UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Tecnologia

Luís Fernando Lopes Grim

Solar Flare Forecasting with Deep Learning based on Magnetogram Sequences and Data Augmentation Techniques

Previsão de Explosões Solares com Aprendizado Profundo baseado em Sequências de Magnetogramas e Técnicas de Aumento de Dados

Limeira
2024

Luís Fernando Lopes Grim

Solar Flare Forecasting with Deep Learning based on Magnetogram Sequences and Data Augmentation Techniques

Previsão de Explosões Solares com Aprendizado Profundo baseado em Sequências de Magnetogramas e Técnicas de Aumento de Dados

Tese apresentada à Faculdade de Tecnologia da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Doutor em Tecnologia, na área de Sistemas de Informação e Comunicação.

Thesis presented to the School of Technology of the University of Campinas in partial fulfillment of the requirements for the degree of Doctor in Technology in Computer Science, in the area of Sistemas de Informação e Comunicação.

Supervisor/Orientador: Prof. Dr. André Leon Sampaio Gradvohl

Este trabalho corresponde à versão final da Tese defendida por Luís Fernando Lopes Grim e orientada pelo Prof. Dr. André Leon Sampaio Gradvohl.

Limeira
2024

Ficha catalográfica
Universidade Estadual de Campinas (UNICAMP)
Biblioteca da Faculdade de Tecnologia
Felipe de Souza Bueno - CRB 8/8577

G88s

Grim, Luís Fernando Lopes, 1987-
Solar flare forecasting with deep learning based on magnetogram sequences and data augmentation techniques / Luís Fernando Lopes Grim. – Limeira, SP : [s.n.], 2024.

Orientador: André Leon Sampaio Gradvohl.
Tese (doutorado) – Universidade Estadual de Campinas (UNICAMP), Faculdade de Tecnologia.

1. Aprendizado profundo. 2. Erupções solares. 3. Transformer (Arquitetura de computador). 4. Processamento de imagens. 5. Astrofísica. I. Gradvohl, André Leon Sampaio, 1973-. II. Universidade Estadual de Campinas (UNICAMP). Faculdade de Tecnologia. III. Título.

Informações Complementares

Título em outro idioma: Previsão de explosões solares com aprendizado profundo baseado em sequências de magnetogramas e técnicas de aumento de dados
Palavras-chave em inglês:
Deep learning
Solar flares
Transformer (Computer architecture)
Image processing
Astrophysics
Área de concentração: Sistemas de Informação e Comunicação
Titulação: Doutor em Tecnologia
Banca examinadora:
André Leon Sampaio Gradvohl [Orientador]
Nina Sumiko Tomita Hirata
Alessandra Abe Pacini
Rafael Duarte Coelho dos Santos
Marcos Gomes-Borges
Data de defesa: 18-09-2024
Programa de Pós-Graduação: Tecnologia

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0000-0002-1221-4095
- Currículo Lattes do autor: http://lattes.cnpq.br/2777577510233420

FOLHA DE APROVAÇÃO

Abaixo se apresentam os membros da comissão julgadora da sessão pública de defesa de dissertação para o Título de Doutor em Tecnologia na área de concentração Sistemas de Informação e Comunicação, a que se submeteu o aluno Luís Fernando Lopes Grim, em 18 de setembro de 2024 na Faculdade de Tecnologia – FT/UNICAMP, em Limeira/SP.

Prof. Dr. André Leon Sampaio Gradvohl
Presidente da Comissão Julgadora

Profa. Dra. Nina Sumiko Tomita Hirata
IME/USP

Dra. Alessandra Abe Pacini
NOAA

Dr. Rafael Duarte Coelho dos Santos
INPE

Dr. Marcos Gomes-Borges
ENGIE

Ata da defesa, assinada pelos membros da Comissão Examinadora, encontra-se no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria de Pós-graduação da Faculdade de Tecnologia.

Acknowledgements

First, I would like to thank my advisor, Prof. Dr. André Leon Sampaio Gradvohl, for his support and trust, despite all the obstacles and unforeseen events that arose and were exacerbated by the COVID pandemic in 2020-2021.

I want to thank my family, my wife Marciene, and my children, Fernando and Natália, for their support and unconditional love. I would also like to thank my parents, Pasquina and José Roberto, for supporting and helping me to build a more appropriate environment for concluding my doctorate.

I thank the SDO/HMI and NOAA/SWPC for the original data, the Space Environment Warning and AI Technology Interdisciplinary Innovation Working Group for providing the SOLAR-STORM2 dataset, and the Centro Nacional de Processamento de Alto Desempenho em São Paulo (CENAPAD-SP) for delivering a high-performance computing environment to train the models developed in this research.

I would also like to thank the Federal Institute of São Paulo (IFSP), for the leave granted in November 2020 and for the incentives to pursue graduate studies, especially to the staff at the Piracicaba campus involved.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001

Resumo

As explosões solares são liberações intensas de energia eletromagnética que ocorrem normalmente em regiões solares ativas do Sol com fortes campos magnéticos, conhecidas como manchas solares. A radiação de uma explosão solar pode atingir a atmosfera da Terra em poucos minutos. As explosões solares de alta intensidade, como as de classes M e X, podem impactar significativamente as tecnologias e atividades da Terra, incluindo satélites, telecomunicações e sistemas de energia. Portanto, é fundamental desenvolver sistemas de previsão com boas taxas de acerto para explosões solares de alta intensidade. Um modelo de previsão que monitora a evolução de regiões solares ativas pode analisar vários atributos para prever quais regiões podem tornar-se precursoras de explosões solares. As pesquisas recentes têm se concentrado cada vez mais em modelos de aprendizagem profunda que acompanham a evolução destas regiões ativas. No entanto, as explosões de classe M e X são relativamente raras no ciclo solar, resultando em conjuntos de dados desequilibrados que dificultam o desenvolvimento de modelos de previsão eficazes. Para enfrentar esse desafio, propusemos a adoção de modelos baseados em *transformers*, treinados com o método de ajuste fino para previsão de explosões solares de classes ≥M, usando sequências de imagens de magnetogramas visíveis como entrada. Também aplicamos técnicas de aumento de dados, aumentando artificialmente as amostras positivas de explosões solares de classe M e X, para lidar com os desequilíbrios nos conjuntos de treinamento e validação. Nossos modelos ajustados superaram o estado da arte, alcançando um escore de *True Skill Statistic* (TSS) de aproximadamente 0,8 para explosões ≥classe M, no horizonte de previsão de 48 h. Também alcançamos resultados compatíveis com estado da arte (TSS $\approx 0,7$) no horizonte de previsão de 24 h. Além disso, as técnicas de aumento de dados empregadas apenas no conjunto de treinamento mantiveram um TSS estável e melhoraram a maioria das métricas secundárias analisadas. Essas melhorias atingiram um aumento de +0.07 e +0.03 pontos no Heidke Skill Score dos respectivos horizontes de previsão de 48 h e 24 h.

Abstract

Solar flares are intense releases of electromagnetic energy typically occurring in active solar regions with strong magnetic fields, known as sunspots. The radiation from a solar flare can reach Earth's atmosphere within minutes. High-intensity solar flares, such as M- and X-class, can significantly impact Earth's technologies and activities, including satellites, telecommunications, and power systems. Therefore, it is essential to develop forecasting systems with good accuracy rates for high-intensity solar flares. A forecasting model that monitors the evolution of active solar regions can analyze various attributes to predict which regions might become precursors to solar flares. Recent research has increasingly focused on deep-learning models that track the evolution of these active regions. However, M- and X-class flares are relatively rare within the solar cycle, resulting in imbalanced datasets that complicate the development of effective forecasting models. To address this challenge, we proposed adopting Transformer-based models trained with the fine-tuning method for forecasting flares of ≥M-class, using sequences of line-of-sight magnetogram images as input. We also applied data augmentation techniques, artificially increasing positive samples of M- and X-class flares, to handle the imbalances in the training and validation sets. Our fine-tuned models outperformed state-of-the-art methods, achieving a True Skill Statistic (TSS) score of approximately 0.8 for ≥M-class flares within the 48 h forecasting horizon. We also achieved results compatible with the state-of-the-art (TSS $\approx$ 0.7) in the 24 h forecasting horizon. Additionally, the data augmentation techniques employed only in the training set maintained a stable TSS and improved most of the secondary metrics analyzed. These improvements achieved a +0.07 and +0.03 point increase in the Heidke Skill Score of the respective 48 h and 24 h forecasting horizons.

List of Figures

2.1 Sun layers 23
2.2 Grainy appearance of the photosphere. 24
2.3 Sunspots 24
2.4 Solar cycles progression 25
2.5 The Butterfly Diagram 26
2.6 Formation of magnetic fields and magnetic arcs 26
2.7 Magnetogram and magnetic arcs 27
2.8 X-ray fluxes captured by GOES−15 satellites. 30
2.9 X9.3 flare on September 6, 2017 30
2.10 Sun – Earth interactions 31
2.11 Impacts of solar flares on Earth 32
2.12 The Perceptron 34
2.13 Forward pass 35
2.14 Backward pass 36
2.15 Supervised learning scheme 37
2.16 Typical architecture of a CNN. 38
2.17 Common tasks in image processing. 39
2.18 Video classification/recognition task. 40
2.19 Transformer models for computer vision 42

4.1 Data we used from SOLAR-STORM2 dataset 62
4.2 Artificial Seq_{TS} creation from incomplete Seq_S 65
4.3 Organization of text files to reference the sequences. 67
4.4 Dataset splitting. 68
4.5 MViTv2_s Multiscale Block Modules 71
4.6 MViTv2_s initial and final layers. 73
4.7 Oversampling ≥M-class flares sequences 77
4.8 Methodology overview. 79

5.1 SF_MViT convergence in the M24 dataset. 81
5.2 SF_MViT_oT convergence in the M24 dataset. 82
5.3 SF_MViT_oTV convergence in the M24 dataset. 82
5.4 SF_MViT convergence in the M48 dataset. 84
5.5 SF_MViT_oT convergence in the M48 dataset. 85
5.6 SF_MViT_oTV convergence in the M48 dataset. 85
5.7 Models' performance means on the M24 dataset. 89
5.8 Models' performance means on the M48 dataset. 91
5.9 Forced timestamp reorder in the original dataset. 94

List of Tables

2.1 Solar flares' classes 29

3.1 Precursor works 47
3.2 Binary ≥C-class flares forecasting tasks with deep learning 57
3.3 Binary ≥M-class flares forecasting tasks with deep learning 58

4.1 Summary of different periods in the SOLAR-STORM2 dataset. 63
4.2 RoPS of each fold and test data for M24 and M48 datasets. 68
4.3 Torchvision's Video MViTv2_s loaded parameters. 72
4.4 Hyperparameters' initial configuration. 75
4.5 RoPS of each fold for M24, M48, M24_Over, and M48_Over datasets. 78

5.1 Convergence status of the best epoch on the M24 dataset. 83
5.2 Estimated times to train the models in the M24 dataset. 84
5.3 Convergence status of the best epoch in the M48 dataset. 86
5.4 Estimated times to train the models in the M48 dataset. 86
5.5 SF_MViT test results for ≥M-class flares within the 24 h forecasting horizon. . 87
5.6 SF_MViT_oT test results for ≥M-class flares within the 24 h forecasting horizon. 88
5.7 SF_MViT_oTV test results for ≥M-class flares within the 24 h forecasting horizon. 88
5.8 SF_MViT test results for ≥M-class flares within the 48 h forecasting horizon. . 89
5.9 SF_MViT_oT test results for ≥M-class flares within the 48 h forecasting horizon. 90
5.10 SF_MViT_oTV test results for ≥M-class flares within the 48 h forecasting horizon. 90
5.11 Inference testing times 92
5.12 Performance comparison with Tang, Liao, et al. (2021) results in the same context. 93
5.13 Comparison for predicting ≥M-class flares within the 48 h forecasting horizon. 95
5.14 Comparison for predicting ≥M-class flares within the 24 h forecasting horizon. 96
5.15 Comparison for ≥M-class flares with biased test sets. 98

A.1 Outputs of a Confusion Matrix 114

List of Symbols

Å	Ångström, it is a metric unit of length equal to 10^{-10} m.
$\mathrm{W/m^2}$	Watts per square meter.
I	Peak intensity.
x	input of an Artificial Neural Network (ANN).
w	weight of an ANN.
$f(.)$	activation function of an ANN.
$\hat{y}$	output of an ANN.
y	desired output of an ANN.
$(z)^+$	ReLU function of z output.
$\tanh(z)$	Hyperbolic Tangent function of z output.
$\sigma(z)$	Logistic Sigmoid Function of z output.
J	Cost Function.
η	Learning rate.
∂	Partial derivative.
E	Error between $\hat{y}$ and the respective target value y.
∇	Gradient Descent.
$\mathcal{P}$	Pooling operations/layers.
R	Relative position embedding.
≥C-class	Includes C-, M-, and X-class flares.
≥M-class	Includes M- and X-class flares.
Seq_T	Complete sequence.
Seq_S	Incomplete sequence.
$\mathrm{Seq}_{\mathrm{TS}}$	Artificial sequence.
Δt	Window size of a complete sequence.
Δs	Window size of an incomplete sequence.
$\Phi(z)$	Cumulative Distribution Function for the Gaussian Distribution.
ω_c	Weight associated with each class.

List of Acronyms

3D Conv.	3D Convolutional layer
ACC	Accuracy
AIA	Atmospheric Imaging Assembly
ANN	Artificial Neural Networks
AR	Active Region
ASAP	Automated Solar Activity Prediction
bi-LSTM	bidirectional Long Short Term Memory
BN	Batch Normalization
BSS	Brier Skill Score
CENAPAD-SP	Centro Nacional de Processamento de Alto Desempenho em São Paulo
CME	Coronal Mass Ejection
CNN	Convolutional Neural Networks
CSI	Critical Success Index
DeFN	Deep Flare Net
DNN	Deep Neural Network
ERT	Extremely Randomized Trees
FAR	False Alarm Ratio
FH	Forecasting Horizon
FITS	Flexible Image Transport System
FN	False Negatives
FP	False Positives
FT	School of Technology
GAN	Generative Adversarial Network
GELU	Gaussian Error Linear Units

GMGS	Gandin–Murphy–Gerrity Score
GOES	Geostationary Operational Environmental Satellite
GPS	Global Positioning System
GPU	Graphical Processing Units
HMI	Helioseismic and Magnetic Imager
HSS	Heidke Skill Score
INPE	Instituto Nacional de Pesquisas Espaciais
JSOC	Joint Science Operations Center
k-NN	k-Nearest Neighbors
LR	Learning Rate
LRCN	Long-term Recurrent Convolutional Network
LSTM	Long Short Term Memory
MDI	Michelson Doppler Imager
MF	Magnetic Features
MM	Magnetogram Module
MSE	Mean Squared Error
MViT	Multiscale Vision Transformers
MViTv2	Improved Multiscale Vision Transformers
NCEI	National Centers for Environmental Information
NGDC	National Geophysical Data Center
NOAA	National Oceanic and Atmospheric Administration
NOAA-SWPC	National Oceanic and Atmospheric Administration Space Weather Prediction Center
POD / Recall	Probability of Detection, also known as Recall
PRE	Precision
R-CNN	Region-Based Convolutional Neural Network
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Networks
RoPS	Rate of Positive Samples

SD	Standard Deviation
SDO	Solar Dynamics Observatory
SF_MViT	Solar Flare MViT
SF_MViT_oT	Solar Flare MViT over Train
SF_MViT_oTV	Solar Flare MViT over Train and Validation
SFM	Sunspot Feature Module
SHARP	Spaceweather HMI Active Region Patch
SMART	Solar Monitor Active Region Tracker
SNARK	Stochastic Neural Analog Reinforcement Calculator
SOHO	Solar and Heliospheric Observatory
SVM	Support Vector Machines
SWPC	Space Weather Prediction Center
TN	True Negatives
TP	True Positives
TSS	True Skill Statistic
UNICAMP	University of Campinas
UTC	Coordinated Universal Time
Val_ACC	Validation Accuracy
Val_Loss	Validation Loss
ViT	Vision Transformer

Contents

1 Introduction **16**
1.1 Hypothesis 19
1.2 Objectives 20
1.2.1 Specific Objectives 20
1.3 Thesis Structure 20

2 Theoretical Framework **22**
2.1 Space Weather 22
2.1.1 The Sun’s Layers 22
2.1.2 Sunspots 24
2.1.3 Magnetograms 27
2.1.4 Active Regions and SHARP Patches 28
2.1.5 Solar Flares 28
2.1.6 Solar Flares’ Impacts on Earth 31
2.2 Deep Learning 33
2.2.1 Artificial Neural Networks 33
2.2.2 Backpropagation Algorithm 34
2.2.3 Supervised Learning 36
2.2.4 Convolutional Neural Network 37
2.2.5 Image Processing with Deep Learning 39
2.2.6 Video Classification/Recognition Tasks 40
2.3 Transformer Models for Image Processing 41
2.3.1 Multiscale Vision Transformers 43
2.4 Concluding Remarks 45

3 Literature Review **46**
3.1 Precursor Works 46
3.1.1 Works Adopting Machine Learning 48
3.2 Solar Flare Forecasting with Deep Learning 50
3.2.1 Pioneer Works Considering Magnetogram Images as Inputs 51
3.2.2 Works Adopting Resampling Techniques 53
3.2.3 Works Considering Magnetograms Images and Sequences 54
3.3 Concluding Remarks 59

4 Methodology **61**
4.1 The SOLAR-STORM2 dataset 61
4.2 Building SHARP Magnetogram Image Sequences 64
4.2.1 Text Files for Storing SHARP Sequences 66
4.3 Dataset Splitting 67

4.4 Execution environment 69
4.4.1 Software Resources 69
4.4.2 Hardware Resources 69
4.5 Implementation and Configuration of SF_MViTs 70
4.5.1 Torchvision’s Video MViTv2_s Loaded Structure 71
4.5.2 SF_MViTs Adaptations and Configurations 73
4.6 Oversampling SHARP Magnetogram Sequences 76
4.7 Methodology Overview 78
4.8 Concluding Remarks 79

5 Results and Discussions 81
5.1 M24 Dataset Training Convergence 81
5.1.1 M24 Dataset Training Times 83
5.2 M48 Dataset Training Convergence 84
5.2.1 M48 Dataset Training Times 86
5.3 Results Obtained in the Test Step 87
5.3.1 Inference Testing Times 92
5.4 Literature Comparison 93
5.4.1 Indirectly Compared Works 95
5.4.2 Biased Results Comparison 97
5.5 Concluding Remarks 99

6 Conclusions 101
6.1 Related Publications 103
6.2 Future Works 103

References 105

A Metrics to evaluate the model’s performances 114

B Operational Challenges Faced 117

Chapter 1

Introduction

Solar flares are powerful and short-lived bursts of energy that occur due to abrupt shifts in the Sun's surface magnetic fields. This energy (electromagnetic waves) can travel as fast as the speed of light and can affect Earth's outer atmosphere and nearby space within minutes. The effects upon the sunlit side of Earth's atmosphere occur when we observe the event. Solar flares are connected with some Coronal Mass Ejection (CME) and influence particle acceleration. Therefore, they are considered important events in space weather (ECHER et al., 2005).

People can perceive the effects of solar flares, such as the shutdown or failures of aerospace systems, on the Earth's atmosphere and near space. An example occurred on February 3rd, 2022 when SpaceX Starlink launched and subsequently lost 38 of 49 satellites one day after launch[1] due to a solar storm (FANG, T.-W. et al., 2022). Solar flares can also affect communication systems, GPS, energy generation, and oil pipelines (ECHER et al., 2005).

The strength of solar flares determines the impact and damage in these systems. Scientists categorize solar flares based on the flux of X-ray emissions (NOAA, 2024) through classes A, B, C, M, and X. Each successive class has a peak X-ray flux ten times greater than the previous class. Apart from the X-class, each class progresses linearly from 1 to 9, indicating the flare's intensity (CINTO; GRADVOHL, et al., 2020a). The X-class does not have a maximum limit.

