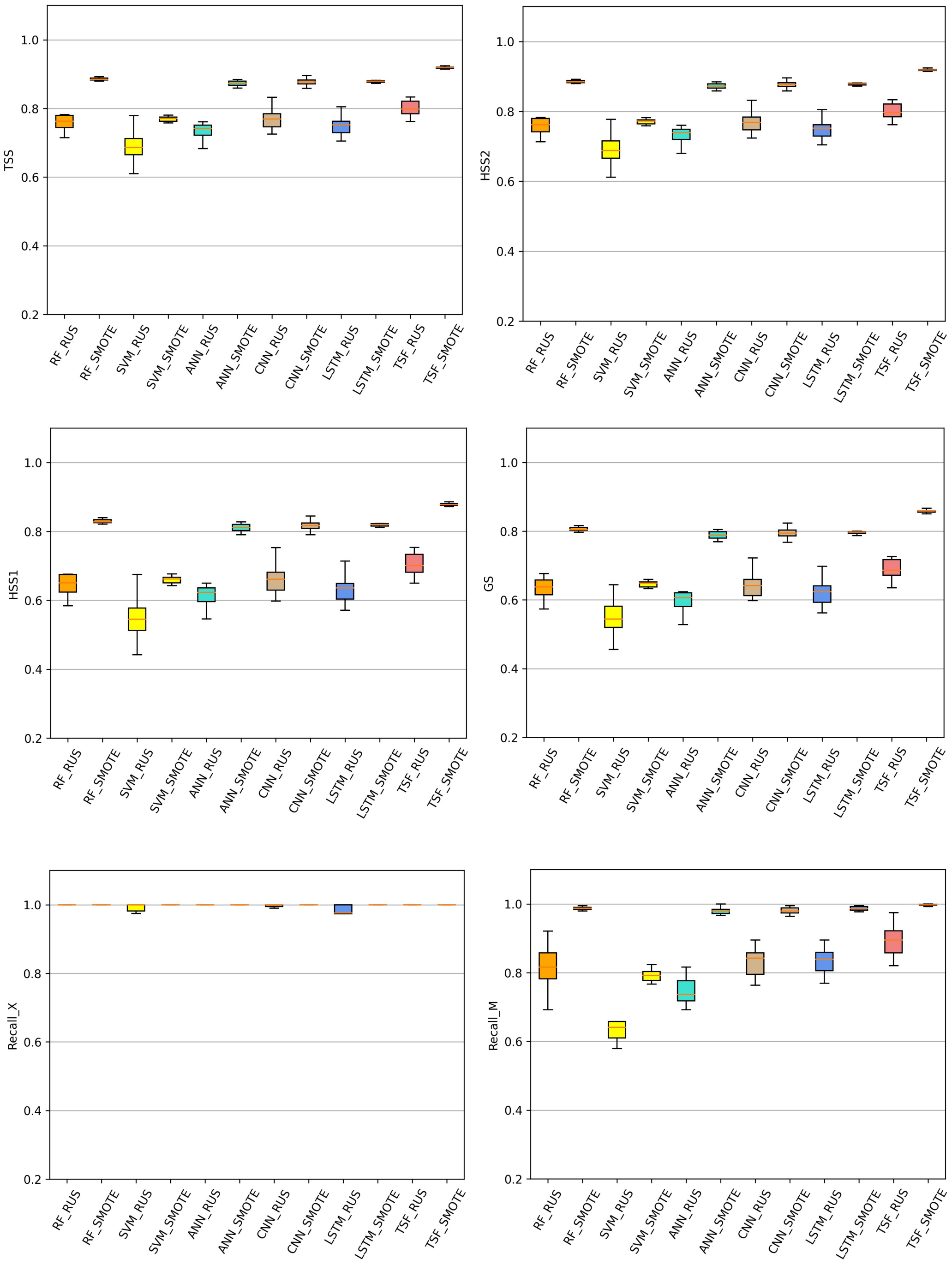


Fig. 4.11: Classifiers across stratified 10-fold cross validation setting on SMOTE_RUS dataset. Smaller box plots represent more consistent performance of classifier and larger ones represent the inconsistency in performance in different folds.