According to the National Oceanic and Atmospheric Administration Space Weather Prediction Center (NOAA-SWPC), X-class flares can cause radio blackouts (ECHER et al., 2005). Their effects span from one (X1-flare) to several hours (X20-flare), causing a loss in

[1] Available at `https://www.bbc.com/news/world-60317806`.

positioning and increasing satellite navigation errors. Its related radiation storms can even cause total satellite loss and high radiation hazards to astronauts and passengers of high-flying aircraft at high latitudes. M-class flares are medium and can cause brief radio blackouts (until 10 minutes) (NOAA-SWPC, 2024). C-, B-, and A-class flares generally cause minor problems on Earth (ECHER et al., 2005). Hence, it is crucial to focus on developing forecasting systems for X- and M-class solar flares, as predicting the occurrence of these flares helps mitigate their effects.

Solar flares often stem from disruptions in an Active Region (AR), which arise when magnetic fields of opposite polarities converge, generating magnetic arcs. These arcs' footprints can be identified as sunspots, and invisible light wavelengths with their magnetic complexity can serve as potential indicators of forthcoming solar flares. Observation of sunspots' magnetic fields is possible through imagery that detects the intensity and direction of magnetic fields on the Sun's surface. Such images, called magnetograms, show white zones denoting north polarity (expanding away from the Sun), black zones representing south magnetic polarity (directing the Sun's core), and gray zones indicating neutral polarity.

Since the 2010s, several authors have adopted numerical parameters obtained from magnetogram images as features for solar flare forecasting through conventional machine-learning models, such as Random Forest, Support Vector Machines (SVM), Artificial Neural Networks (ANN), and similar techniques (COLAK; QAHWAJI, 2009; BOBRA; COUVIDAT, 2015; NISHIZUKA; SUGIURA; KUBO; DEN; WATARI, et al., 2017).

Most of them trained their models with data provided by Spaceweather HMI Active Region Patch (SHARP), which is a series containing several space weather parameters calculated from the photospheric vector magnetogram data and stored as Flexible Image Transport System (FITS) header keywords and 31 data segments, including images and numerical parameters. These data segments did not comprise the solar full disk. Instead, they are partial-disk automatically identified AR patches (BOBRA; SUN, X., et al., 2014).

In the past half-decade, researchers have increasingly turned to deep-learning methodologies for solar flare forecasting, adopting magnetogram SHARP‘s numerical parameters and directly employing line-of-sight magnetogram images as inputs. Deep-learning methods, a subset of machine-learning techniques rooted in ANN, have seen notable adoption, with Convolutional Neural Networks (CNN) emerging as the primary choice for image processing. Conversely, Recurrent Neural Networks (RNN), particularly

Long Short Term Memory (LSTM) networks, are typically utilized in sequential data analysis domains, such as handling time-series data (LECUN; BENGIO; HINTON, G., 2015).

Taking numerical magnetogram parameters from cropped ARs as input, Nishizuka, Sugiura, Kubo, Den, and Ishii (2018) constructed a predictive model training a Deep Neural Network (DNN) for solar flare forecasting. Conversely, Xin Huang, Huaning Wang, Xu, Jinfu Liu, et al. (2018), Park et al. (2018), Zheng, Xuebao Li, and Xinshuo Wang (2019), Xuebao Li et al. (2020) and Deng et al. (2021) adopted CNN-based models for solar flare forecasting taking images from magnetograms as inputs.

Given that the flaring AR cycle spans 72 to 96 hours (3 to 5 days) (YU et al., 2009), a separate cohort of researchers focused on deep-learning models built upon RNN, predominantly employing LSTM networks. Notable studies include those by Tang, Zeng, et al. (2020), Xiantong Wang et al. (2020) and Yi et al. (2020).

Subsequent research endeavors have been dedicated to integrating image-based and sequential data deep-learning models, such as combining CNNs with LSTM networks. In line with this strategy, Tang, Liao, et al. (2021) devised a fusion model merging a DNN, a CNN, and an LSTM network to forecast ≥C- and ≥M-class flares. Still, Guastavino et al. (2022) employed image sequences from cropped ARs to forecast ≥M-class flares using a dedicated CNN to each image and an LSTM to process the CNN's outputs as a time series.

In turn, Pengchao Sun et al. (2022) introduced an automated solar flare forecasting system founded on 3D CNN, eliminating the necessity to combine different models. This system can learn spatiotemporal associations within magnetogram sequences and videos to predict ≥M-class flares within the 24 h forecasting horizon.

Although CNN-based methodologies have been predominant, models based on transformers have recently surfaced for image processing and computer vision applications. Original transformer-based models have an encoder/decoder architecture, dispensing convolutional and recurrent operations in favor of stacked self-attention and fully connected layers. Notably, self-attention layers exhibit superior efficiency in terms of computational demand compared to convolutional and recurrent layers (VASWANI et al., 2017).

Kaneda et al. (2023) recently presented the Flare Transformer, designed for solar flare forecasting, which aligns with this evolution. This hybrid system comprises two integrated transformer-based modules: the Sunspot Feature Module, responsible for processing numerical time series inputs, and the Magnetogram Module, responsible for handling image and time-series numerical data from magnetograms.

It is worth noting that most of the studies mentioned earlier focused on training models from scratch, specifically for the solar flare forecasting problem. These models often comprised scaled-down versions of well-established CNN or transformer-based architectures. Additionally, these works typically framed the solar flare forecasting task as a binary classification, distinguished for predicting ≥C- or ≥M-class flares within the 24 h forecasting horizon.

Considering that C-class flares pose minimal space weather threats and given the limited exploration of forecasting horizons beyond 24 h, we concentrated on ≥M-class flares within the 24 h and 48 h forecasting horizon. This approach addresses the heightened significance of ≥M-class flares while extending the forecasting horizon to enhance preparation and response capabilities to related threats.

Moreover, our choice presents additional challenges due to the infrequent occurrence of M- and X-class flares throughout a solar cycle. Previous studies by Xuebao Li et al. (2020) and Deng et al. (2021) tackled this issue by employing data augmentation techniques like oversampling the minority class and subsampling the majority class. However, a notable limitation is that the authors applied these techniques uniformly across the overall dataset, including the test set. This indiscriminate application of data augmentation methods can potentially introduce bias into the attained results. Furthermore, there was a lack of analysis regarding the impact of these data augmentation techniques on the model's performance.

1.1 Hypothesis

Our main hypothesis is that applying fine-tuning training on a general-purpose model tailored for video recognition, considering sequences of magnetogram images, can outperform the current results achieved in solar flare forecasting. We also believe that applying data augmentation only in training and validation sets can improve the evaluation of the test set without creating biased results.

1.2 Objectives

Therefore, this thesis proposes adopting transformer models tailored for general video recognition, using just line-of-sight magnetogram sequenced images as inputs, like video data, to predict ≥M-class flares within the next 24 h and 48 h. Our approach involves fine-tuning these models with data augmentation techniques. We opted for fine-tuning due to the resource-intensive nature of training from scratch transformer-based models, which requires high-performance computational resources and large datasets.

Furthermore, we advocate applying data augmentation techniques to address imbalances within the datasets, particularly concerning the infrequent occurrence of X- and M-class flares. We aim to evaluate the usefulness of these techniques by applying them to the training and validation sets. Notably, our methodology diverges from the prevailing approach in related literature, which primarily focuses on training them from scratch downgraded deep-learning models explicitly tailored for the solar flare forecasting problem.

1.2.1 Specific Objectives

To achieve the primary goal of this thesis, we propose the following specific objectives.

- Build a dataset with magnetogram sequences that allow us to use it in transformer-based models for video recognition.
- Implement transformer-based models for solar flare forecasting, with and without data augmentation techniques, in the training and validation sets.
- Evaluate the improvements obtained with the data augmentation techniques applied.
- Evaluate our models' performance with correlated state-of-the-art.

1.3 Thesis Structure

We organized this thesis into six chapters as follows:

- Chapter 2 presents a theoretical framework of the main concepts related to space weather, deep-learning techniques, and the transformer models we adopted to develop our models.

- Chapter 3 examines a bibliographical survey of the main works concerning solar flare forecasting with deep learning focused on image processing.
- Chapter 4 covers the methodology for implementing our models and the methods for generating the dataset, including the creation of AR magnetogram sequences, and the oversampling techniques assumed.
- Chapter 5 discusses the results, including comparisons between our models and the correlated works.
- Finally, Chapter 6 presents our conclusions.

Chapter 2

Theoretical Framework

This chapter presents the main theoretical concepts related to this thesis. The initial sections define the space weather phenomena, including solar flares and their impacts on Earth. We then briefly introduce Deep Learning, Convolutional Neural Networks (CNN), and transformers' concepts. Subsequently, we present the main image-processing tasks in which we apply Deep Learning techniques in conjunction with some reference models.

2.1 Space Weather

Space weather refers to the interaction of radiation, magnetic fields, and energetic particles with Earth's magnetosphere and upper atmosphere (MCATEER; GALLAGHER; CONLON, 2010). This environment comprises three main parts: the Sun and its atmosphere as the main source of energy; the propagation medium, which is interplanetary space plasmas; and Earth's magnetosphere and upper atmosphere, where the energy generated by the Sun and disseminated through interplanetary space lies (ECHER et al., 2005).

To better understand space weather, it is crucial to review some solar atmosphere concepts, including specific solar phenomena and their respective influences on the Earth's environment (RIBEIRO, 2020). Section 2.1.1 begins this review by examining the Sun's layers.

2.1.1 The Sun's Layers

The Sun is the primary source of space weather in the solar system. The conditions of interplanetary space are influenced by the temporal and spatial fluctuations of both quasi-stationary and transient particles and the electromagnetic emissions originating from the Sun (MESSEROTTI et al., 2009). Its composition comprises 90% hydrogen gas and 10% helium gas ionized at high temperatures, also called plasma (RIBEIRO, 2020).

The solar plasma remains confined due to the balance between its gravitational force, which keeps the matter compacted, and its internal pressure, which expands the matter. As Figure 2.1 shows, the Sun has some layers and phenomena that influence the transfer of solar energy produced (SILVA, 2013).

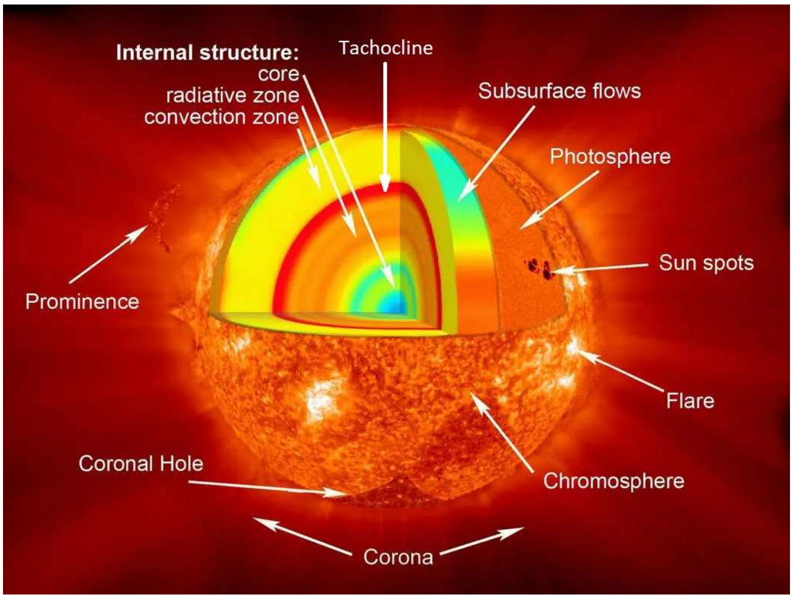


Figure 2.1: Sun layers. Adapted from NASA (2013).

The Sun's core occupies 20% to 25% of its radius and generates energy through nuclear fusion, where hydrogen is converted into helium (SILVA, 2013). This energy propagates out of the solar core through photons emitted by hydrogen and helium ions in the radiative zone, which extends to approximately 70% of the Sun's radius (ECHER et al., 2005; RIBEIRO, 2020).

Then, the plasma on top of the radiative zone rises to the surface of the convection zone. When its temperature decreases at the top of the convection zone, the plasma returns to its base, where it is reheated again. This circulation movement forms thermal columns that transport energy beyond the convection zone. Convection through thermal columns gives a grainy appearance to the photosphere, which is the "visible surface" of the Sun. Figure 2.2 shows an example of this phenomenon with the scale comparison of kilometers and miles.

Still, a thin layer called tachocline is located between the radiative and the convection zones. In this layer, the rotation regime of the Sun's internal structures changes, which is considered the basis of the solar dynamo and is responsible for the formation of intense magnetic fields (SILVA, 2013). These magnetic fields can be visible by forming sunspots.

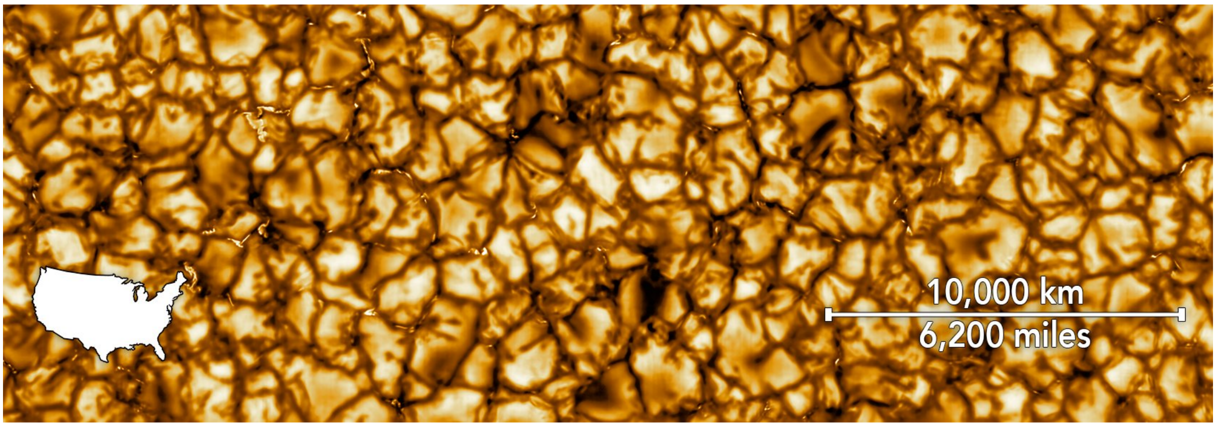


Figure 2.2: Grainy appearance of the photosphere and its comparison with surface size of the USA and the km/miles scales. Source: NSO (2021).

2.1.2 Sunspots

Sunspots are dark regions related to intense – and sometimes complex – magnetic fields visible in the photosphere (CINTO, 2020). A well-developed sunspot consists of a dark center, called the umbra, and a lighter dark part around it, called the penumbra. Figure 2.3 highlights the umbra and penumbra of a gigantic sunspot – nearly 128,000 kilometers across – seen in the lower center of the Sun in an image captured by the Solar Dynamics Observatory (SDO) NASA's mission on October 23rd, 2014. It was considered one of the largest sunspots in Solar Cycle 24, which began in 2008 and ended in 2020.

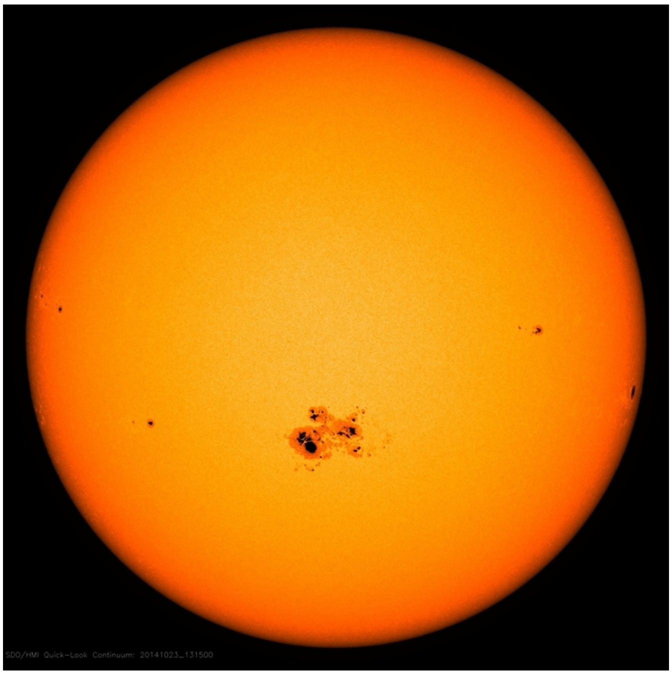


Figure 2.3: Sunspots. Source: NASA (2014).

Because the strong magnetic field nullifies convection in the magnetized plasma, sunspots have a lower temperature and radiation in visible light than their surroundings. Then sunspots appear as dark areas in white light images. We can also consider sunspots as temporary phenomena, lasting from a few days to several weeks, or even months in some cases (FANG, Y.; CUI; AO, 2019).

The frequency of sunspot occurrences presents a cyclical behavior lasting approximately 11 years. It is also known as solar cycles, which occur because of the Sun's magnetic dynamo process (PARKER, 1955), the Sun's magnetic polarity reversal approximately every 11 years (BABCOCK, 1961), and its related differential rotation regimes (HOWARD; LABONTE, 1980).

Figure 2.4 presents a graph provided by the Space Weather Prediction Center (SWPC) with the monthly number of sunspot occurrences concerning the time and their respective solar cycles. In this figure, it is possible to observe a decrease in solar activity between cycles 23 and 24. Solar Cycle 25, which began at the end of 2020 (HAUTALUOMA et al., 2020), has shown more significant solar activity than Solar Cycle 24 but still less activity than Solar Cycle 23.

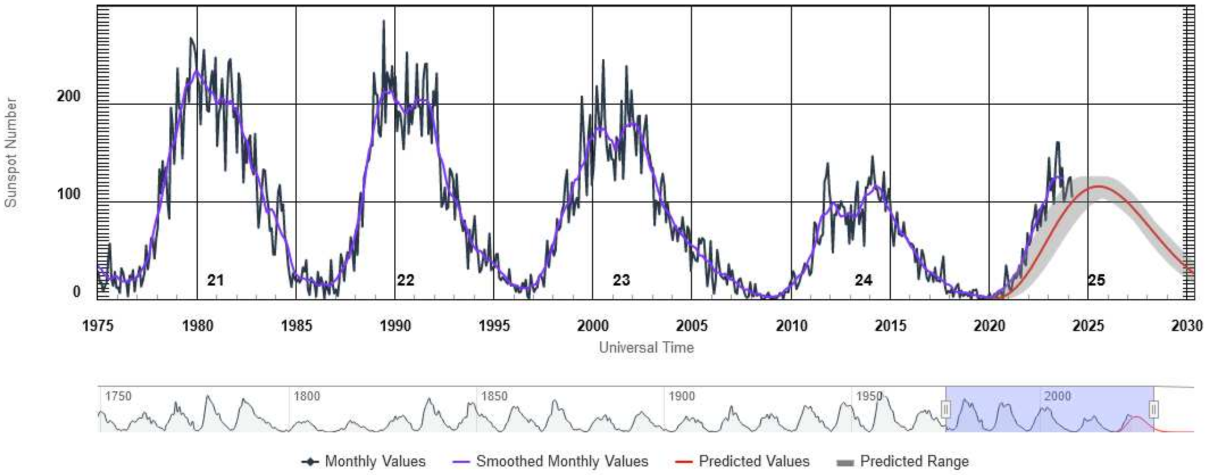


Figure 2.4: Solar cycles progression. Captured from: NOAA/SWPC (2024)

At the beginning of a solar cycle (solar minimum), sunspots typically form at higher solar latitudes, around 30 – 35 degrees north and south of the solar equator. As the cycle progresses and solar activity increases, these sunspots emerge closer to the equator. When the solar maximum is reached (the period of most significant sunspot activity), the sunspots are located at latitudes closer to 10 – 15 degrees. Toward the end of the cycle, sunspots appear nearer the equator, forming the "Butterfly Diagram". Figure 2.5 shows the "Butterfly Diagram" formations across the solar cycles from 1870 to 2015 (HATHAWAY, 2015).