Classifier\Metric	**Accuracy**	**F1-score**	**Recall**	**Precision**	**HSS1**	**HSS2**	**GS**	**TSS**
ANN_SMOTE	0.899 (0.022)	0.899 (0.022)	0.899 (0.022)	0.898 (0.022)	0.799 (0.043)	0.864 (0.03)	0.777 (0.041)	0.865 (0.03)
CNN_SMOTE	0.904 (0.019)	0.903 (0.019)	0.904 (0.019)	0.904 (0.019)	0.807 (0.037)	0.87 (0.026)	0.786 (0.036)	0.87 (0.026)
LSTM_SMOTE	0.91 (0.004)	0.909 (0.004)	0.91 (0.004)	0.91 (0.004)	0.82 (0.009)	0.879 (0.006)	0.797 (0.009)	0.879 (0.006)
TSF_SMOTE	0.939 (0.003)	0.939 (0.003)	0.939 (0.003)	0.94 (0.002)	0.878 (0.006)	0.918 (0.004)	0.858 (0.006)	0.919 (0.004)
RF_SMOTE	0.915 (0.004)	0.915 (0.004)	0.915 (0.004)	0.916 (0.003)	0.83 (0.007)	0.886 (0.005)	0.807 (0.008)	0.886 (0.005)
SVM_SMOTE	0.831 (0.007)	0.833 (0.007)	0.831 (0.007)	0.843 (0.007)	0.662 (0.014)	0.772 (0.01)	0.649 (0.012)	0.771 (0.01)

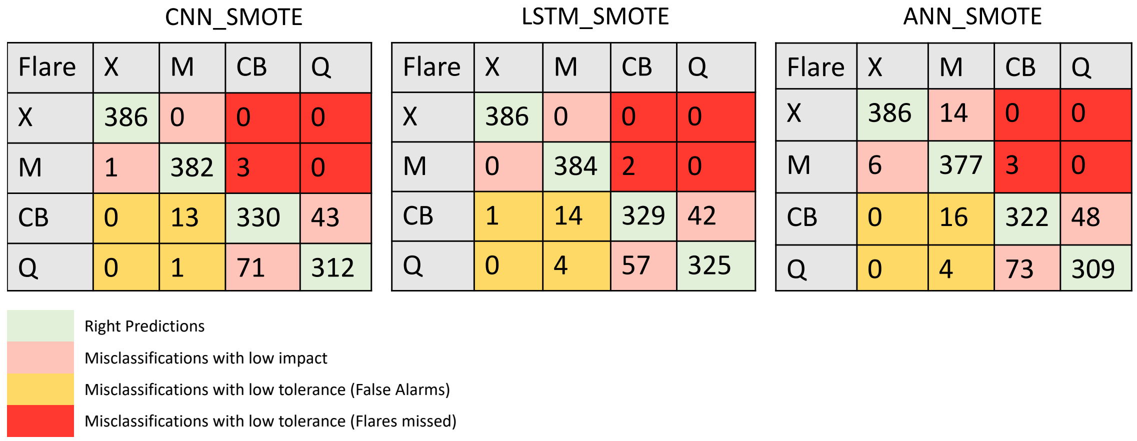
Table 4.4: Classifier Performance Analysis on the SMOTE dataset. Reported values are the mean of the 10-fold stratified cross-validation test accuracies. Value in the parenthesis denote the standard error of each measure across the folds.

Performance in terms of skill scores

Time series forest with SMOTE dataset has achieved the highest TSS of 0.92, indicating that its false-negative rate and false alarm rate have been significantly reduced. Another area of improvement compared to TSF trained using the RUS dataset is that M- flares have achieved a recall of greater than 99.5% along with 100% recall of X- flares are making it a very reliable model. HSS1 and HSS2, which indicate the performance improvement over random predictions and always predicting a negative class, have also spiked along with TSS, showing that the models have not been biased. Random forests, LSTM, CNN, ANN, and SVM, have seen similar behavior in their skill scores as TSF and have also surpassed the skills scores of RUS_ensemble. The box plots in the figure 4.11 show us the big picture of where each model stands in terms of different skill scores and how each of the models benefited from more data SMOTE dataset in comparison to the RUS dataset. Another quality to notice is that the variance in performance across ten-folds is almost negligible, revealing the consistency of models.

4.2.1 Performance of Neural networks

The neural networks trained using SMOTE dataset, LSTM has pulled the top TSS of 0.879, with CNN and ANN very close at TSS of 0.870 and 0.865, respectively. Their consistency across ten folds is exceptional. LSTM has taken an edge over the other two because of its strength in recalling the Q class flares better. However, LSTM has produced low tolerance misclassifications more often than CNN, making it slightly less reliable. Considering the fact that false alarms can pose unnecessary loss, ANN and LSTM stand behind CNN. Moreover, ANN has also struggled to classify between X- and M- class flares. Hence CNN is our winner in the neural networks. The confusion matrices in figure 4.12 make it succinct the areas of strengths and weaknesses of each neural network.



CNN_SMOTE

Flare	X	M	CB	Q
X	386	0	0	0
M	1	382	3	0
CB	0	13	330	43
Q	0	1	71	312

LSTM_SMOTE

Flare	X	M	CB	Q
X	386	0	0	0
M	0	384	2	0
CB	1	14	329	42
Q	0	4	57	325

ANN_SMOTE

Flare	X	M	CB	Q
X	386	14	0	0
M	6	377	3	0
CB	0	16	322	48
Q	0	4	73	309

🟩 Right Predictions
🟥 Misclassifications with low impact
🟨 Misclassifications with low tolerance (False Alarms)
🟥 Misclassifications with low tolerance (Flares missed)

Fig. 4.12: Class confusion matrix of the neural networks in median performing fold

4.2.2 Performance of Time series forests

TSF is the only classifier experimented with SMOTE dataset in the time series classifiers. As we know that TSF is the best performer with the RUS dataset; let's see how the TSF with SMOTE has reached the near operational level model. The most crucial factor to consider in determining the model's feasibility in the solar flare prediction task is its ability to recall all the X- and M- class flares. TSF has managed to recall 99.8% of the X and M class flares, with only one or zero misses across the different folds. The second factor to consider is how less often the model is raising false alarms. TSF has achieved a 99.82% precision for X class flares and 98% precision for M class flares. TSF with SMOTE has well-satisfied both factors and brought us to an almost perfect flare occurrence prediction.

TSF_SMOTE

Flare	X	M	CB	Q
X	386	0	0	0
M	0	386	0	0
CB	0	5	353	28
Q	0	0	61	325

TSF_RUS

Flare	X	M	CB	Q
X	39	0	0	0
M	1	35	2	0
CB	0	10	26	3
Q	0	1	6	31

🟩 Right Predictions
🟧 Misclassifications with low impact
🟨 Misclassifications with low tolerance (False Alarms)
🟥 Misclassifications with low tolerance (Flares missed)

Fig. 4.13: Class confusion matrices comparison of TSF between SMOTE and RUS datasets.

4.2.3 Performance of Random Forest and SVM

These are the two baselines experimented with SMOTE dataset. As the neural networks and TSF, random forest, and SVM have also shown improved performance and consistency across ten folds. Random forests managed to recall flares with near perfection, but they have raised too many false alarms, which affected its TSS (0.88). On the other hand, though SVM has been gained from SMOTE dataset, it is not close to the performance of neural networks or TSF. The number of misses and false alarms is too many to consider as a potential flare prediction model.