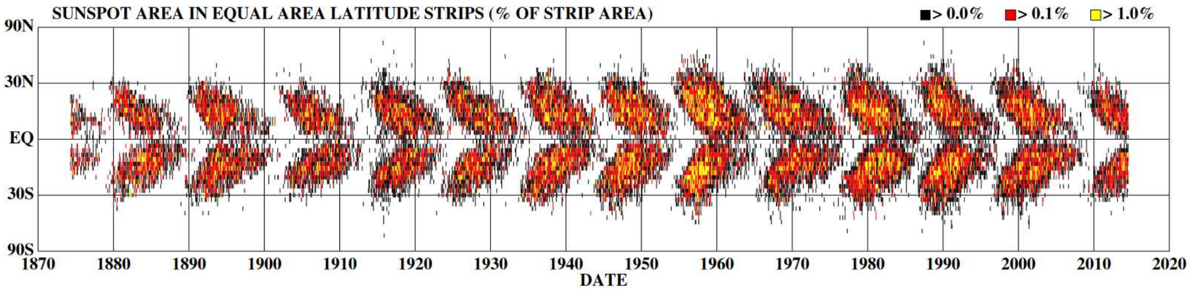


Figure 2.5: The Butterfly Diagram. Source: Hathaway (2015)

Sunspots arise due to the different rotation regimes in the tachocline, causing moving charges to generate currents that induce magnetic fields in the lower layers of the Sun (radiative zone and convection zone). These magnetic fields cross the photosphere, where the convective process is inhibited. It results in the emergence of sunspots that act as the “signature” of their presence (SILVA, 2013) and extend to the chromosphere until the solar corona (the upper layers of the Sun), forming magnetic arcs.

The phenomenon of magnetic arcs, represented in the diagram in Figure 2.6, occurs when spots of different polarities come together. Magnetograms can show sunspots with different polarities approaching each other.

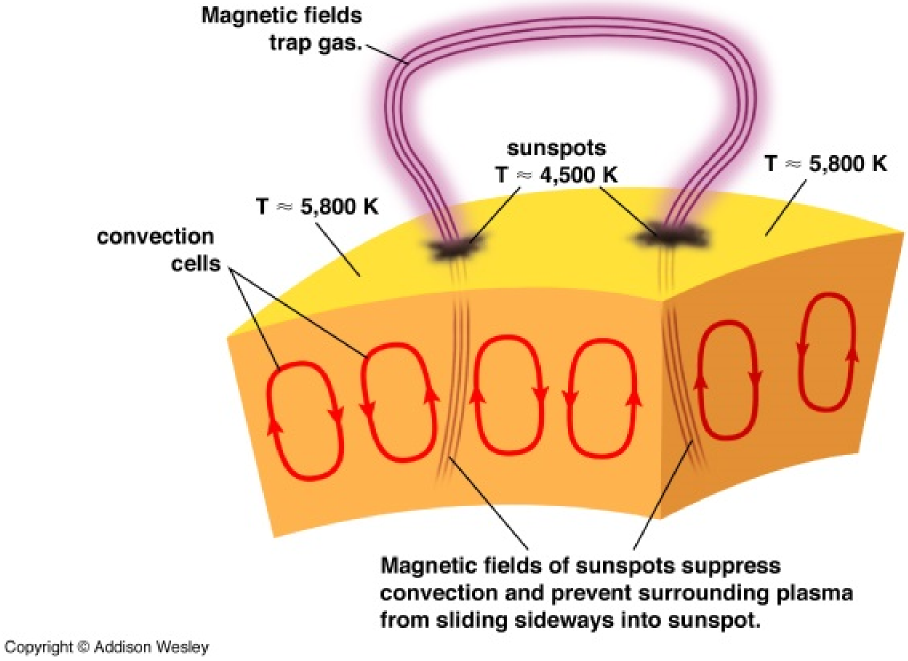

Magnetic fields
trap gas.
sunspots
T ≈ 4,500 K
T ≈ 5,800 K
T ≈ 5,800 K
convection
cells
Magnetic fields of sunspots suppress
convection and prevent surrounding plasma
from sliding sideways into sunspot.
Copyright © Addison Wesley

Figure 2.6: Formation of magnetic fields and magnetic arcs. Source: Hentzau (2012).

2.1.3 Magnetograms

A solar magnetogram is an image obtained by an instrument capable of detecting the location, strength, and direction of magnetic fields on the Sun. In a magnetogram, neutral areas are gray, indicating no magnetic field. Typically, the darker areas are regions of negative or south magnetic polarity (directing to the Sun's core), and the lighter areas are regions indicating positive or north polarity (directing to Earth) (LANG, 2006).

Figure 2.7-A highlights a magnetogram captured by the instrument Helioseismic and Magnetic Imager (HMI) and the respective formation of magnetic arcs (Figure 2.7-B) captured by the Atmospheric Imaging Assembly (AIA) instrument at the wavelength of 171 Ångström (Å)[1], from the same sunspot shown in Figure 2.3. The HMI and AIA are the Solar Dynamics Observatory instruments.

In the magnetogram Figure 2.7-(A), the regions with negative polarity are evident in black, and those with positive polarity are in white. In Figure 2.7-(B), it is possible to observe the formation of magnetic arcs between the closest regions of different polarities. When magnetic arcs form, the groups of sunspots corresponding to these arcs are called Active Regions. The Helioviewer portal captured the two images, which refer to October 23rd, 2014, at 17:00:16 Coordinated Universal Time (UTC).

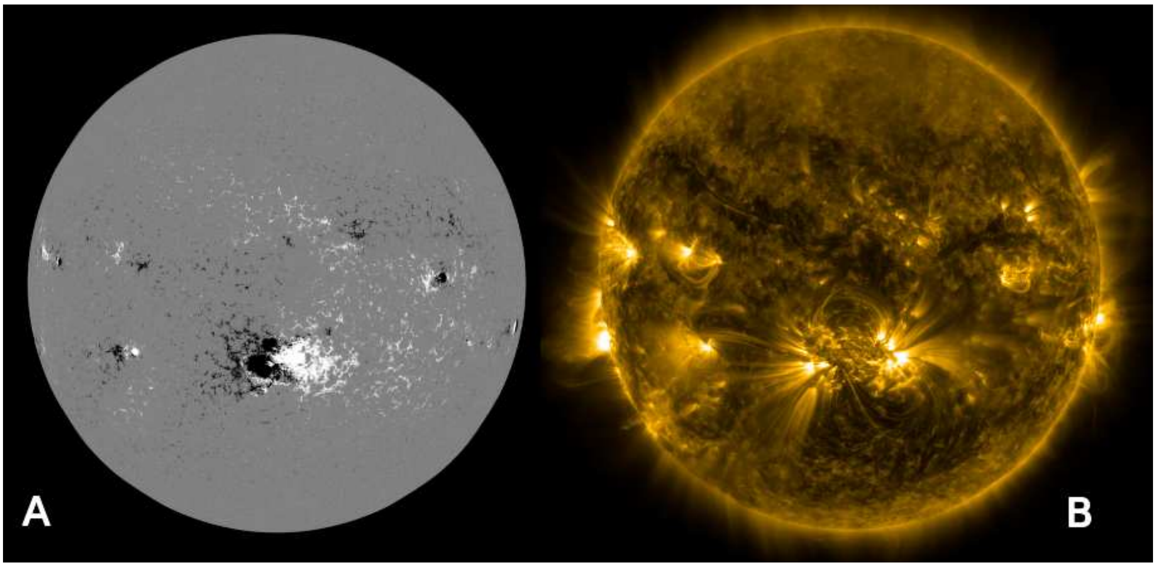


Figure 2.7: The Magnetogram (A) shows the spots of different polarities close together, and image B shows the formation of magnetic arcs. Source: Helioviewer.org (2021).

[1] It is a metric unit of length equal to 10^{-10} m.

2.1.4 Active Regions and SHARP Patches

Active regions are initiated when magnetic arcs emerge from within the Sun, cross the photosphere through sunspots, and reach the chromosphere, potentially reaching the solar corona. The magnetic structure of an AR changes appearance as new magnetic arcs on the surface of sunspots move and shift. Over weeks or months, the magnetic arcs break, disintegrate, or move into the Sun, where they came from (LANG, 2006).

To classify and distinguish the magnetic configuration of ARs, the taxonomies of Mt. Wilson (HALE et al., 1919) and McIntosh (1990) are mainly used. Mt. Wilson's taxonomy classifies ARs according to the magnetic fields within them, as summarized by Oliveira and Gradvohl (2020). On the other hand, McIntosh (1990)'s taxonomy describes the shapes of ARs through three distinct components (Zpc), as summarized by Cinto (2020).

Space-weather HMI Active Region Patches (SHARPs) data series provide maps in patches covering the entire lifespan of automatically tracked ARs, including numeric vector magnetic fields and line-of-sight magnetogram images. SHARP indices are computed for each segment and are updated every twelve minutes. Preliminary data can be accessed roughly three hours after observation, while final scientific products are generated about five weeks later (BOBRA; SUN, X., et al., 2014). SHARP data are available online[2] at Joint Science Operations Center (JSOC) repository. Since its release, several authors have adopted SHARP parameters in solar flare forecasting, as we can see in more detail in Chapter 3.

ARs have dynamic behavior and continually change their magnetic shape. Stressed magnetic fields accumulate magnetic energy waiting to be released, and the ongoing magnetic interaction can trigger events such as powerful solar flares (LANG, 2006) or Coronal Mass Ejection (CME), which are phenomena that can cause impacts on Earth.

2.1.5 Solar Flares

Solar Flares are intense and temporary energy releases in the solar atmosphere. According to Echer et al. (2005) and Shibata and Magara (2011), evidence suggests that the primary source of energy for solar flares is the breaking and reconnection of strong magnetic fields that emit sudden releases of electromagnetic radiation – including X-rays at wavelengths of 1 Å to 8 Å, whose intensities are represented in Watts per square meter (W/m^2).

[2] Available at `jsoc.stanford.edu`

These events can reach temperatures between 10 and 20 million Kelvin (CINTO, 2020) and are among the most forceful in the solar system. Solar flares influence a panorama of physical systems from the Sun's photosphere through the heliosphere and up to the geospace (MCATEER; GALLAGHER; CONLON, 2010), which is also known as the solar-terrestrial environment and comprises the upper atmosphere, the Ionosphere, and the magnetosphere, where Sun-Earth interactions occur.

Solar flare events are classified on a logarithmic scale in the X-ray band (1 Å to 8 Å) varying between classes A, B, C, M, and X, according to their peak intensity (I), as Table 2.1 shows. Each class of solar flare has its peak X-ray flux ten times greater than the previous class.

Table 2.1: Solar flares' classes. Source: Ribeiro (2020)

Flare class	X-ray peak flux in band 1 Å to 8 Å, in W/m^2
A	$I < 10^{-7}$
B	$10^{-7} \leq I < 10^{-6}$
C	$10^{-6} \leq I < 10^{-5}$
M	$10^{-5} \leq I < 10^{-4}$
X	$10^{-4} \leq I$

Furthermore, each flare class has a sub-classification varying from 1 to 9, representing the event's intensity. Therefore, to report solar flares, the peak values are multiplied by their intensity factors, i. e., X9, X2, and M6 (M6 means 6×10^{-5} W/m^2). It is worth noting that the most remarkable X-class solar flare ever cataloged was classified as X.28 and occurred in November 2003 (ECHER et al., 2005).

To illustrate the occurrence of solar flares, Figure 2.8 shows a time series with X-ray fluxes captured by Geostationary Operational Environmental Satellite (GOES) over three days. The red curve corresponds to the 1 Å to 8 Å band wavelength observed by GOES−15, and its peaks show a sequence of solar flares between 2017 September 5 and 8.

Figure 2.8 shows that on September 5th, there was a sequence of M-class flares, followed by an X2.2 and another X9.3 flare on September 6th. Soon after, on September 7th, an M7.3 and an X1.3 flare occurred. Figure 2.9 shows the X9.3 flare on September 6th, captured by the SDO AIA in 171 Å wavelength, which was the most significant in Solar Cycle 24 (REDMON, R. J. et al., 2018).

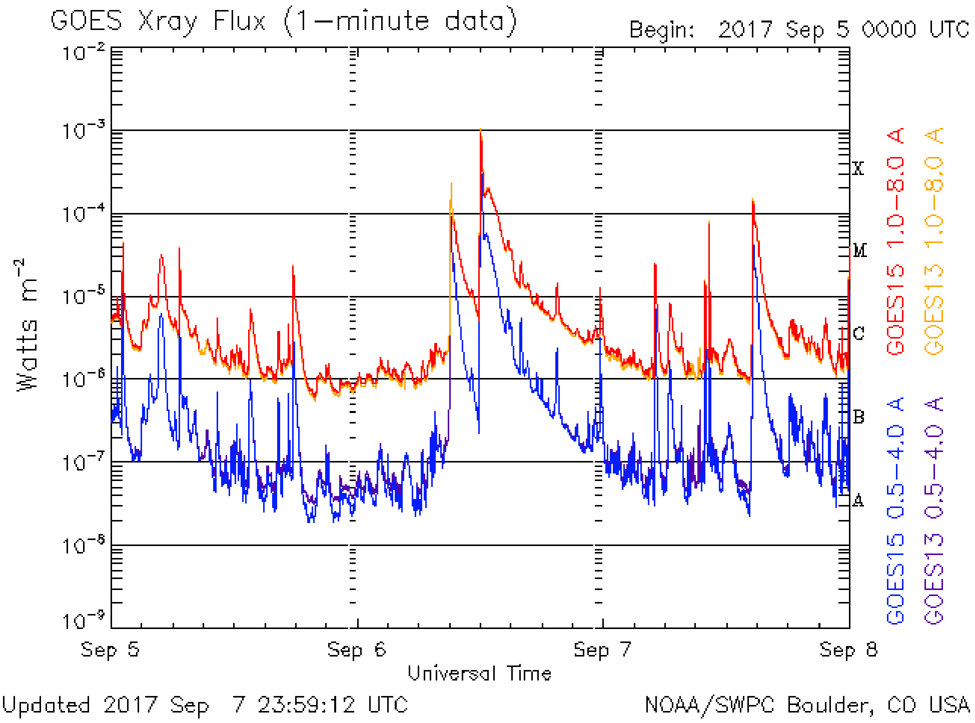


Figure 2.8: X-ray fluxes captured by GOES−15 satellites indicating the occurrence of X- and M-class flares. Source: NOAA/SWPC (2017).

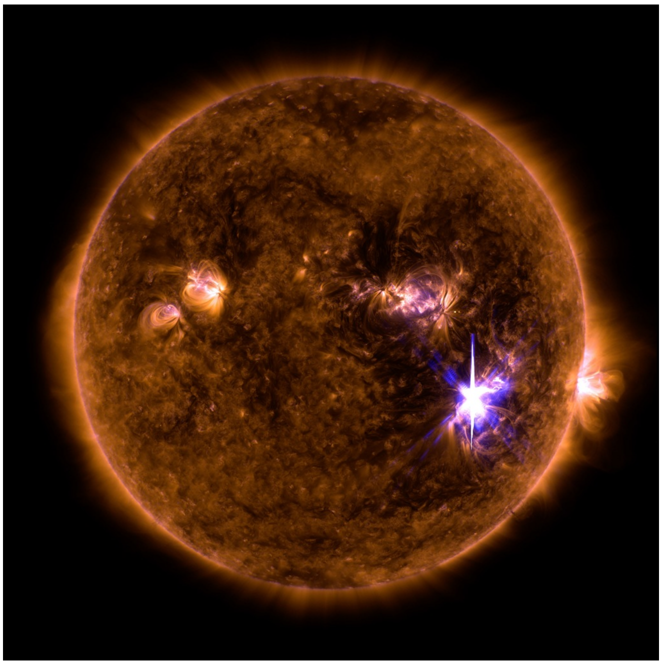


Figure 2.9: X9.3 flare on September 6, 2017 in SDO AIA 171 Å. Source: NASA (2017).

Hence, solar flares (and often related CMEs) can be considered the most powerful events in the Solar System, with the radiation and particles emitted by these explosions interacting strongly with the Earth's ionosphere and magnetosphere (MESSEROTTI et al., 2009), as described in Section 2.1.6.

2.1.6 Solar Flares' Impacts on Earth

The region between planets and the Sun is called the interplanetary medium, dominated by the solar wind, which flows from 250 km/s to 1,000 km/s. Near Earth, the solar wind is a flow of ionized particles coming from the Sun, with a speed typically around 400 km/s to 500 km/s (ECHER et al., 2005). Solar winds result from the pressure difference between the solar corona and interplanetary space (RIBEIRO, 2020). Figure 2.10 illustrates Sun–Earth interactions during space weather events.

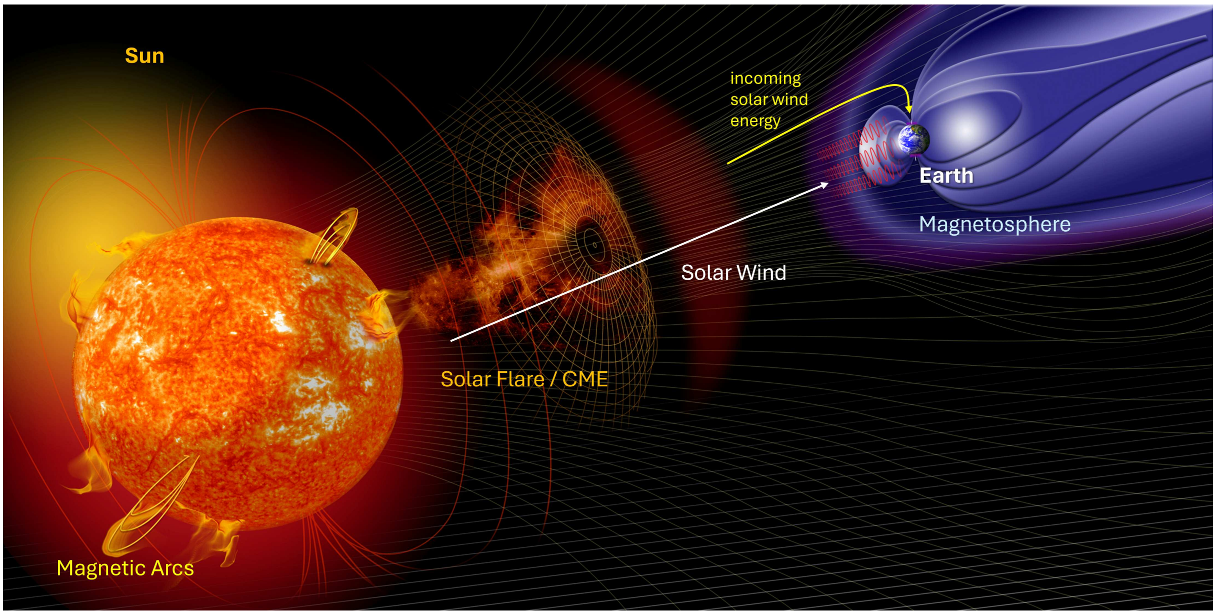


Figure 2.10: Sun – Earth interactions. Adapted from NASA (2024).

The magnetized solar wind flows continuously by the interplanetary medium and interacts with planetary magnetic fields, delimiting their magnetospheres (ECHER et al., 2005). The Earth's magnetosphere and atmosphere work as a natural shield, protecting us from most solar threats. The Earth's magnetosphere deflects the solar energetic particles, while the atmosphere, including its ozone layer, blocks the high-energy solar radiation.

However, the high-energy particles and radiation associated with X- and M-class flares can threaten the astronauts' safety, reduce the satellites' life, disrupt radio communications, and deteriorate the Global Positioning System (GPS) accuracy since they orbit the Earth above their magnetosphere. After a solar flare, their radiation arrives on Earth in around 8 minutes, and their high-energy particles arrive in about 30 minutes. Therefore, forecasting solar flares is critical to provide sufficient time to respond to the effects of space weather (HUANG, X.; WANG, H.; XU; SUN, W., 2017).

As for the effects on the ground, X-class flares can affect systems and communications that reflect waves in the Ionosphere (in addition to those based on satellites) and cause electrical power failures, for example. M-class flares can cause minor radiation storms and brief radio blackouts (ECHER et al., 2005). C-, B-, and A-class flares have few effects on Earth. Figure 2.11 illustrates these and other effects arising from solar flares on Earth.