Random Forests_SMOTE

Flare	X	M	CB	Q
X	386	0	0	0
M	0	384	1	1
CB	1	17	332	36
Q	0	2	60	324

SVM_SMOTE

Flare	X	M	CB	Q
X	386	14	0	0
M	13	314	59	0
CB	6	52	305	23
Q	0	2	95	289

🟩 Right Predictions
🟧 Misclassifications with low impact
🟨 Misclassifications with low tolerance (False Alarms)
🟥 Misclassifications with low tolerance (Flares missed)

Fig. 4.14: Class confusion matrix of the random forests and SVM in median performing fold

4.2.4 Comparison with previous work

In this section, the top three classifiers, Time-series forests, CNN, and Random forests, have been compared with the latest work published on this solar flare prediction task. Although we compare previous work, which has modeled the study as a binary problem, and our classifiers modeled as 4-class classification, our models have still achieved performance on par with the established state-of-the-art performance. In terms of TSS, our TSF and random forests have surpassed the work of [5] comfortably. The results of Wang [9], and Hamdi [5] have exceptional accuracies, but their relatively low TSS indicates that their models are prone to false alarms and false negatives. Our models have maintained less gap between accuracy, HSS, and TSS, exhibiting the unbiased nature of models. All our neural networks, TSF, and random forests with SMOTE have outperformed the works of Zheng [45] and Ji. Table 4.5 shows us the statistical comparison of our classifiers with previously published research.

Metric\Classifier	**Accuracy**	**HSS2**	**TSS**
TSF_SMOTE*	0.939	0.918	0.919
CNN_SMOTE*	0.905	0.870	0.870
RF_SMOTE*	0.915	0.885	0.886
LSTM_SMOTE*	0.910	0.878	0.879
LSTM_Wang	0.945	0.382	0.681
kNN_Hamdi	0.975	0.852	0.885
CNN_Zheng	0.891	0.759	0.749
TSF_Ji	-	0.57	0.75

Table 4.5: Statistical comparison of our top four classifiers with previous literature. Note that classifiers with * are from this thesis.

4.2.5 Critical difference diagram

According to the diagram, we can say time series forest is the most statistically significant classifier, followed by random forests with a margin of 1.1. CNN and LSTM are under the same bracket of statistical significance and are in third and fourth. Among the six classifiers trained using SMOTE dataset, ANN and SVM are on the left most part of the diagram ending up as the relatively insignificant models.

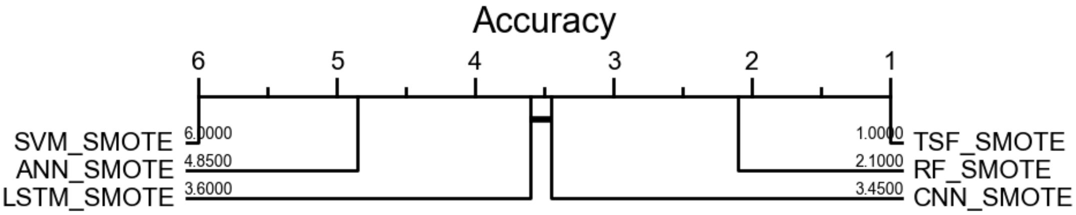


Fig. 4.15: Statistical significance diagram of the solar flare prediction classifiers

CHAPTER 5

Conclusion and Future scope

5.1 Conclusion

The astrophysics community does not have any particular physical theory explaining the mechanism behind the occurrence of solar flares, which constrains the attempts to forecast and classify them. [46] Though several groups of physicists are researching to unfold the definitive theory to forecast the flares, plausibility is low. With AI and machine learning advancing at a quick pace, our best hope is to take a data-driven approach by using the Active region parameters observed by the solar dynamics observatory. The aim is to develop a model which could find an empirical relationship between AR parameters and flare occurrences.

In our thesis, we attempted to predict the occurrence of flare based on 28 AR time-series features with a look-back of 12 hours. We successfully tested nine different Machine learning models in the quest to find the most suitable prediction model. Based on our results, the time series forest has achieved the state-of-the-art 0.92 TSS for the first time in a solar flare prediction problem modeled as a 4 class problem. The TSF was able to recall all the X- and M- flares with 98.87% average precision.

Our first challenge is the huge class imbalance in the dataset with less than 2% flaring samples; it makes sense as the flares are rare. To tackle this problem, we prepared two balanced data sets called RUS and SMOTE_RUS. RUS is prepared by employing random under-sampling on majority classes and the SMOTE_RUS by oversampling the minority class by generating synthetic data with Synthetic minority oversampling technique. We then preprocessed the data, got rid of outliers, imputed the missing values, and standardized the data sets.

Once the data sets were ready, our first objective was to find the optimal set of features that makes our multivariate time series(MVTS) sample. We implemented a feature selection pipeline to rate features based on scores of linear correlation, mutual information, and random forest tests. We identified our optimal MVTS with 28 features using backward recursive feature elimination. Each feature has 60 instances; our MVTS dataset dimension was 28 x 60. We implemented all the machine learning models with the same feature set to be consistent while comparing the models. The major contributions of this thesis are as follows:

1. Established the baseline performance of nine different classifiers using the same datasets, enabling the critical comparison between classifiers. Also, for the first time, a multivariate time series dataset with AR parameters as many as 28 has been utilized to perform solar flare prediction.

2. Tackled the class imbalance problem with two derived datasets from the actual data partition based on random under-sampling and synthetic minority oversampling. The latter has given us well-trained and better-generalized models.

3. Identified that computationally inexpensive time series forest as a potential model that can be improved to an operational level. It has achieved the best performance with an exceptional TSS of 0.92 and decent skill scores. Also, the results of neural networks with the SMOTE_RUS dataset are comparable or even better than the recently published work using LSTM, TSF, kNN, and CNN.

4. Explored shapelet transforms, though not among the top performers for this classification task, they revealed discriminative shapelets which could lay the path for interpretability studies in the future.

The main difference in the results between the classifiers trained on RUS and SMOTE_RUS data sets is the spike in recall of M- class flares. This is solely because of accommodating more samples for M- class in the later dataset. As we have over-sampled only X- class data one can not argue that the synthetic samples have made the classifier biased to M- class. At the same time the recall for X- class flares on time series forest with both the data sets is same, ruling out the possibility of synthetic samples causing bias to X- class. Hence we can justify that the improvement of models with SMOTE_RUS dataset is only because of additional data accommodated.

5.2 Future scope

In our future research, we would like to experiment using weighted cost functions, as this helps us maintain the class imbalance even while training. However, as the imbalance is huge, we can't overlook the sampling process for training the models. Another way to investigate the robustness of models is to train and test using data from different years of solar cycle and see if there is any temporal dependency. For example, we can train the models with data from 2012 - 2014 and try to predict the 2015 data. In light of CNN and LSTM's success, one potential model we could experiment with in the future is CONV-LSTM, which can learn spatio-temporal relationships [47–50]. To improve the skill scores of flare prediction, we plan to perform research in the direction of points mentioned above.

The research in this thesis will serve as a baseline for comparing the results obtained with new experiments and techniques. The entire code base of preprocessing, feature selection, and machine learning implementations are delivered as an easy-to-use component, saving researchers crucial time. We can reuse shapelet algorithm implementations to mine more greedily to extract the shapelets with the highest discriminative power. Mapping shapelets back to the original features could help make the classification process more interpretable.

REFERENCES

[1] R. A. Angryk, P. C. Martens, B. Aydin, D. Kempton, S. S. Mahajan, S. Basodi, A. Ahmadzadeh, X. Cai, S. F. Boubrahimi, S. M. Hamdi *et al.*, "Multivariate time series dataset for space weather data analytics," *Scientific data*, vol. 7, no. 1, pp. 1–13, 2020.