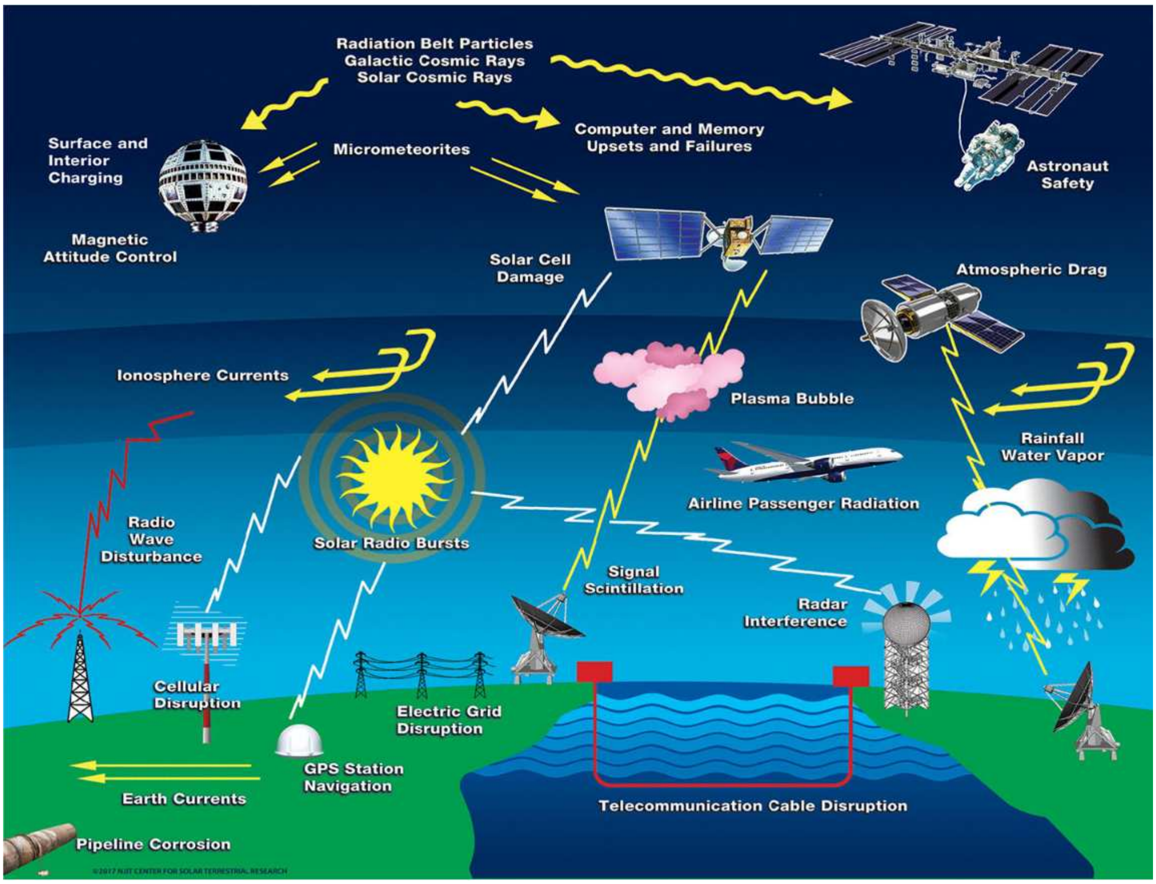


Figure 2.11: Impacts of solar flares on Earth. Source: Lanzerotti (2017).

Considering the situation depicted before, the need to employ efforts in prediction systems for X- and M-class flares is evident, as forecasting the occurrence of these flares helps mitigate their effects on Earth. Traditionally, several SWPCs, such as NOAA–SWPC[3] and Instituto Nacional de Pesquisas Espaciais (INPE)–EMBRACE[4], work on solar flare forecasting. These centers generally employ human-based hybrid forecasts, adopting expert systems that output cadenced forecasts, further confirmed or adjusted by solar physicists (CINTO, 2020).

However, machine learning become a hot topic in solar flare forecasting, with researchers achieving good results in the past twenty years. The surveys presented by Messerotti et al.

[3] Available at `https://www.swpc.noaa.gov/`

[4] Available at `https://www2.inpe.br/climaespacial/portal/pt/`

(2009) and Leka et al. (2019), as well as the works of Bobra and Couvidat (2015) and Cinto, Gradvohl, et al. (2020b), make this evident. Despite this, a striking characteristic of most of these works is the extensive and laborious data pre-processing step and the choice and combination of input features for a classic machine learning system.

Since 2018, several works have used deep learning for solar flare forecasting, as we can see in Nishizuka, Sugiura, Kubo, Den, and Ishii (2018), Xuebao Li et al. (2020), Tang, Liao, et al. (2021), Pengchao Sun et al. (2022) and Kaneda et al. (2023). Some of these works significantly reduce the effort required for pre-processing and input selection features, working directly with AR magnetograms. Therefore, the following sections cover the main deep-learning concepts and techniques for image processing.

2.2 Deep Learning

Deep-learning comprises a range of machine-learning methods supported by Artificial Neural Networks (ANN), which are versatile, robust, and scalable. These are essential attributes for handling large and complex machine learning tasks (LECUN; BENGIO; HINTON, G., 2015), such as processing images, videos, audio, and natural language texts.

2.2.1 Artificial Neural Networks

The main concepts of the artificial neuron and the main components of ANNs emerged in the 1940s by McCulloch and Pitts (1943). In 1951, Marvin Minsky created the first neurocomputer, called the Stochastic Neural Analog Reinforcement Calculator (SNARK) and in 1967 also published a book called “Computation: Finite and Infinite Machines” (MINSKY, M. L., 1967), which extended the results of McCulloch and Pitts (1943).

Between 1951 and 1967 Rosenblatt (1958) developed the Perceptron, a single-layer ANN that introduced a novel method for addressing the pattern recognition problem. In summary, the Perceptron receives multiple input signals $(x_1, x_2, \dots, x_m)$, each associated with a correspondent weight $(w_{k1}, w_{k2}, \dots, w_{km})$. Then, a weighted sum of these inputs is computed and passed through an activation function $f(v_k)$, typically a threshold function, to generate an output $\hat{y}_k$, as illustrated in Figure 2.12.

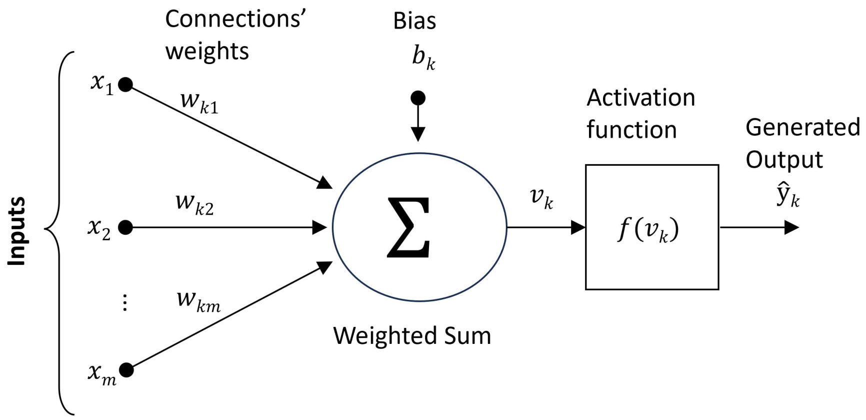


Figure 2.12: The Perceptron. Adapted from Haykin (2001).

The Perceptron introduced an innovative method of supervised learning (HAYKIN, 2001): the Perceptron Learning Rule, which adjusts the inputs' weights ($w_{k1}, w_{k2}, \dots, w_{km}$) based on the error between the generated output $\hat{y}_k$ and the target value y_k (ground truth).

During the height of the classic Perceptron period of the 1960s, researchers considered that ANNs could accomplish anything. However, M. Minsky and Papert (1969) used mathematics to reveal the essential limits of what single-layer Perceptrons can calculate (HAYKIN, 2001), i. e., they cannot implement simple functions like XOR.

Despite the limitation of dealing only with linearly separable problems, the Perceptron laid the groundwork for more complex ANN architectures with multiple layers and learning algorithms, like the Backpropagation algorithm, explained in the next section.

2.2.2 Backpropagation Algorithm

The Backpropagation algorithm came into evidence in the 1980s, emerging as an efficient learning algorithm for training multi-layer Perceptrons. Most of the credit is attributed to Rumelhart, Geoffrey E. Hinton, and Williams (1986) who proposed its use in machine learning and demonstrated how it could work despite the evidence that it had been previously described in other works from the late 1960s to the mid-1970s (HAYKIN, 2001).

Over time, Backpropagation became the most popular algorithm for training ANNs, particularly in multi-layer architectures. It involves two principal phases: the forward pass and the backward pass. During the forward pass, the input x is propagated over the network for computing the output $\hat{y}$. Figure 2.13 illustrates the equations to calculate the forward pass in an ANN with two hidden layers and one output layer.

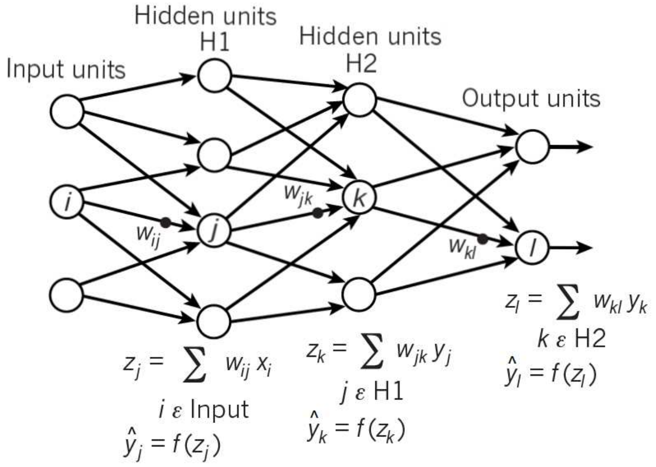


Figure 2.13: Forward pass. Adapted from LeCun, Bengio, and Geoffrey Hinton (2015).

In each layer, it initially calculated the total input z, for each unit. This z value results from a weighted sum ($\sum w$) of the outputs from the units in the preceding layer. Subsequently, a non-linear function, $f(.)$, is applied to z, yielding the unit's output (LECUN; BENGIO; HINTON, G., 2015). We omitted the bias to simplify this example.

Common non-linear functions employed in neural networks comprise the Rectified Linear Unit (ReLU) and traditional Sigmoid functions, such as the Hyperbolic Tangent and the Logistic Function, according to equations 2.1, 2.2, and 2.3 respectively.

$$\text{ReLU}(z)^+ = \max(0, z) \tag{2.1}$$

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \tag{2.2}$$

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{2.3}$$

In the backward pass, the error E between the generated output $\hat{y}$ and the target value y is propagated backward through the network using the chain rule to compute the gradients concerning the network parameters. Figure 2.14 illustrates the equations of the backward pass.

The output layer calculates the error derivative to a unit's output by differentiating the cost function. Suppose the cost function for unit l is given by $(\frac{1}{2}(\hat{y}_l - y_l)^2)$, then the derivative for the output of the unit l is $(\hat{y}_l - y_l)$. Since the partial derivative $(\frac{\partial E}{\partial z_k})$ is determined, the error derivative to the weight (w_{jk}) connecting unit (j) from the previous layer is $(\hat{y}_j \frac{\partial E}{\partial z_k})$ (LECUN; BENGIO; HINTON, G., 2015).

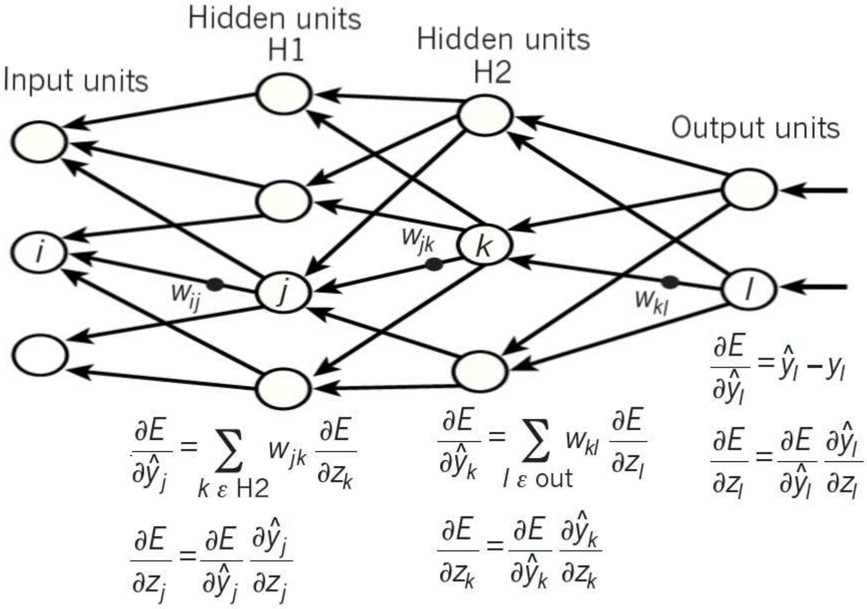

Hidden units H1
Hidden units H2
Input units
Output units
i
j
k
l
w_{ij}
w_{jk}
w_{kl}
$\frac{\partial E}{\partial \hat{y}_l} = \hat{y}_l - y_l$
$\frac{\partial E}{\partial z_l} = \frac{\partial E}{\partial \hat{y}_l} \frac{\partial \hat{y}_l}{\partial z_l}$
$\frac{\partial E}{\partial \hat{y}_j} = \sum_{k \varepsilon H2} w_{jk} \frac{\partial E}{\partial z_k}$
$\frac{\partial E}{\partial z_j} = \frac{\partial E}{\partial \hat{y}_j} \frac{\partial \hat{y}_j}{\partial z_j}$
$\frac{\partial E}{\partial \hat{y}_k} = \sum_{l \varepsilon \text{out}} w_{kl} \frac{\partial E}{\partial z_l}$
$\frac{\partial E}{\partial z_k} = \frac{\partial E}{\partial \hat{y}_k} \frac{\partial \hat{y}_k}{\partial z_k}$

Figure 2.14: Backward pass. Adapted from LeCun, Bengio, and Geoffrey Hinton (2015).

The error derivative concerning each unit's output is calculated at each hidden layer. It involves summing the weighted error derivatives ($\sum w\frac{\partial E}{\partial z}$) for all units in the preceding layer. Next, the error derivative of the output $\hat{y}$ is transformed into the error derivative of the input x by multiplying it by the $f(z)$ gradient (LECUN; BENGIO; HINTON, G., 2015).

2.2.3 Supervised Learning

Supervised learning involves establishing a relationship between a collection of input variables, denoted as x, and an output variable, y. This relationship is then used to forecast outputs for new, unseen data. It is considered one of the most crucial techniques within the field of machine learning (CUNNINGHAM; CORD; DELANY, 2008).

To achieve this relationship, in the supervised learning method, we try to adjust the connections' layer weights w, which are usually initially assigned randomly, in such a way as to approximate the generated output values $\hat{y}$ to come as close as possible to the desired output values y.

Figure 2.15 illustrates an example of the supervised learning method in a single-layer ANN, where the Training Data comprises the inputs x and the known outputs y. The attribute x_0 represents an input sample associated with a Bias, w_0 is the bias weight, w_1 is the sample weight, and $\hat{y}$ represents the output generated by the network. J represents the Cost Function, which, in this example, is the Mean Squared Error (MSE).

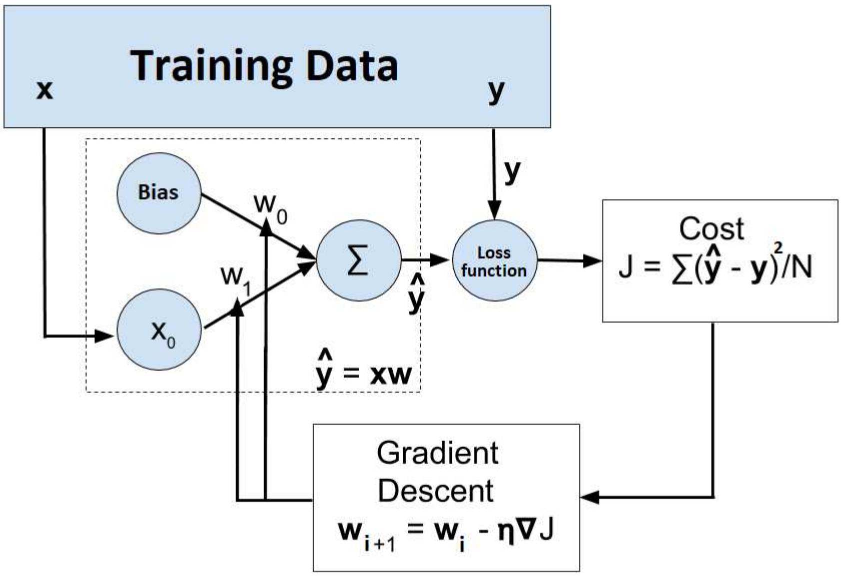


Figure 2.15: Supervised training scheme. Adapted from Lotufo (2019).

The supervised learning process occurs through consecutive training steps, also known as epochs, in which the weights of the next epoch (w_{k+1}) are adjusted based on the weights of the current epoch (w_k) subtracted from a factor of learning (learning rate − η) multiplied by the Descending Gradient, starting from the Cost Function (∇J), backpropagated to the inputs.

In summary, deep-learning ANNs comprise multiple hidden layers capable of learning data representations with several levels of abstraction without prior domain knowledge or the need for feature extraction. This procedure has become feasible through backpropagation algorithms like the Gradient Descent (LECUN; BENGIO; HINTON, G., 2015).

2.2.4 Convolutional Neural Network

Deep-learning models came into the spotlight for drastically improving the results of the computer vision field in the 2012 Imagenet competition, specifically with the AlexNet CNN (KRIZHEVSKY; SUTSKEVER; HINTON, G. E., 2017). Deep CNNs have made considerable progress in image processing, such as pattern-based classification and object detection (LECUN; BENGIO; HINTON, G., 2015).

CNNs are feedforward ANNs, as the flow of information occurs in a single direction, from inputs to outputs (GÉRON, 2017). Since 2012, researchers made considerable advances in other domains of sequential data processing, such as text and speech, with emphasis on Recurrent Neural Networks (RNN), such as Long Short-Term Memory (LSTM) neural networks.

The architecture of a deep CNN consists of convolutional layers followed by non-linear ReLU functions and pooling layers grouped into units. The convolutional layer's function is to identify local combinations of features (feature maps) from the previous layer, while the pooling layer consolidates semantically similar features by merging them. One or more fully connected layers follow these stacked units, forming a deep model, and the backpropagation algorithm adjusts the weights in training (LECUN; BENGIO; HINTON, G., 2015).

Although they can present many variations, the architectures of most CNNs consist of these units mentioned above (Convolutions, ReLU, and Pooling) followed by a fully connected layer. Figure 2.16 illustrates the typical architecture of a CNN for an image classification task.