[2] J. Eastwood, E. Biffis, M. Hapgood, L. Green, M. Bisi, R. Bentley, R. Wicks, L.-A. McKinnell, M. Gibbs, and C. Burnett, "The economic impact of space weather: Where do we stand?" *Risk Analysis*, vol. 37, no. 2, pp. 206–218, 2017.

[3] S. F. Boubrahimi, S. M. Hamdi, R. Ma, and R. Angryk, "On the mining of the minimal set of time series data shapelets," in *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 2020, pp. 493–502.

[4] T. Howard, *Coronal mass ejections: An introduction.* Springer Science & Business Media, 2011, vol. 376.

[5] S. M. Hamdi, D. Kempton, R. Ma, S. F. Boubrahimi, and R. A. Angryk, "A time series classification-based approach for solar flare prediction," in *2017 IEEE International Conference on Big Data (Big Data)*. IEEE, 2017, pp. 2543–2551.

[6] P. S. McIntosh, "The classification of sunspot groups," *Solar Physics*, vol. 125, no. 2, pp. 251–267, 1990.

[7] J. Jing, H. Song, V. Abramenko, C. Tan, and H. Wang, "The statistical relationship between the photospheric magnetic parameters and the flare productivity of active regions," *The Astrophysical Journal*, vol. 644, no. 2, p. 1273, 2006.

[8] A. Ji, B. Aydin, M. K. Georgoulis, and R. Angryk, "All-clear flare prediction using interval-based time series classifiers," in *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 2020, pp. 4218–4225.

[9] X. Wang, Y. Chen, G. Toth, W. B. Manchester, T. I. Gombosi, A. O. Hero, Z. Jiao, H. Sun, M. Jin, and Y. Liu, "Predicting solar flares with machine learning: investigating solar cycle dependence," *The Astrophysical Journal*, vol. 895, no. 1, p. 3, 2020.

[10] T. Nagem, R. Qahwaji, S. Ipson, and A. Alasta, "Predicting solar flares by converting goes x-ray data to gramian angular fields (gaf) images," in *Proceedings of the World Congress on Engineering*, vol. 1, 2018.

[11] R. Ma, S. F. Boubrahimi, S. M. Hamdi, and R. A. Angryk, "Solar flare prediction using multivariate time series decision trees," in *2017 IEEE International Conference on Big Data (Big Data)*. IEEE, 2017, pp. 2569–2578.

[12] S. F. Boubrahimi, B. Aydin, D. Kempton, and R. Angryk, "Spatio-temporal interpolation methods for solar events metadata," in *2016 IEEE International Conference on Big Data (Big Data)*. IEEE, 2016, pp. 3149–3157.

[13] S. F. Boubrahimi, B. Aydin, D. Kempton, S. S. Mahajan, and R. Angryk, "Filling the gaps in solar big data: Interpolation of solar filament event instances," in *2016 IEEE International Conferences on Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom)(BDCloud-SocialCom-SustainCom).* IEEE, 2016, pp. 97–104.

[14] R. Ma, A. Ahmadzadeh, S. F. Boubrahimi, and R. A. Angryk, "Segmented dynamic time warping: A comparative and applicational study," in *Emerging Technologies and Applications in Data Processing and Management.* IGI Global, 2019, pp. 1–19.

[15] R. Ma, S. F. Boubrahimi, R. A. Angryk, and Z. Ma, "Evaluation of hierarchical structures for time series data," in *2020 5th IEEE International Conference on Big Data Analytics (ICBDA).* IEEE, 2020, pp. 94–99.

[16] N. Nishizuka, K. Sugiura, Y. Kubo, M. Den, S. Watari, and M. Ishii, "Solar flare prediction model with three machine-learning algorithms using ultraviolet brightening and vector magnetograms," *The Astrophysical Journal*, vol. 835, no. 2, p. 156, 2017.

[17] M. G. Bobra and S. Couvidat, "Solar flare prediction using sdo/hmi vector magnetic field data with a machine-learning algorithm," *The Astrophysical Journal*, vol. 798, no. 2, p. 135, 2015.

[18] A. Al-Ghraibah, L. Boucheron, and R. McAteer, "An automated classification approach to ranking photospheric proxies of magnetic energy build-up," *Astronomy & Astrophysics*, vol. 579, p. A64, 2015.

[19] O. W. Ahmed, R. Qahwaji, T. Colak, P. A. Higgins, P. T. Gallagher, and D. S. Bloomfield, "Solar flare prediction using advanced feature extraction, machine learning, and feature selection," *Solar Physics*, vol. 283, no. 1, pp. 157–175, 2013.

[20] Y. Yuan, F. Y. Shih, J. Jing, and H.-M. Wang, "Automated flare forecasting using a statistical learning technique," *Research in Astronomy and Astrophysics*, vol. 10, no. 8, p. 785, 2010.

[21] D. Yu, X. Huang, H. Wang, and Y. Cui, "Short-term solar flare prediction using a sequential supervised learning method," *Solar Physics*, vol. 255, no. 1, pp. 91–105, 2009.

[22] R. A. F. Borda, P. D. Mininni, C. H. Mandrini, D. O. Gómez, O. H. Bauer, and M. G. Rovira, "Automatic solar flare detection using neural network techniques," *Solar Physics*, vol. 206, no. 2, pp. 347–357, 2002.

[23] J. Oloketuyi, Y. Liu, and M. Zhao, "The periodic and temporal behaviors of solar x-ray flares in solar cycles 23 and 24," *The Astrophysical Journal*, vol. 874, no. 1, p. 20, 2019.

[24] G. Aulanier, M. Janvier, and B. Schmieder, "The standard flare model in three dimensions-i. strong-to-weak shear transition in post-flare loops," *Astronomy & Astrophysics*, vol. 543, p. A110, 2012.

[25] M. G. Bobra, X. Sun, J. T. Hoeksema, M. Turmon, Y. Liu, K. Hayashi, G. Barnes, and K. Leka, "The helioseismic and magnetic imager (hmi) vector magnetic field pipeline: Sharps–space-weather hmi active region patches," *Solar Physics*, vol. 289, no. 9, pp. 3549–3578, 2014.

[26] J. L. Leevy, T. M. Khoshgoftaar, R. A. Bauder, and N. Seliya, "A survey on addressing high-class imbalance in big data," *Journal of Big Data*, vol. 5, no. 1, pp. 1–30, 2018.

[27] R. Dubey, J. Zhou, Y. Wang, P. M. Thompson, J. Ye, A. D. N. Initiative *et al.*, "Analysis of sampling techniques for imbalanced data: An n= 648 adni study," *NeuroImage*, vol. 87, pp. 220–241, 2014.

[28] S. Mishra, "Handling imbalanced data: Smote vs. random undersampling," *International Research Journal of Engineering and Technology (IRJET)*, vol. 4, no. 8, 2017.

[29] S. F. Boubrahimi, R. Ma, B. Aydin, S. M. Hamdi, and R. Angryk, "Scalable knn search approximation for time series data," in *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, 2018, pp. 970–975.

[30] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[31] C. C. Balch, "Updated verification of the space weather prediction center's solar energetic particle prediction model," *Space Weather*, vol. 6, no. 1, 2008.

[32] G. Barnes and K. Leka, "Evaluating the performance of solar flare forecasting methods," *The Astrophysical Journal Letters*, vol. 688, no. 2, p. L107, 2008.

[33] J. P. Mason and J. Hoeksema, "Testing automated solar flare forecasting with 13 years of michelson doppler imager magnetograms," *The Astrophysical Journal*, vol. 723, no. 1, p. 634, 2010.