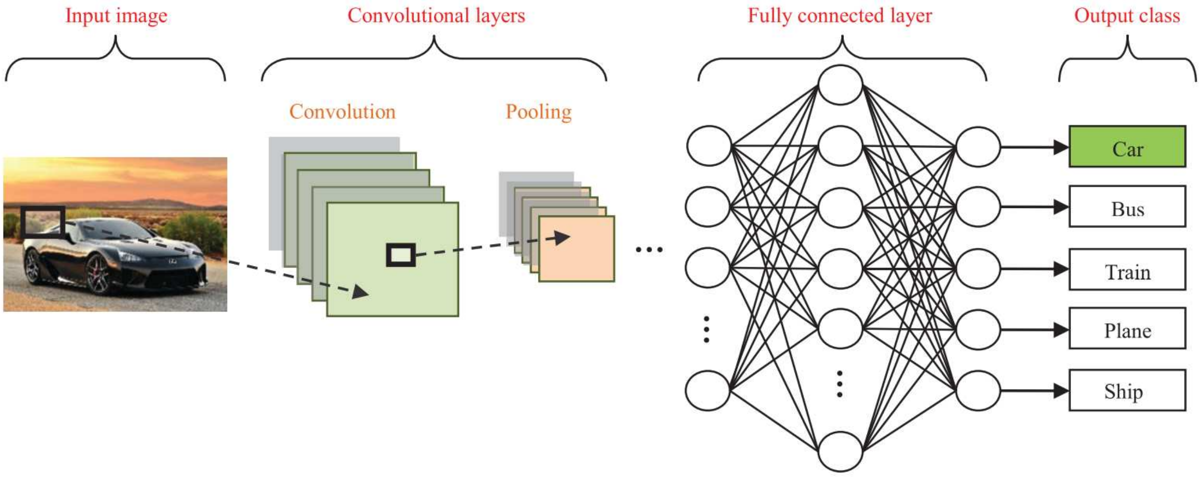


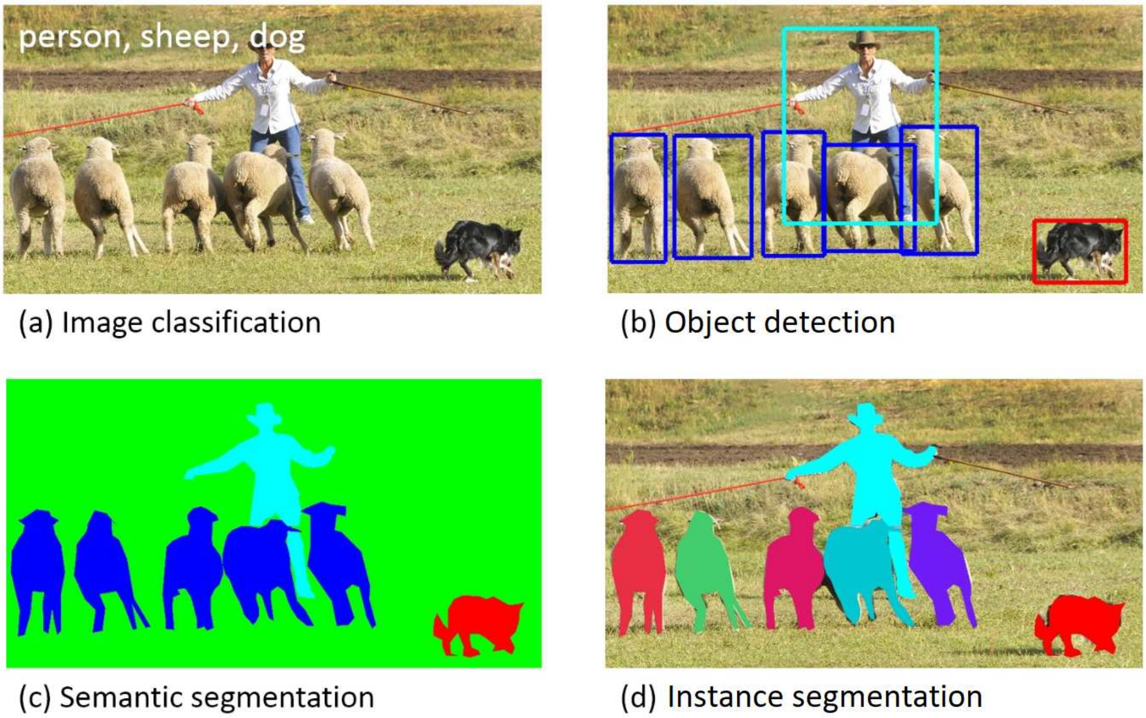
Figure 2.16: Typical architecture of a CNN. Source: Rawat and Zenghui Wang (2017).

LeCun, Bengio, and Geoffrey Hinton (2015) credit CNNs' success to the efficient use of Graphical Processing Units (GPU), ReLUs, techniques to increase training samples by deforming existing ones (oversampling), and the dropout technique. The dropout technique consists of zeroing the output of each hidden neuron with a specific probability (generally 0.5). This way, the "eliminated" neurons do not contribute to learning. Despite this, using dropout practically doubles the number of iterations needed for the network to converge.

Due to the increase in computational power and training data, in recent years, CNNs have managed to achieve performance superior to that of humans in some complex visual tasks (GÉRON, 2017), such as image classification, object detection, and image segmentation, as discussed in Section 2.2.5.

2.2.5 Image Processing with Deep Learning

As Figure 2.17 shows, we can divide the most common image-processing tasks related to deep learning into four groups: image classification, object detection, semantic segmentation, and instance segmentation in images.



(a) Image classification

(b) Object detection

(c) Semantic segmentation

(d) Instance segmentation

Figure 2.17: Common tasks in image processing. Adapted from Lin et al. (2014).

The image classification task consists of assigning images in a collection to their corresponding classes (categories), according to Figure 2.17-(a). Therefore, its objective is to predict accurately the target class of new images. For this task, training sets for image classification require a label that indicates which class an image belongs to. Practitioners consider it the most popular task in image processing, and it has numerous deep-learning models aimed at its application, such as the pioneer AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, G. E., 2017) and the already consolidated ResNets (HE; ZHANG, et al., 2016) and EfficientNets (TAN; LE, 2019).

The object detection task involves declaring that an object belonging to a specified class is present and locating it in the image. The location of an object is usually represented by a bounding box (LIN et al., 2014), as Figure 2.17-(b) illustrates. The Region-Based Convolutional Neural Network (R-CNN) and their extensions (Fast R-CNN, Faster R-CNN, Mask R-CNN) have proven to be successful in applications for object detection (MINAEE et al., 2021).

We can define the semantic segmentation task as classifying each pixel of an image into a class based on a desired context, as Figure 2.17-(c) shows, in which the algorithm segmented all sheep with the blue mask. Whereas, in instance segmentation, the primary purpose is to identify each of the objects in the image with a specific label or color mask, even if they belong to the same class in a given context, as Figure 2.17-(d) shows, in which the algorithm segmented each of the sheep with a specific color mask.

According to Minaee et al. (2021), several methods and techniques for image segmentation have emerged, most of them based on CNN models. Among these models, variations of Faster R-CNN, such as Mask R-CNN, are popular. As a Faster R-CNN extension, the Mask R-CNN (HE; GKIOXARI, et al., 2020) is a model that can detect objects in an image and, at the same time, generate a segmentation mask for each instance. We can also use the Mask R-CNN model in semantic segmentation with some adaptations. However, the U-Net (RONNEBERGER; FISCHER; BROX, 2015) model, initially proposed for semantic segmentation in medical images, has also proven to be quite versatile and effective for general-purpose semantic segmentation tasks.

2.2.6 Video Classification/Recognition Tasks

We can extend the image processing tasks explained in Section 2.2.5 to videos or image sequences, such as video classification/recognition tasks, illustrated in Figure 2.18.

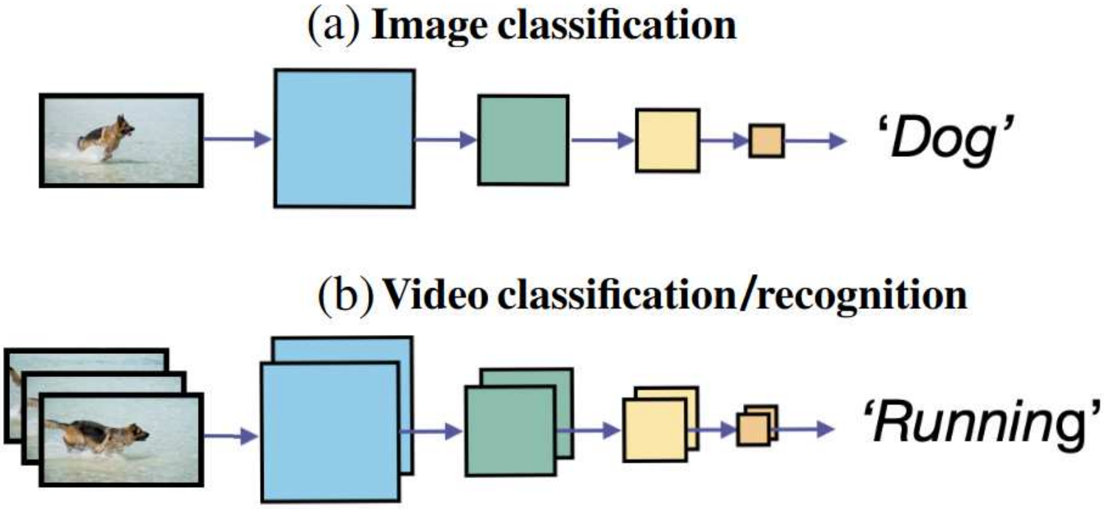


Figure 2.18: Video classification/recognition task. Adapted from Yanghao Li et al. (2022).

There are two basic approaches to video classification based on CNNs. Early approaches consisted of hybrid models composed of CNN + LSTM, such as the one proposed by Wu et al. (2015), in which there is a fusion of features extracted from two CNNs, one responsible for extracting features from spatial movements and the other responsible for extracting features of short-term movements. Subsequently, the features extracted from each CNN are considered time series inputs to LSTM networks, which acted to model more long-term temporal features.

Another CNN approach to video classification relies on the concept of 3D convolutions, as is the case with ResNet 3D and ResNet MC (Mixed Convolution), ResNet (2+1)D (TRAN et al., 2018) to learn spatiotemporal correlations in sequences of images or videos. The specific case of ResNet (2+1)D presents 3D convolutional filters factored into separate spatial and temporal components. This procedure produced accuracy gains over ResNet 3D and ResNet MC around 2% to 5% (TRAN et al., 2018).

Despite the predominance of CNNs, transformer-based models have also been proposed for image-processing tasks in the last few years. Therefore, Section 2.3 describes the evolution of transformers from the original to those dedicated to image processing.

2.3 Transformer Models for Image Processing

The transformer models use an attention-based approach to establish global relationships between the input and output, dispensing convolutional and recurrent layers. In the original transformer design, the encoder and decoder components consist of self-attention layers and fully connected neural networks stacked on each other (VASWANI et al., 2017). This design often outpaces recurrent and convolutional layers in processing speed, particularly in sequence-based tasks like language understanding and machine translation.

The encoder takes in a series of symbolic inputs $(x_1, \dots, x_n)$ and transforms them into continuous representations $\mathbf{z} = (z_1, \dots, z_n)$. Based on $\mathbf{z}$ representations, the decoder creates an output sequence of symbols $(y_1, \dots, y_m)$. The authors design the attention mechanism as a function that takes a query (Q) and a set of key-value pairs (K, V) to produce an output. This process is defined in Equation 2.4 (VASWANI et al., 2017) and Figure 2.19-(a):

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{2.4}$$

where d_k refers to the queries and keys dimensions and K^T is the transpose of K.

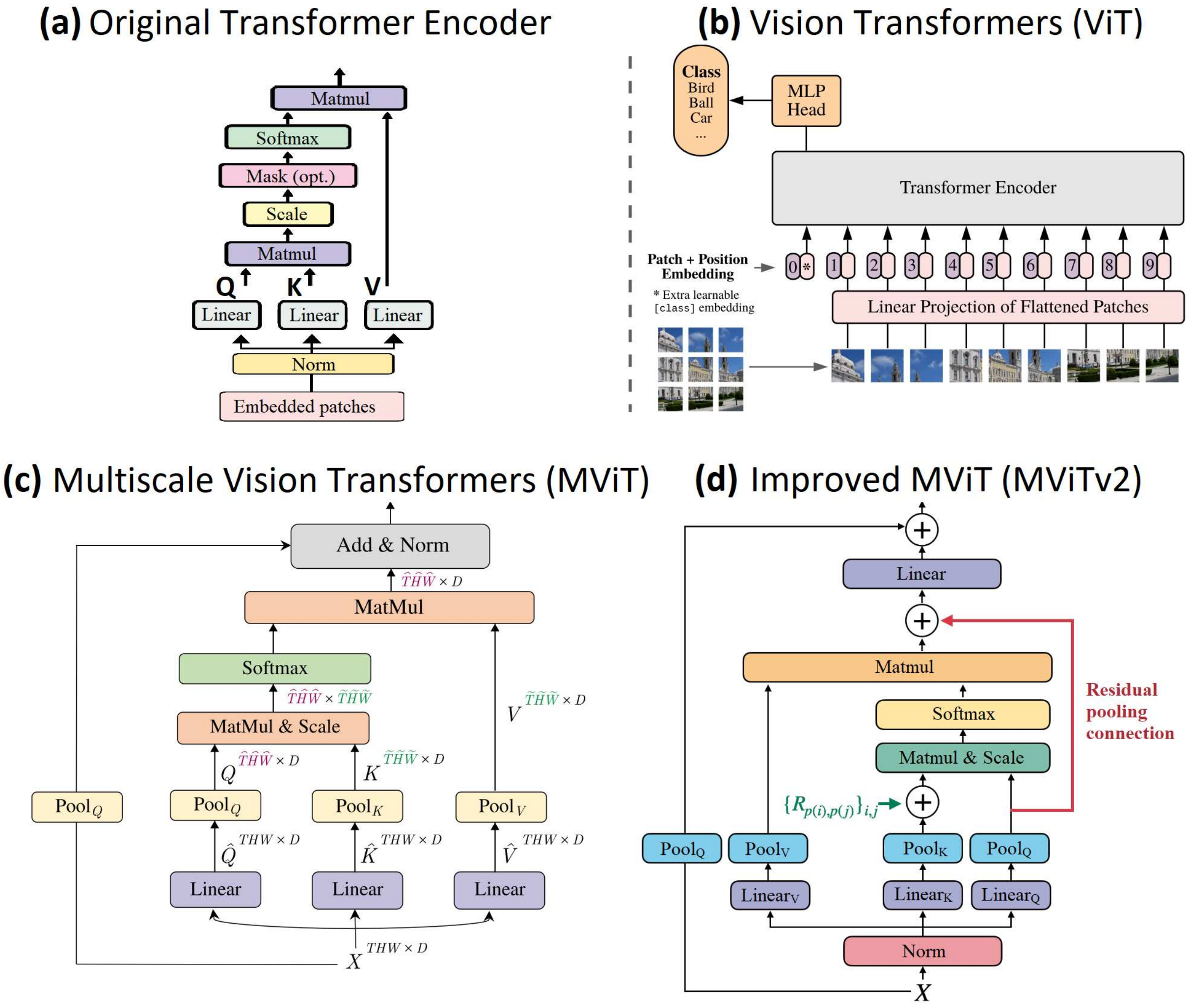


Figure 2.19: Transformer models for computer vision. Adapted from Vaswani et al. (2017), Dosovitskiy et al. (2021), Fan et al. (2021) and Yanghao Li et al. (2022).

The Vision Transformer (ViT) (DOSOVITSKIY et al., 2021) took the original transformer model, which processes sequences in one dimension, and adapted it for image classification by applying the transformer architecture to image patch sequences. ViT's key innovation was partitioning an image at fixed-size patches, transforming the image $\mathbf{x} \in \mathbb{R}^{H\times W\times D}$ into a series of flattened patches. Each patch is linearly embedded, creating a sequence $\mathbf{x}_p \in \mathbb{R}^{N\times(P^2\cdot D)}$, where (H, W) represent the original image's dimensions, D denotes the number of channels, and (P, P) specifies the size of each patch (DOSOVITSKIY et al., 2021).

To maintain positional information, the authors introduced a 1D positional embedding vector to the sequence of embeddings. This sequence then served as the input to a default transformer encoder. They also added a special learnable "classification token" ($\mathbf{z}_0^0 = \mathbf{x}_{\text{class}}$) for classification tasks. The final classification step was performed using a Multi-Layer Perceptron (MLP) posterior to the transformer encoder, as Figure 2.19-(b) depicts.

ViT demonstrated versatility by combining the transformer's fundamental operations with techniques like normalization, pooling, and residual connections, adapting to different tasks like video classification. In this context, Fan et al. (2021) developed the Multiscale Vision Transformers (MViT) to extend ViT's capabilities, targeting video analysis and classification.

While ViT divides the image into small chunks (typically 16 × 16 pixels) that are treated as individual word-like "tokens" in a natural language processing model, MViTs goes a step further and operates at multiple resolution scales, allowing it to extract patches at different sizes to capture both local and global details. This conversion is part of the model's forward pass process. MViTs are designed to transform the image into patches early in its architecture before passing the patches to the transformer layers. In simple terms, the model expects to receive the image in its usual format (as a 3D tensor: Channels, Height, and Width) and then works to divide and process it into small chunks.

When dealing with video, the model works not only with spatial (2D) patches but also with spatiotemporal (3D) patches (Height, Width, Time). This means that the model needed to process more than what an image looks like in each frame. It also needed to know how that appearance changes over time. In the same way that the ViT model divides an image into 2D patches and treats them as "tokens", MViTs transform 3D cubes (or spatiotemporal patches) into tokens. These tokens carry both spatial and temporal information.

It allows the MViTs to capture patterns of movements and temporal interactions over time. In the Transformer, the self-attention mechanism is primarily responsible for learning the relationships between tokens. In the MViT video models, this self-attention observes the relationship between spatial and temporal tokens. In addition to multi-scale attention, MViT can also perform temporal downsampling, which means it can reduce the temporal resolution at different stages of the model, focusing on the essential parts of the video as it progresses through the layers.

2.3.1 Multiscale Vision Transformers

The Improved Multiscale Vision Transformers (MViTv2) is an evolution of MViT, both designed to capture spatial and temporal patterns in videos or image sequences, extending ViT by a multistage feature hierarchy from higher to lower resolution (LI, Y. et al., 2022).

To implement downsampling inside a transformer block, MViT uses Pooling Attention, gradually increasing the channel width D while decreasing the overall length ($L = T \times H \times W$), where (H, W) represent the initial image resolution and T is the number of images in the sequence. Given an input sequence $X \in \mathbb{R}^{L \times D}$, MViT applies linear transformations via matrices W_Q, W_K, and W_V (all of size $\mathbb{R}^{D \times D}$), followed by pooling operations ($\mathcal{P}$) to queries (Q), keys (K), and values (V). Equations 2.5, 2.6, and 2.7 outline these matrices.

$$Q = \mathcal{P}_Q(XW_Q), \tag{2.5}$$

$$K = \mathcal{P}_K(XW_K), \tag{2.6}$$

$$V = \mathcal{P}_V(XW_V), \tag{2.7}$$

where $\mathcal{P}_Q$ reduces the length of Q, resulting in $Q \in \mathbb{R}^{\tilde{L} \times D}$, while $\mathcal{P}_K$ and $\mathcal{P}_V$ similarly downsample K and V (LI, Y. et al., 2022).

Pooled self-attention then creates the output sequence $Z \in \mathbb{R}^{\tilde{L} \times D}$ with variable length (FAN et al., 2021), as expressed in Equation 2.8.

$$Z := \text{Attention}_{\text{MViT}}(Q, K, V) = \text{Softmax}\left(\frac{QK^\top}{\sqrt{D}}\right)V. \tag{2.8}$$

MViTv2 introduces a new feature with decomposed relative position embedding $R_{p(i),p(j)} \in \mathbb{R}^d$, which represents the spatial or spatiotemporal relationship between the input elements i and j. Yanghao Li et al. (2022) used this within the self-attention module as descibed in Equation 2.9.

$$\text{Attention}_{\text{MViTv2}}(Q, K, V) = \text{Softmax}\Big(\frac{QK^\top + E^{(\text{rel})}}{\sqrt{d}}\Big)V, \tag{2.9}$$

where, the relative position encoding, $E_{ij}^{(\text{rel})} = Q_i \cdot R_{p(i),p(j)}$, contributes to the self-attention computation. To simplify this process, we need to break down the distance calculation between elements i and j along the spatiotemporal axes according to Equation 2.10 (LI, Y. et al., 2022):

$$R_{p(i),p(j)} = R^{\text{h}}_{h(i),h(j)} + R^{\text{w}}_{w(i),w(j)} + R^{\text{t}}_{t(i),t(j)}, \tag{2.10}$$

where R^h, R^w, and R^t represent the relative positional embeddings along the height, width, and temporal dimensions, with $h(i)$, $w(i)$, and $t(i)$, indicating the respective positions of token i.