[34] A. Manzato, "An odds ratio parameterization for roc diagram and skill score indices," *Weather and Forecasting*, vol. 20, no. 6, pp. 918–930, 2005.

[35] Y. Zhang, "Support vector machine classification algorithm and its application," in *International Conference on Information Computing and Applications*. Springer, 2012, pp. 179–186.

[36] S. M. Hamdi, Y. Wu, S. F. Boubrahimi, R. Angryk, L. C. Krishnamurthy, and R. Morris, "Tensor decomposition for neurodevelopmental disorder prediction," in *International Conference on Brain Informatics*. Springer, 2018, pp. 339–348.

[37] S. M. Hamdi, S. Filali Boubrahimi, and R. Angryk, "Tensor decomposition-based node embedding," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 2105–2108.

[38] S. M. Hamdi, B. Aydin, S. F. Boubrahimi, R. Angryk, L. C. Krishnamurthy, and R. Morris, "Biomarker detection from fmri-based complete functional connectivity networks," in *2018 IEEE First International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*. IEEE, 2018, pp. 17–24.

[39] A. Cutler, D. R. Cutler, and J. R. Stevens, “Random forests,” in *Ensemble machine learning.* Springer, 2012, pp. 157–175.

[40] A. Bagnall, A. Bostrom, J. Large, and J. Lines, “The great time series classification bake off: An experimental evaluation of recently proposed algorithms,” *Extended Version. CoRR, abs*, vol. 1602.

[41] H. Deng, G. Runger, E. Tuv, and M. Vladimir, “A time series forest for classification and feature extraction,” *Information Sciences*, vol. 239, pp. 142–153, 2013.

[42] S. F. Boubrahimi, B. Aydin, P. Martens, and R. Angryk, “On the prediction of >100 mev solar energetic particle events using goes satellite data,” in *2017 IEEE International Conference on Big Data (Big Data).* IEEE, 2017, pp. 2533–2542.

[43] S. F. Boubrahimi, R. Ma, and R. Angryk, “Neuro-ensemble for time series data classification,” in *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA).* IEEE, 2018, pp. 50–59.

[44] S. F. Boubrahimi, R. Ma, B. Aydin, and R. Angryk, “Neuro-ensemble,” in *2018 IEEE First International Conference on Artificial Intelligence and Knowledge Engineering (AIKE).* IEEE, 2018, pp. 54–61.

[45] Y. Zheng, X. Li, and X. Wang, “Solar flare prediction with the hybrid deep convolutional neural network,” *The Astrophysical Journal*, vol. 885, no. 1, p. 73, 2019.

[46] K. Kusano, T. Iju, Y. Bamba, and S. Inoue, “A physics-based method that can predict imminent large solar flares,” *Science*, vol. 369, no. 6503, pp. 587–591, 2020.

[47] B. Aydin, A. Kucuk, S. F. Boubrahimi, and R. A. Angryk, “Top-(r%, k) spatiotemporal event sequence mining,” in *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, 2017, pp. 250–257.

[48] A. Kucuk, B. Aydin, S. F. Boubrahimi, D. Kempton, and R. A. Angryk, “An integrated solar database (isd) with extended spatiotemporal querying capabilities,” in *International Symposium on Spatial and Temporal Databases.* Springer, 2017, pp. 405–410.

[49] B. Aydin, R. Angryk, S. Filali Boubrahimi, and S. M. Hamdi, “Spatiotemporal frequent pattern discovery from solar event metadata,” in *AGU Fall meeting abstracts*, 2016, pp. SH34A–08.

[50] B. Aydin, S. F. Boubrahimi, A. Kucuk, B. Nezamdoust, and R. A. Angryk, “Spatiotemporal event sequence discovery without thresholds,” *Geoinformatica*, vol. 25, no. 1, pp. 149–177, 2021.