MViTv2 also incorporates a residual pooling connection within its attention blocks, combining the pooled query tensor with the resulting sequence Z, as Equation 2.11 shows (LI, Y. et al., 2022):

$$Z := \text{Attention}_{\text{MViTv2}}(Q, K, V) + Q. \tag{2.11}$$

Figure 2.19-(c) and -(d) represent an overview of MViT and MViTv2 attention blocks.

2.4 Concluding Remarks

This chapter defined the main theoretical concepts of this thesis, such as space weather, solar flares, and the need for efforts to predict these events capable of generating significant impacts. The text also presented the main concepts of deep learning, which enabled meaningful advances in image-processing tasks.

The intense Sun's magnetic formations can be viewed on the photosphere as dark regions called Sunspots through magnetograms. They are a temporal phenomenon and can last from days to weeks. When two or more Sunspots approach each other, they form an Active Region. Some kinds of active regions are related to the occurrence of solar flares.

Sunspots can be considered the feet of magnetic arcs in the photosphere. Since the literature links the solar flare events with the disruption or reconnection of these magnetic arcs, we can infer that a sequence of line-of-sight magnetogram images can contain implicit spatial and temporal parameters of the ARs evolution and its related magnetic arcs' future behavior.

The most intense solar flares (from M- and X-classes) can significantly impact Earth and the near Earth's atmosphere. Forecasting the occurrence of M- and X-class flares in advance is crucial to mitigating their potential impacts. Several works have applied deep-learning techniques to solar flare forecasting in the last few years, considering line-of-sight magnetogram data as inputs.

As deep-learning techniques concerning image processing are a hot topic in the solar flares forecasting area, Chapter 3 presents the most relevant works encompassing these issues, considering the classification of solar flares based on magnetogram images.

Chapter 3

Literature Review

This chapter presents the literature review of important solar flare forecasting works. Section 3.1 presents some precursor works based on statistical methods and classic machine-learning techniques, frequently cited and compared within works presented in Section 3.2. Then, Section 3.2 discusses works that use deep-learning techniques, focusing on image processing, which receives magnetogram solar images as inputs.

3.1 Precursor Works

There are several works dedicated to solar flare forecasting that employ different methods. In the last fifteen years, we have had expert systems based on the McIntosh (1990) classification, going through works that employ classical statistical distributions and reaching those that use machine learning techniques (BARNES, G. et al., 2016), in addition to those that combine two or more different methods. G. Barnes et al. (2016), Camporeale (2019) and Cinto, Gradvohl, et al. (2020a) present an extensive review of the main recently published works. Here we considered works that do not use deep learning techniques as precursor works.

Therefore, this section reviews the precursor works that are most compared and cited by the works that use deep-learning techniques, presented in Section 3.2. It is worth noting that directly comparing these works would be unfair due to the different techniques, methods, and datasets they use (BARNES, G. et al., 2016; CINTO; GRADVOHL, et al., 2020b). The comparison becomes even more complex due to the small amount of X- and M-class flare events, which cause a significant class imbalance in the general solar flare forecasting datasets. Given this scenario, we observed, in most cases, the attempt to make comparisons between works that use similar techniques, datasets, or even the division of the datasets.

To evaluate the performance of all related work highlighted here, we consider the metrics Accuracy (ACC), Probability of Detection (POD – also known as Recall), False-Alarm Ratio (FAR), Critical Success Index (CSI), Heidke Skill Score (HSS), and True Skill Statistic (TSS), obtained through a Confusion Matrix. Appendix A shows how we compute these metrics and the motivations to consider them since most of these works are binary classification problems with highly imbalanced datasets.

Table 3.1 presents the precursor work surveyed in this thesis for predicting ≥M-class flares, which includes M and X flares, and ≥C-class flares, which includes C-, M-, X flares. For comparison purposes, we only considered the 24 h and 48 h Forecasting Horizon (FH), although Colak and Qahwaji (2009) also considered the 6 h, 12 h, and 72 h FH.

Table 3.1: Precursor works.

Authors	FH	Class	ACC	POD	FAR	CSI	HSS	TSS
Bloomfield et al. (2012)[⋄]	24 h	$\geq$ M	0.83	0.70	0.85	—	0.19	0.54
Bobra and Couvidat (2015)[×]	24 h	$\geq$ M	0.92	0.83	—	—	0.52	0.76
Chang Liu et al. (2017)[§]	24 h	$\geq$ M	0.77	0.75	0.25	0.61	0.53	0.53
Nishizuka, Sugiura, Kubo, Den, Watari, et al. (2017)[†]	24 h	$\geq$ M	0.99	0.91	0.07	0.85	0.92	**0.91**
Bloomfield et al. (2012)[⋄]	24 h	$\geq$ C	0.71	0.75	0.65	—	0.32	0.45
Colak and Qahwaji (2009)[‡]	24 h	$\geq$ C	0.81	0.81	0.30	—	0.51	—
Ahmed et al. (2013)[⊙]	24 h	$\geq$ C	0.96	0.52	0.26	—	0.60	**0.53**[⊗]
Colak and Qahwaji (2009)[‡]	48 h	$\geq$ C	0.74	0.86	0.17	—	0.49	—

[⋄] Results considering the optimum TSS in Table 4 of Bloomfield et al. (2012).

[×] Results considering the optimum TSS in Table 3 of Bobra and Couvidat (2015). The authors did not inform the FAR and CSI.

[§] Results computed through Table 4 of Chang Liu et al. (2017).

[†] Results calculated from the confusion matrix of Table 3 of Nishizuka, Sugiura, Kubo, Den, Watari, et al. (2017), referencing the KNN model.

[‡] Results collected from Table 3 of Colak and Qahwaji (2009). The authors did not mention the CSI and TSS.

[⊙] Results considering the Operational scenario in the Training and Testing set, from Table 6 of Ahmed et al. (2013). The authors did not mention the CSI and TSS.

[⊗] TSS computed with the TPR and TNR of Table 6 of Ahmed et al. (2013), as indicated by Cinto (2020).

Bloomfield et al. (2012) was the most cited and compared work in our literature review. They used X-ray fluxes from the GOES satellite to calculate the averaged solar flares from solar cycles 21 and 22 for each group in McIntosh (1990)'s classification. Subsequently, they applied the Poisson distribution to calculate the probability of different solar flare classes from Solar Cycle 23, considering the 24 h forecast horizon and optimizations for HSS and TSS.

Considering the results for the optimal TSS, the predictions for ≥M-class flares reached ACC = 0.83, POD = 0.70, HSS = 0.19, and TSS = 0.54, but with a FAR = 0.85 high. For ≥C-class flares, it presented a similar scenario with ACC = 0.71, POD = 0.75, HSS = 0.32, TSS = 0.45, and FAR = 0.65. In addition to good results for the time of its publication, Bloomfield et al. (2012) raised a discussion regarding performance metrics. They recommended using TSS instead of HSS, which until then was considered the primary metric for comparing results in solar flare forecasting works.

3.1.1 Works Adopting Machine Learning

Except for Bloomfield et al. (2012), all the other precursor works adopted machine-learning techniques. As the second most cited work, Colak and Qahwaji (2009) introduced the Automated Solar Activity Prediction (ASAP), which is composed of two modules. In the first module, Colak and Qahwaji (2008) perform image processing by classifying ARs according to the McIntosh (1990) taxonomy. The second module is a machine learning system composed of two neural networks: one responsible for predicting whether an AR will cause a ≥C-class flare and another for predicting the correct flare class (C, M, or X).

The input of the second module was the AR classes and the respective sunspot area group provided by the first module. The input for training the first ASAP module was *Continuum* magnetogram images from the Michelson Doppler Imager (MDI) project of the Solar and Heliospheric Observatory (SOHO) mission (previous to HMI/SDO, deactivated in 2011), cataloged solar flare data from National Geophysical Data Center – NGDC (now National Centers for Environmental Information – NCEI) and cataloged data from National Oceanic and Atmospheric Administration (NOAA).

ASAP was assessed with images from 1999 to 2002, presenting ACC = 0.81, POD = 0.81, FAR = 0.30, and HSS = 0.51 for predicting ≥C-class flares within the 24 h forecasting horizon and ACC = 0.74, POD = 0.86, FAR = 0.17, and HSS = 0.49 for ≥C-class flares within the 48 h forecasting horizon. The authors did not present values for the TSS metric. They also considered 6 h, 12 h, and 72 h forecast horizons and compared the Quadratic Error in some cases with the forecasts made by NOAA Space Weather Prediction Center (SWPC), in which ASAP proved superior.

In turn, Bobra and Couvidat (2015) used machine-learning techniques with Support Vector Machines (SVM) to forecast ≥M-class flares in an operational and segmented scenario, as was defined in Ahmed et al. (2013). The authors used numeric data from the Space-weather HMI Active Region Patches (SHARP) magnetograms (BOBRA; SUN, X., et al., 2014) from May 2010 to May 2014. Instead of dividing them chronologically, they randomly divided the training and testing sets into 70% and 30%, respectively.

At its publication, it achieved ACC = 0.92, POD = 0.83, HSS = 0.52, and TSS = 0.76 for the operational scenario without information on the FAR value. However, they optimized the TSS value regarding the test set, which may have caused biased results. As Bloomfield et al. (2012), Bobra and Couvidat (2015) also recommended TSS as the primary metric for evaluating performance in solar flare forecasting works.

Chang Liu et al. (2017) adopted the Random Forest algorithm to predict solar flares in AR within the 24 h forecasting horizon. The authors considered data from HMI SHARP and GOES X-ray fluxes between May 2010 and December 2016 from AR between ±70° of the Sun's central meridian. Here, we consider the ≥M-class flares results. They also subsample the negative samples and create 100 data subsets. Instead of separating a specific set for testing, they evaluated the classifier's performance through a 10-fold cross-validation scheme, achieving ACC = 0.77, POD = 0.75, FAR = 0.25, CSI = 0.61, HSS = 0.53, and TSS = 0.53.

Nishizuka, Sugiura, Kubo, Den, Watari, et al. (2017) compared three different machine-learning algorithms for predicting ≥M-class flares within the 24 h forecasting horizon: SVM, k-Nearest Neighbors (k-NN) and Extremely Randomized Trees (ERT). The authors used data from HMI magnetograms, ultraviolet emission from the AIA, and GOES X-ray emission data from 2010 to 2015, randomly dividing the training and testing set into 70% and 30%. They also considered scenarios for the full solar disk, ±53° and ±37° of the Sun's central meridian. The best result for the full solar disk scenario was from the k-NN algorithm, with ACC = 0.99, POD = 0.91, FAR = 0.07, CSI = 0.85, HSS = 0.92, and TSS = 0.91.

However, the authors themselves claim "that the validation set acts as a test set, but technically it is called a validation set" (NISHIZUKA; SUGIURA; KUBO; DEN; WATARI, et al., 2017) as they are using the test set to optimize performance. This decision causes a drastic performance drop in operational conditions (unknown test set). Nishizuka, Sugiura, Kubo, Den, and Ishii (2018) and subsequent works reported this issue.

Finally, Ahmed et al. (2013) used data defined as Magnetic Features (MF), composed of magnetic features tracked and extracted from SOHO/MDI images by Solar Monitor Active Region Tracker (SMART) (HIGGINS et al., 2011) from April 1996 to December 2010. The authors considered only magnetograms between $\pm 45^{\circ}$ of the Sun's central meridian. Most of the works presented in Section 3.2 followed and referenced this trend. They used a machine-learning algorithm known as the Cascade Correlation Neural Network.

Ahmed et al. (2013) obtained the following results: ACC = 0.96, FAR = 0.26, HSS = 0.60, and TSS = 0.53, considering the dataset's chronological division for training and testing and the operational sample scenario.

3.2 Solar Flare Forecasting with Deep Learning

In this section, we survey the main works that use deep learning for solar flare forecasting, focusing on image processing techniques. These works apply CNNs to magnetograms or other solar images.

Nishizuka, Sugiura, Kubo, Den, and Ishii (2018) and Xin Huang, Huaning Wang, Xu, Jinfu Liu, et al. (2018) were the first to apply deep-learning techniques to solar flares forecasting. Despite not operating directly on the magnetogram images, Deep Flare Net (DeFN) calculates the probability of ≥C-class and ≥M-class flares occurring within the 24 h forecasting horizon for each AR. The core of the DeFN model consists of a Deep Neural Network (DNN) with shortcut connections similar to the concept of ResNets (HE; ZHANG, et al., 2016) composed of seven units, each with a linear layer and a ReLU activation function.

There is a Batch Normalization (BN) (IOFFE; SZEGEDY, 2015) procedure between all units. DeFN took 79 manually feature-extracted numerical attributes from AR-cropped magnetograms as inputs. It extracted the AR tracking system and part of the attributes according to previous work by Nishizuka, Sugiura, Kubo, Den, Watari, et al. (2017). Nishizuka, Sugiura, Kubo, Den, and Ishii (2018) also added new attributes and, in general, DeFN processes attributes extracted from four different SDO image sources and instruments: HMI magnetograms, HMI Continuum, AIA with a 1600 Å filter, and AIA with a 131 Å filter.

The authors considered data between 2010 and 2014 for training and the year of 2015 for testing. However, they did not separate a specific validation set, stating that they obtained the best model optimizing the TSS on the test set. Thus, the results were ACC = 0.82, POD = 0.81, FAR = 0.47, CSI = 0.47, HSS = 0.53 and TSS = 0.63 for ≥C-class flares and ACC = 0.86, POD = 0.95, FAR = 0.82, CSI = 0.18, HSS = 0.26, and TSS = 0.80 for ≥M-class flares.

3.2.1 Pioneer Works Considering Magnetogram Images as Inputs

Xin Huang, Huaning Wang, Xu, Jinfu Liu, et al. (2018) was the first to apply deep-learning techniques to solar flare forecasting directly in captured magnetogram images as inputs, eliminating the necessity to extract numerical parameters from the magnetograms. The authors proposed a simple CNN model, with just two units composed of their respective convolution layers, a ReLU function, and pooling layers, followed by a fully connected layer.

They used a dataset from magnetograms of AR tracked by SOHO/MDI and SDO/HMI from April 1996 to October 2015. The SOHO/MDI images were used for training, while the SDO/HMI images formed the test set. Xin Huang, Huaning Wang, Xu, Jinfu Liu, et al. (2018) considered only ARs between ±30° of the Sun's central meridian and optimized the cost function on the training set itself. The work presented results within the forecasting horizons of 6 h, 12 h, 24 h, and 48 h for predicting ≥C-class, ≥M-class, and X-class flares.

The results for the 24 h forecasting horizon with ≥C-class flares were ACC = 0.75, POD = 0.73, FAR = 0.65, CSI = 0.31, HSS = 0.34, and TSS = 0.49. For the same horizon with ≥M-class flares, they obtained: ACC = 0.81, POD = 0.85, FAR = 0.90, CSI = 0.10, HSS = 0.14, and TSS = 0.66. For the 48 h forecasting horizon with ≥M-class flares, they achieved ACC = 0.81, POD = 0.81, FAR = 0.84, CSI = 0.14, HSS = 0.20, and TSS = 0.62.

As in Nishizuka, Sugiura, Kubo, Den, and Ishii (2018), high FAR significantly impacts CSI and HSS. The authors also explain how the deep-learning model focuses on areas with magnetic polarity inversion lines or regions with strong magnetic fields in ARs through the feature map generation by CNN. This procedure allows the direct use of the magnetogram images for solar flare forecasting.

In the same context, Park et al. (2018) used full-disk magnetograms rather than tracked AR-cropped magnetograms with data from SOHO/MDI (May 1996 to Dec. 2010) and SDO/HMI (Jan. 2011 to June 2017) divided chronologically for training and testing. The training set comprised Solar Cycle 23 (1996 to 2008), and the test set partially comprised Solar Cycle 24

(2009 to 2017). They compared three different CNNs for predicting ≥C-class flares within the 24 h forecasting horizon: AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, G. E., 2017), GoogLeNet (SZEGEDY et al., 2014), and a new proposed CNN model (Model 3).

Model 3 combines GoogLeNet and DenseNet (HUANG, G. et al., 2017) and comprises modules with six convolutional blocks. Each convolutional block has six convolution layers and direct connectivity with other blocks. Inside the block, after the convolutional layers, there are BN layers preceded by ReLU functions.

Park et al. (2018) compared the results to works that use the same training and testing techniques but different learning methods and models, such as statistical methods, traditional machine learning, and deep learning. In general, the results they obtained were superior to compared works that used chronological order to divide the training and test sets. Model 3 achieved ACC = 0.82, POD = 0.85, FAR = 0.17, CSI = 0.73, HSS = 0.63, TSS = 0.63. They reached a considerable improvement in FAR, CSI, and HSS regarding previous works.

Xin Huang, Huaning Wang, Xu, Jinfu Liu, et al. (2018) and Park et al. (2018) also evaluated their models with 10-fold cross-validation. They randomly separated both datasets into ten parts, using nine parts as the training set and one part as the test set. Then, they repeated the operation ten times to calculate its average performance and the standard deviation. In both works, the results were slightly better than the chronological division into training and testing. The standard deviations were also notably small.

On the other hand, Ali K Abed, Qahwaji, and Ahmed Abed (2021) proposed the ASAP_Deep, a hybrid system based on Colak and Qahwaji (2009) for automatic detection of AR and solar flare forecasting of ≥C-class with a 24 h horizon. The system comprises three units. Unit 1 used the Colak and Qahwaji (2008) method for AR automatic detection and has solar intensity-gram and magnetogram images from SDO/HMI between ±45° of the Sun's central meridian as inputs. Unit 2 cuts, fills, and labels groups of sunspots detected by Unit 1. Unit 3 is a CNN with five convolutional layers, each followed by a pooling layer. The authors also reported that they used the dropout technique.

Ali K Abed, Qahwaji, and Ahmed Abed (2021) trained the ASAP_Deep with data from May 2010 to December 2011 and January 2014 to May 2017. The authors used data from 2012 and 2013 for testing because they consider this period to include many relevant solar flares. The results were ACC = 0.80, POD = 0.90, FAR = 0.21, CSI = 0.72, HSS = 0.84, and TSS = 0.89, being the work that obtained the best HSS and TSS in this survey.

3.2.2 Works Adopting Resampling Techniques

Zheng, Xuebao Li, and Xinshuo Wang (2019) and Xuebao Li et al. (2020) proposed a new way to treat the training and testing set. First, they divide the HMI/SHARP data cataloged by NOAA AR from May 2010 to September 2018 and assign a classification (Non-flare, C-, M-, or X-class) according to the GOES X-ray peak within the 24 h forecasting horizon. Then, each block of its respective class has the AR randomly shuffled to be divided into 80% for training and 20% for testing, according to the AR. It is worth noting that no data on the same AR in the training set was replicated in the test set and vice versa.

Then, the researchers subsampled the "Non-flare" and C-class flare blocks. The M-class and X-class flare blocks passed through a data augmentation process (image rotation and mirroring) to reduce the unbalanced class ratio. After the resampling, they divided the data blocks again for training (80%) and testing (20%). They repeated the entire process ten times to make ten distinct datasets.

Zheng, Xuebao Li, and Xinshuo Wang (2019) compared two CNN models based on the VGG network (SIMONYAN; ZISSERMAN, 2015). Model 1 has been revised to work on single-channel images and undergoes some adjustments to the layer arrangement. Model 2 is a hybrid model composed of two sub-models. Model 2−1 has eight units: the first five comprise a convolutional layer followed by batch normalization, a ReLU, and a pooling layer at the end. Units 6 and 7 have a fully connected layer followed by batch normalization and a dropout.

Starting from Model 2−1, the work of Zheng, Xuebao Li, and Xinshuo Wang (2019) further proposed two other CNN models (Model 2−2 and Model 2−3) to generate a multi-class forecasting system for solar flares, inspired by Colak and Qahwaji (2009). The results for Model 2−1 (i. e., ≥M-class flares over the 24 h forecasting horizon) were ACC = 0.89, POD = 0.82, FAR = 0.11, HSS = 0.76, and TSS = 0.75. Xuebao Li et al. (2020) also trained and evaluated Model 2−1 for ≥C-class flares, achieving ACC = 0.86, POD = 0.89, FAR = 0.09, HSS = 0.67 and TSS = 0.68. Zheng, Xuebao Li, and Xinshuo Wang (2019) and Xuebao Li et al. (2020) generally presented better results than those compared. However, they used resampling on the test set, which can lead to positively biased results.

Deng et al. (2021) innovated using Generative Adversarial Network (GAN) networks to artificially increase X-class flares magnetogram images. A GAN consists of one ANN as the data generator (G) and another ANN as data discriminator (D). They compete, taking available training data as input and generating new data with a certain similarity

(GOODFELLOW et al., 2014; RADFORD; METZ; CHINTALA, 2015). The input images are initially full disk magnetograms (hmi.M_720s) from the HMI/SDO from May 2010 to November 2019. Once downloaded, they collect the positions of the ARs between ±45° of the central meridian through the SolarMonitor[1] cropping in 512 × 512 pixels images.

The authors followed the same strategy as Zheng, Xuebao Li, and Xinshuo Wang (2019) to label the images. The dataset was also randomly subsampled for "Non-flare" and "C-class" samples. They generated six distinct training and testing sets, but the authors did not explain how they divided data between training and testing.

The CNNs proposed by Deng et al. (2021) were inspired by the VGG network (SIMONYAN; ZISSERMAN, 2015) and AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, G. E., 2017). They proposed three distinct sub-models. The results of the M1 model for predicting ≥M-class flares within the 24 h forecasting horizon were ACC = 0.89, POD = 0.89, HSS = 0.76, and TSS = 0.77, not to mention FAR and CSI when considering the entire period of the dataset.

3.2.3 Works Considering Magnetograms Images and Sequences

Tang, Liao, et al. (2021) innovated by merging different deep learning forecasting methods, aggregating a DNN, a bidirectional Long Short Term Memory (bi-LSTM) network, and a CNN for predicting ≥C-class and ≥M-class flares within 24 h and 48 h forecasting horizons. Their dataset contains 59,706 samples from HMI/SDO from May 2010 to December 2015, where each sample is composed of ten numerical SHARP parameters and the respective cropped magnetogram images. The input to the DNN is the feature-extracted magnetogram numerical parameters, the input to the CNN is the magnetogram image, and the input to the bi-LSTM is a time series with all numerical parameters from time t to time $t + n$ of each sample.

The fusion model takes the outputs of the DNN, the CNN, and the bi-LSTM and passes the data into a fully connected layer followed by an activation function. In preliminary tests, the authors divided the dataset chronologically. They used the first 40,000 samples to train the models, the validation set was between sample 40,001 and sample 50,000, and the last 10,000 samples formed the test set. Preliminary tests showed that DNN (F1 = 0.54) was the best individual model, followed by bi-LSTM (F1 = 0.45) and CNN (F1 = 0.41). The fusion model improved the results, increasing the F1 Score to 0.57.

[1] Available at `https://www.solarmonitor.org/index.php`.

After preliminary testing, Tang, Liao, et al. (2021) optimized two versions of each model: one for the TSS and another for the F1. In the final tests, the authors reorganized the dataset with five-fold cross-validation tests to compare results. Despite cross-validation, they ensured that data from the same AR were not replied to between training and test sets. The fusion model optimized for TSS achieved ACC = 0.82, POD = 0.82, and TSS = 0.64 for forecasting ≥C-class flares, and ACC = 0.84, POD = 0.88 and TSS = 0.72, for ≥M-class flares within the 24 h forecasting horizon. The 48 h forecasting horizon results were ACC = 0.80, POD = 0.84, and TSS = 0.62 for ≥C-class flares and ACC = 0.89, POD = 0.83, and TSS = 0.72 for ≥M-class flares. The results did not mention FAR, CSI, and HSS metrics.

Guastavino et al. (2022) proposed a model that takes video data with AR-cropped magnetogram image sequences as input to predict ≥M-class flares within 24 h forecasting horizon. The authors build a hybrid deep-learning model named Long-term Recurrent Convolutional Network (LRCN). It works by applying a particular CNN to each sequence image and using each CNN output to build a time series to feed the LSTM.

The authors used HMI/SHARP between 2012 September 14 and 2017 September 30 to generate videos lasting 24 h, composed of 40 SHARP-sequenced images of AR-cropped magnetograms captured at intervals of 36 minutes. Before making the videos, they resized each image to 128 × 128 pixels. They built a CNN with four convolutional layers, each followed by a max-pooling and a ReLU function. At the end, there is a dense layer with 64 units, where they apply the dropout.

Guastavino et al. (2022) pointed out that they treated the input videos as time series and applied CNN concurrently to each video frame. Then, the outputs of the CNNs were sequentially processed over time and fed into the LSTM, which has 50 units, applying dropout with a fraction of 0.5. In the end, there is a dense sigmoid function that performs binary classification. They reached TSS = 0.55 for predicting ≥C-class flares and TSS = 0.68 for predicting ≥M-class flares within 24 h forecasting horizon.

Pengchao Sun et al. (2022) introduced a solar flare forecasting system with 3D CNNs. Using 3D convolutional layers, they can take magnetogram sequences as inputs to learn spatiotemporal correlations for forecasting solar flares without the necessity for combining multiple predictive models. They used SHARP AR-cropped magnetogram images between May 2010 and December 2019 located in the ±45° central meridian. As 3D CNNs require an image sequence of $T \times H \times W$, they resized the images to 128 × 128 pixels and created magnetogram sequences with 15 frames. Subsequently, the authors randomly divided the dataset into five subsets to perform the 5-fold cross-validation method, taking four subsets as training data and considering one subset for testing data.

The 3D CNN model had five 3D convolutional layers, followed by 3D max-pooling layers, one adaptive average-pooling layer, and three fully connected layers. They also added a batch normalization layer and a ReLU activation function after each 3D convolutional layer. The results were ACC = 0.90, POD = 0.93, FAR = 0.10, HSS = 0.67, and TSS = 0.83 for predicting ≥M-class flares within the 24 h forecasting horizon. They did not mention the CSI metric.

Most recently, Kaneda et al. (2023) presented the "Flare Transformer" for predicting ≥M-class flares within the 24 h forecasting horizon. The authors considered hourly line-of-sight magnetograms from the SDO[2] online physical feature database from Nishizuka, Sugiura, Kubo, Den, and Ishii (2018)[3] from June 2010 to December 2017. They split the training and test sets using time series 4-fold cross-validation, creating four distinct training and testing set pairs with the following periods:

- The first pair comprised training data from 2010 to 2013 and testing with 2014 data.
- In the second pair, the training data was from 2010 to 2014, and the testing set was from 2015 data.
- The third pair comprises training data from 2010 to 2015 and testing with 2016 data.
- In the fourth pair, training data was from 2010 to 2016, and testing was from 2017.

The "Flare Transformer" consisted of two integrated transformer-based modules: the Sunspot Feature Module (SFM), which takes time series with numerical parameters as input, and the Magnetogram Module (MM), which accepts time series with image data from magnetograms. Before the MM transformer layer, they applied an image feature extractor that consisted of multiple convolutional, max-pooling, average pooling, and batch normalization layers. In the SFM, physical features are the query. After this process, the transformer layers calculate the temporal relationships between time series magnetograms and physical features.

They reached TSS = 0.53, surpassing Nishizuka, Sugiura, Kubo, Den, and Ishii (2018) with the same dataset arrangement and human predictions. The authors also considered the

[2] Available at `https://sdo.gsfc.nasa.gov/data`.

[3] Available at `https://wdc.nict.go.jp/IONO/wdc/solarflare/index.html`.

Gandin–Murphy–Gerrity Score (GMGS) and Brier Skill Score (BSS) but did not present the confusion matrix or any other metric like ACC, POD, FAR, CSI, and HSS. It is worth noting that this dataset arrangement is more challenging than other works compared here, especially in pairs with less data training, since the transformer learning process requires larger datasets than CNNs to perform efficiently.

Tables 3.2 and 3.3 summarize the work results in this section by the 24 h and 48 h Forecasting Horizons (FH). Direct comparison is still tricky because of the different sources, dividing schema, and periods used in the datasets of each work.

Table 3.2: Binary ≥C-class flares forecasting tasks with deep learning.

Authors	FH	Class	ACC	POD	FAR	CSI	HSS	TSS
Guastavino et al. (2022)[⋄]	24 h	≥ C	—	—	—	—	—	0.55
Tang, Liao, et al. (2021)[×]	24 h	≥ C	0.82	0.82	—	—	—	0.64
Ali K Abed, Qahwaji, and Ahmed Abed (2021)[§]	24 h	≥ C	0.80	0.90	0.21	0.72	0.84	**0.89**
Xuebao Li et al. (2020)[†]	24 h	≥ C	0.86	0.89	0.09	—	0.67	0.68
Park et al. (2018)[‡]	24 h	≥ C	0.82	0.85	0.17	0.73	0.63	0.63
Xin Huang, Huaning Wang, Xu, Jinfu Liu, et al. (2018)[⊙]	24 h	≥ C	0.75	0.73	0.65	0.31	0.34	0.49
Nishizuka, Sugiura, Kubo, Den, and Ishii (2018)[⊗]	24 h	≥ C	0.82	0.81	0.47	0.47	0.53	0.63
Tang, Liao, et al. (2021)[×]	48 h	≥ C	0.80	0.84	—	—	—	**0.62**
Xin Huang, Huaning Wang, Xu, Jinfu Liu, et al. (2018)[⊙]	48 h	≥ C	0.80	0.67	0.46	0.43	0.47	0.50

[⋄] Results from Table 2 and Figure 5 of Guastavino et al. (2022) considering C1+ flares forecasting mean TSS including all samples. The authors did not mention confusion matrices or other metrics besides the TSS.

[×] Results from Table 9 of Tang, Liao, et al. (2021) considering the TSS_FFM(≥ C) model. The authors did not mention the FAR, CSI, and HSS.

[§] Results from Table 6 of Ali K Abed, Qahwaji, and Ahmed Abed (2021).

[†] Results from Table 2 of Xuebao Li et al. (2020). The authors did not mention the CSI.

[‡] Results referring to Model 3 of Table 3 of Park et al. (2018).

[⊙] Values computed according to Table 4 of Xin Huang, Huaning Wang, Xu, Jinfu Liu, et al. (2018).

[⊗] Results from Table 3 from Nishizuka, Sugiura, Kubo, Den, and Ishii (2018) considering the DeFN model for predicting ≥ C-class flares.

Table 3.3: Binary ≥M-class flares forecasting tasks with deep learning.

Authors	FH	Class	ACC	POD	FAR	CSI	HSS	TSS
Kaneda et al. (2023)§	24 h	≥ M	—	—	—	—	—	0.53
Pengchao Sun et al. (2022)‡	24 h	≥ M	0.90	0.93	0.10	—	0.67	**0.83**
Guastavino et al. (2022)⋄	24 h	≥ M	—	—	—	—	—	0.68
Tang, Liao, et al. (2021)×	24 h	≥ M	0.84	0.88	—	—	—	0.72
Deng et al. (2021)*	24 h	≥ M	0.88	0.89	—	—	0.76	0.77
Zheng, Xuebao Li, and Xinshuo Wang (2019)†	24 h	≥ M	0.89	0.82	0.11	—	0.76	0.75
Xin Huang, Huaning Wang, Xu, Jinfu Liu, et al. (2018)⊙	24 h	≥M	0.81	0.85	0.90	0.10	0.14	0.66
Nishizuka, Sugiura, Kubo, Den, and Ishii (2018)⊗	24 h	≥M	0.86	0.95	0.82	0.18	0.26	0.80
Tang, Liao, et al. (2021)×	48 h	≥ M	0.89	0.83	—	—	—	**0.72**
Xin Huang, Huaning Wang, Xu, Jinfu Liu, et al. (2018)⊙	48 h	≥ M	0.81	0.81	0.84	0.14	0.20	0.62

§ Results from Table 4 of Kaneda et al. (2023) considering the mean TSS. The authors did not mention confusion matrices or other metrics we compared but considered the Gandin–Murphy–Gerrity Score (GMGS) and Brier Skill Score (BSS).

‡ Results from Table 5 of Pengchao Sun et al. (2022). The authors did not mention the CSI score.

⋄ Results from Table 2 and Figure 5 from Guastavino et al. (2022) of M1+ flares forecasting considering the mean TSS including all samples. The authors did not mention confusion matrices or other metrics besides the TSS.

× Results from Table 9 of Tang, Liao, et al. (2021) considering the TSS_FFM(≥ M) model. The authors did not mention the FAR, CSI, and HSS.

* Results from Table 8 of Deng et al. (2021) considering the column "≥ M-class (M_1)". The authors did not inform the FAR and CSI score.

† Results from Table 5 of Zheng, Xuebao Li, and Xinshuo Wang (2019). The authors did not mention the CSI score.

⊙ Values computed according to Table 4 of Xin Huang, Huaning Wang, Xu, Jinfu Liu, et al. (2018).

⊗ Results from Table 3 from Nishizuka, Sugiura, Kubo, Den, and Ishii (2018) considering the DeFN model for predicting ≥ M-class flares.

Observing tables 3.2 and 3.3, we notice that the results for forecasting ≥M-class flares, which cause the most impacts on the Earth's atmosphere, are generally lower than those presented for forecasting ≥C-class flares. Although Nishizuka, Sugiura, Kubo, Den, and Ishii (2018) presented the best TSS for this case (0.80), on the other hand, they presented an equally lousy result for the FAR (0.82). It is also worth mentioning that most works focus on the 24 h FH, with a lack of works exploring FH greater than or equal to 48 h.

Except for Park et al. (2018), which used full-disk magnetogram images as inputs, and Ali K Abed, Qahwaji, and Ahmed Abed (2021), who jointly used an AR detector, the other works in this section used ready-made AR-cropped magnetograms as inputs for training and testing the models. The AR-cropped magnetograms are mainly obtained through SHARP patches (BOBRA; SUN, X., et al., 2014), in which the ARs are extracted using traditional image processing techniques, with a cadence of 720 seconds (12 minutes), while full-disk magnetograms are processed every 45 seconds (SCHOU et al., 2012).

3.3 Concluding Remarks

This chapter covered some precursor research. It raised the main works that used deep learning focused on image processing for solar flare forecasting. Works using deep-learning techniques began to emerge in 2018 with the application of simple CNN models and continue evolving in complexity and performance.

Before 2018, the works related to solar flares forecasting usually adopted numeric parameters as vector magnetic fields (BOBRA; COUVIDAT, 2015) or even time-series X-flux measuring (CALDANA et al., 2017) as inputs. However, it demands manual pre-processing or feature-extracting parameters. It does not also take the implicit parameters of line-of-sight magnetogram images about its related magnetic arcs.

From 2019 onward, works adopting line-of-sight magnetogram images began exploring new methods for processing the dataset, generating synthetic data to mitigate class imbalance, adding AR detection systems, combining different classification models, and considering magnetogram image sequences. In general, the advances obtained have improved the performance metrics and stability of the models developed.

It is also important to mention that most of the studies in our survey approach the solar flare forecasting problem as a binary classification task, predicting ≥C-class or ≥M-class flares within a 24 h forecasting horizon. However, since C-class flares generally do not cause substantial damage on Earth‘s ground systems and in its near atmosphere, and given the lack of research exploring the 48 h forecasting horizon, we decided to conduct our efforts on predicting ≥M-class flares within both 24 h and 48 h forecasting horizons.

Despite the predominance of CNN-based models, transformer-based solutions have recently been proposed for image-processing tasks and computer vision. Kaneda et al. (2023) were the first to apply transformer-based solutions for solar flare forecasting in this context. They developed the "Flare Transformer", built specifically for solar flare forecasting. This transformer had a separate image feature extractor and received numerical data together.

Considering all the related works we researched, we saw an opportunity to explore further the adoption of transformer-based predictive models that receive magnetogram image sequences as inputs for solar flare forecasting. In conjunction with the larger learning capacity of transformer-based models, the AR evolution implicit in magnetogram-sequenced inputs can provide crucial parameters for improving the performance for predicting M-class flares within the 24 h and 48 h forecasting horizons.

We also noted a lack of questions concerning adopting data augmentation techniques and their impacts on the forecasting models' performances. Despite Xuebao Li et al. (2020) and Deng et al. (2021) adopting oversampling and subsampling to solar flare forecasting they also modified the test set, which may have created biased results. Moreover, they did not evaluate the performance of the data-augmentation techniques themselves.

Considering these remarks, Chapter 4 presents the methodology employed in this thesis.

Chapter 4

Methodology

This chapter describes our methodology, starting with Section 4.1, which presents the base dataset we adopted and the modifications we applied to build our dataset. Section 4.2 discusses how we created our SHARP magnetogram image sequences dataset. Next, Section 4.3 presents our dataset splitting scheme. Section 4.4 shows the execution environment concerning the software and hardware resources. Section 4.5 describes Torchvision's MViTv2_s loaded model, how we adapt it, and the configurations of our models to receive magnetogram sequences. Section 4.6 discusses the oversampling techniques employed.

4.1 The SOLAR-STORM2 dataset

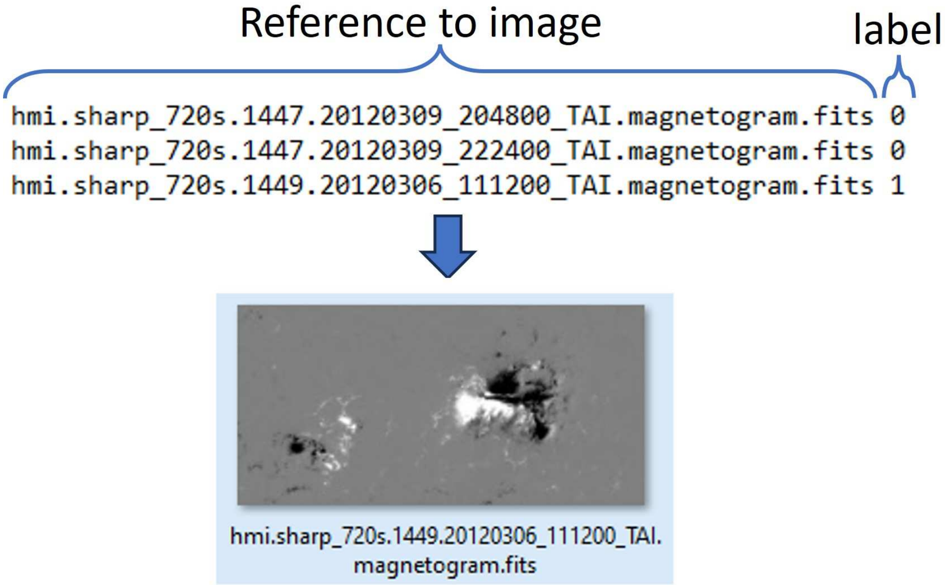
To ensure the comparability of our results, we adopted the SOLAR-STORM2 dataset provided by the "Space Environment Warning and AI Technology Interdisciplinary Innovation Working Group" (TIANCHI, 2020; HUANG, X.; WANG, H.; XU; LIU, J., et al., 2018) to build our magnetogram image sequences. It refers to the same dataset adopted by Tang, Liao, et al. (2021)[1], which we downloaded from their provided link.

The SOLAR-STORM2 dataset comprises 73,810 high-quality Space-weather Helioseismic and Magnetic Imager Active Region Patches (SHARP) (BOBRA; SUN, X., et al., 2014) samples captured by HMI/SDO from May 4th, 2010, to January 26th, 2019. The SHARP data from HMI/SDO are initially stored in the "`hmi.sharp_720s`" dataset, with new samples available every 12 min. However, in the "SOLAR-STORM2" dataset, the sample collection was downsampled every 1 h 36 min.

[1] Available at `https://tianchi-competition.oss-cn-hangzhou.aliyuncs.com/531804/dataset_ss2sff.zip`

Despite most authors adopted limitations in their line-of-sight magnetogram captures, like considering only ARs located within ±30° or ±45° of the Sun's central meridian to avoid the influence of projection effects, the documentation of the SOLAR-STORM2 dataset (TIANCHI, 2020) did not mention any limitation on their captured SHARP's patches. However, as SOLAR-STORM2 is associated with Xin Huang, Huaning Wang, Xu, Jinfu Liu, et al. (2018), it is crucial to note that they considered only ARs located within ±30° of the Sun's central meridian in their work.

Each sample originally included ten SHARP magnetic numeric attributes, a line-of-sight magnetogram image referenced, and their respective label. However, we considered only the line-of-sight SHARP magnetogram images and their respective labels, discarding the numeric attributes. Remembering that the SHARP magnetograms contain one or more ARs as classified by the NOAA, Figure 4.1 illustrates some data samples, part of the SOLAR-STORM2 dataset catalog, in the format we considered, highlighting a referenced image of the SHARP ID 1449, which includes NOAA AR 11429 and NOAA AR 11430.



```
hmi.sharp_720s.1447.20120309_204800_TAI.magnetogram.fits 0
hmi.sharp_720s.1447.20120309_222400_TAI.magnetogram.fits 0
hmi.sharp_720s.1449.20120306_111200_TAI.magnetogram.fits 1
```

Figure 4.1: Data we used from SOLAR-STORM2 dataset. Source: Grim and Gradvohl (2024).

The SOLAR-STORM2's magnetogram images are grayscale (single-channel), and each line-of-sight magnetogram image has a distinct resolution, varying from 50 × 33 to 1985 × 1898 pixels according to the crop made by SHARP. The positive label (1) in Figure 4.1 denotes that the sample produced an ≥M-class flare within the next 48 h (48 h forecasting horizon) from the moment we captured them. Conversely, the negative label (0) implies that the sample did not produce a ≥M-class flare within the same period. The Geostationary

Operational Environmental Satellite (GOES) classified these flares based on the observed soft X-ray flux peak value (TANG; LIAO, et al., 2021).

For our research, we adopted the period pointed out by Tang, Liao, et al. (2021) to enable a direct comparison with their work. This period includes samples from May 4th, 2010, to December 30th, 2015, and also excludes 322 samples with missing data, resulting in 59,384 valid samples. Additionally, we re-labeled the classes to create the 24 h forecasting horizon based on the data provided by GOES-15[2], linking each NOAA-AR with their respective SHARP-ID.

It is worth noting that the GOES-15 X-ray flux data passed through a recalibration process in 2022 (MACHOL et al., 2022), which probably increased the number of M- and X-class flares during the Solar Cycle 24. However, as the SOLAR-STORM2 dataset was published in 2020, it did not consider this recalibration in its original labeling. Thus, we relabeled our 24 h forecasting horizon like Tang, Liao, et al. (2021) without considering the recalibration process of GOES-15 X-ray flux data.

Table 4.1 compares the valid positive samples, negative samples, and the Rate of Positive Samples (RoPS) for the 24 h and 48 h forecasting horizons between the original SOLAR-STORM2's period and the period we adopted. Despite some differences, both datasets were highly imbalanced, with the RoPSs below 5%.

Table 4.1: Summary of different periods in the SOLAR-STORM2 dataset.

Period	Forecasting horizon	Positive samples	Negative samples	RoPS
From May 4th, 2010 to Jan. 26th, 2019⋄	48 h	2,988	70,822	4.05%
From May 4th, 2010 to Dec. 30th, 2015⋆	48 h	2,837	56,547	4.78%
	24 h	1,520	57,864	2.56%

⋄ Original period, as we obtained from the source.
⋆ The period we adopted is the same considered by Tang, Liao, et al. (2021).

The RoPS for the 48 h forecasting horizon in the original period is slightly lower (4.05%) than in the period we adopted (4.78%). This discrepancy occurs because the period from May 2010 to December 2015 comprises only the solar activity rise of Solar Cycle 24 from the beginning to the peak period.

[2] Available at `https://www.ngdc.noaa.gov/stp/space-weather/solar-data/solar-features/solar-flares/x-rays/goes/xrs/`

In contrast, as shown in Table 4.1, the original period spans nearly the entire Solar Cycle 24, including the years of solar activity decline (2016, 2017, and 2018). In highly imbalanced datasets, even minor differences in the RoPS can affect some performance metrics (discussed in Appendix A). Therefore, we chose the same period specified by Tang, Liao, et al. (2021) to facilitate a more accurate comparison.

4.2 Building SHARP Magnetogram Image Sequences

Considering each sample of the SOLAR-STORM2 dataset, we adopted the sliding window method to create the SHARP magnetogram image sequences. It considers taking sequenced observations from an AR, discarding the most outdated, and including a newly updated sample when comparing neighboring sequences.

This approach mirrors the technique employed by Tang, Liao, et al. (2021) for the SHARP numerical attributes. However, our research focused on image sequences, approaching the method adopted by Pengchao Sun et al. (2022). Thus, we created a visual representation of ARs evolution of each SHARP-ID, as described in Equation 4.1.

$$\mathrm{Seq}_T = [T + 0, T + 1, T + 2, T + 3, \dots, T + \Delta t - 1], \tag{4.1}$$

where $T + 0$ is the initial image; Δt is the window size; and Seq_T is the complete sequence starting in $T + 0$, with size Δt, and ending at the current sample image $T + \Delta t - 1$.

Yu et al. (2009) stated that the emergence process of solar flares lasts between 72 h and 96 h. Given the forecasting horizon of 48 h (2 days), it is acceptable to consider sliding windows that include images from 72 h (3 days) earlier up to the current time $T + \Delta t - 1$. With the SOLAR-STORM2 dataset having a 96 min ($\approx$ 1.5 h) cadence, the maximum Δt would be 45 images. However, in our experiments, we limited the maximum Δt to 16 images (approximately one day earlier) due to the limitations of the transformers models we adopted in our experiments.

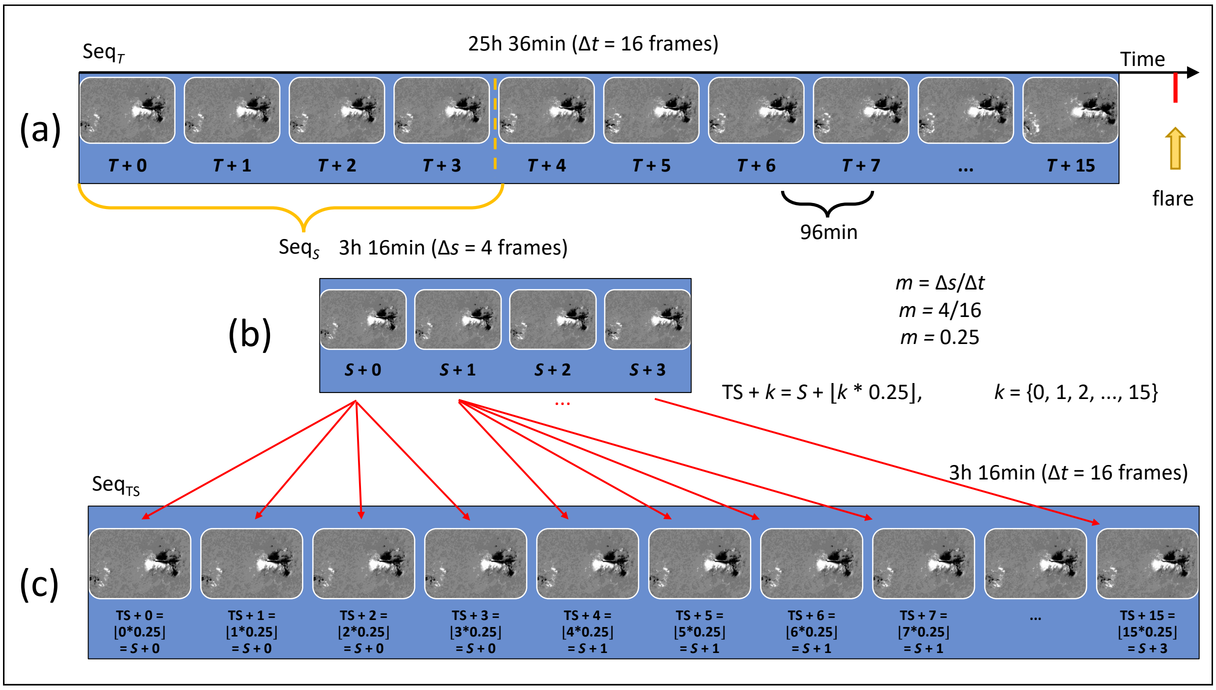
At the start of monitoring each SHARP-ID, sequences (Seq_S) may lack enough images (Δs) to fill out all Δt instants. Generally, deep-learning models require all inputs to have the exact dimensions, including the number of images inside the sequence. Yu et al. (2009) propose creating artificial sequences ($\mathrm{Seq}_{\mathrm{TS}}$) by repeating the available samples to solve this situation. We applied this strategy by repeating the available observations proportionally, as outlined in equations 4.2, 4.3, 4.4, and 4.5, illustrated in Figure 4.2.

$$\text{Seq}_S = [S+0, S+1, S+2, \dots, S+\Delta s-1], \tag{4.2}$$

$$m = \frac{\Delta s}{\Delta t}, \tag{4.3}$$

$$\text{TS}+k = S + \lfloor k \times m \rfloor, \; k = \{0, 1, 2, \dots, \Delta t-1\}, \tag{4.4}$$

$$\text{Seq}_{\text{TS}} = [\text{TS}+0, \text{TS}+1, \text{TS}+2, \text{TS}+3, \dots, \text{TS}+\Delta t-1]. \tag{4.5}$$



Seq$_T$
25h 36min (Δt = 16 frames)
Time
(a)
T + 0
T + 1
T + 2
T + 3
T + 4
T + 5
T + 6
T + 7
...
T + 15
flare
96min
Seq$_S$ 3h 16min (Δs = 4 frames)
(b)
S + 0
S + 1
S + 2
S + 3
m = Δs/Δt
m = 4/16
m = 0.25
TS + k = S + ⌊k * 0.25⌋,
k = {0, 1, 2, ..., 15}
...
Seq$_{TS}$
3h 16min (Δt = 16 frames)
(c)
TS + 0 = ⌊0*0.25⌋ = S + 0
TS + 1 = ⌊1*0.25⌋ = S + 0
TS + 2 = ⌊2*0.25⌋ = S + 0
TS + 3 = ⌊3*0.25⌋ = S + 0
TS + 4 = ⌊4*0.25⌋ = S + 1
TS + 5 = ⌊5*0.25⌋ = S + 1
TS + 6 = ⌊6*0.25⌋ = S + 1
TS + 7 = ⌊7*0.25⌋ = S + 1
...
TS + 15 = ⌊15*0.25⌋ = S + 3

Figure 4.2: Artificial Seq$_{\text{TS}}$ creation from incomplete Seq$_S$. Source: Grim and Gradvohl (2024).

Figure 4.2 provides an example using the SHARP-ID 1449 initial observations. Figure 4.2-(a) presents the complete sequence Seq$_T$ with $\Delta t = 16$ images, corresponding to the observation at instant $T + 15$. Figure 4.2-(b) depicts an incomplete sequence Seq$_S$ with $\Delta s = 4$ images, referring to the observation at instant $S + 3$, equivalent to $T + 3$ of Seq$_T$.

Figure 4.2-(c) demonstrates the artificial sequence Seq$_{\text{TS}}$ creation, derived from Seq$_S$. This approach allows us to consider all samples of the SOLAR-STORM2 dataset, including the initial observations of SHARP-IDs with fewer than Δt images.

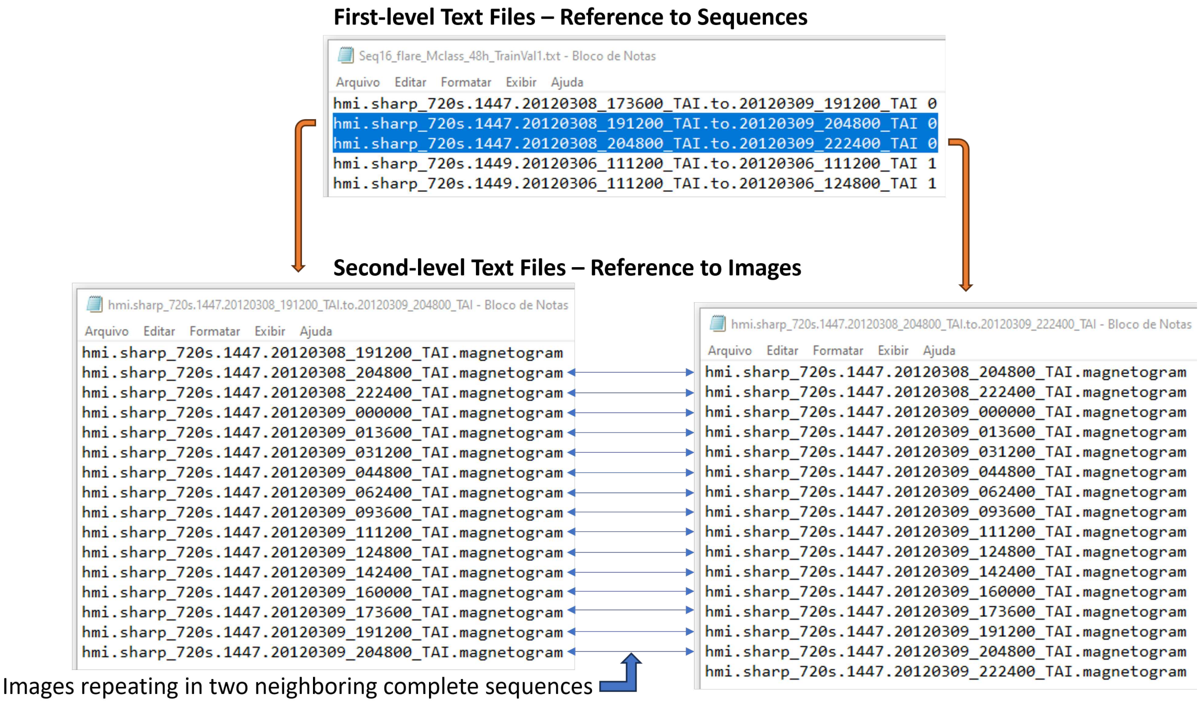
4.2.1 Text Files for Storing SHARP Sequences

As many of the images in the SOLAR-STORM2 dataset become repetitive in close sequences, we chose to store the SHARP Magnetogram Image Sequences in referenced plain text files (extension `.txt`) instead of creating a video itself in a classic format.

When adopting plain text files to reference the sequences, we saved considerable space storing them. We also turned the sequences loading process into the prediction model faster than using classic video formats. We organized our SHARP Sequences dataset into two levels:

- In the first level, we have the reference to the sequence and its label. Each referenced sequence in the first-level files points to a second-level plain text file. We named the sequences into the first-level plain text files with the pattern `hmi.sharp_720s.<SHARP-ID>.<init-date>.to.<end-date>`, where:
  - `<SHARP-ID>`: the SHARP region containing one or more NOAA AR.
  - `<init-date>`: date-time when the sequence starts, following the mask: yyyymmdd_hhnnss_TAI (where y: year, m: month, d: day, h: hours, n: minutes, s: seconds).
  - `<end-date>`: date-time sequence ends, with the same format as `<init-date>`.
- In the second level, the references point to each image belonging to the sequence. The images came named from the base-dataset with the pattern `hmi.sharp720s.<SHARP-ID>.<date>.magnetogram.fits.jpg`, where `<date>` is the date-time when the instrument captured the image and follows the same format of `<init-date>` and `<end-date>`.

Figure 4.3 shows an example regarding the last two SHARP ID 1447 sequences. As it shows, we repeated fifteen of sixteen images in two neighboring complete sequences of the same AR. This pattern repeats throughout the dataset. We would considerably increase the dataset size if we generated traditional video files for each sequence. We also chose to organize it in text files to ease the reproducibility of the dataset for other researchers and interested parties, regardless of the technology and systems they use.



First-level Text Files – Reference to Sequences

Seq16_flare_Mclass_48h_TrainVal1.txt - Bloco de Notas

```
hmi.sharp_720s.1447.20120308_173600_TAI.to.20120309_191200_TAI 0
hmi.sharp_720s.1447.20120308_191200_TAI.to.20120309_204800_TAI 0
hmi.sharp_720s.1447.20120308_204800_TAI.to.20120309_222400_TAI 0
hmi.sharp_720s.1449.20120306_111200_TAI.to.20120306_111200_TAI 1
hmi.sharp_720s.1449.20120306_111200_TAI.to.20120306_124800_TAI 1
```

Second-level Text Files – Reference to Images

hmi.sharp_720s.1447.20120308_191200_TAI.to.20120309_204800_TAI - Bloco de Notas

```
hmi.sharp_720s.1447.20120308_191200_TAI.magnetogram
hmi.sharp_720s.1447.20120308_204800_TAI.magnetogram
hmi.sharp_720s.1447.20120308_222400_TAI.magnetogram
hmi.sharp_720s.1447.20120309_000000_TAI.magnetogram
hmi.sharp_720s.1447.20120309_013600_TAI.magnetogram
hmi.sharp_720s.1447.20120309_031200_TAI.magnetogram
hmi.sharp_720s.1447.20120309_044800_TAI.magnetogram
hmi.sharp_720s.1447.20120309_062400_TAI.magnetogram
hmi.sharp_720s.1447.20120309_093600_TAI.magnetogram
hmi.sharp_720s.1447.20120309_111200_TAI.magnetogram
hmi.sharp_720s.1447.20120309_124800_TAI.magnetogram
hmi.sharp_720s.1447.20120309_142400_TAI.magnetogram
hmi.sharp_720s.1447.20120309_160000_TAI.magnetogram
hmi.sharp_720s.1447.20120309_173600_TAI.magnetogram
hmi.sharp_720s.1447.20120309_191200_TAI.magnetogram
hmi.sharp_720s.1447.20120309_204800_TAI.magnetogram
```

hmi.sharp_720s.1447.20120308_204800_TAI.to.20120309_222400_TAI - Bloco de Notas

```
hmi.sharp_720s.1447.20120308_204800_TAI.magnetogram
hmi.sharp_720s.1447.20120308_222400_TAI.magnetogram
hmi.sharp_720s.1447.20120309_000000_TAI.magnetogram
hmi.sharp_720s.1447.20120309_013600_TAI.magnetogram
hmi.sharp_720s.1447.20120309_031200_TAI.magnetogram
hmi.sharp_720s.1447.20120309_044800_TAI.magnetogram
hmi.sharp_720s.1447.20120309_062400_TAI.magnetogram
hmi.sharp_720s.1447.20120309_093600_TAI.magnetogram
hmi.sharp_720s.1447.20120309_111200_TAI.magnetogram
hmi.sharp_720s.1447.20120309_124800_TAI.magnetogram
hmi.sharp_720s.1447.20120309_142400_TAI.magnetogram
hmi.sharp_720s.1447.20120309_160000_TAI.magnetogram
hmi.sharp_720s.1447.20120309_173600_TAI.magnetogram
hmi.sharp_720s.1447.20120309_191200_TAI.magnetogram
hmi.sharp_720s.1447.20120309_204800_TAI.magnetogram
hmi.sharp_720s.1447.20120309_222400_TAI.magnetogram
```

Images repeating in two neighboring complete sequences

Figure 4.3: Organization of text files to reference the sequences.

4.3 Dataset Splitting

Most studies in our literature survey adopted the chronological order to split the original datasets. In contrast, some researchers used k-fold cross-validation to evaluate the models' stability. However, as Tang, Liao, et al. (2021) highlighted, chronologically ordered datasets are more challenging than randomly shuffled datasets for solar flare forecasting (MURANUSHI et al., 2015; NISHIZUKA; SUGIURA; KUBO; DEN; ISHII, 2018).

Therefore, we adopted a new proposal based on a hybrid dataset split to evaluate the models' stability and preserve the data's chronological order. We used the first 50,000 samples to perform 5-fold cross-validation, training with 40,000 samples and validating with 10,000 samples. The test set remains at the end with the last 9,384 samples.

Figure 4.4 shows the hybrid split we adopted. The training and validation sets act as known data, while the test set remains unknown. This method allows us to assess the model's performance and stability by evaluating the means and the standard deviations between all trained splits. It is essential to highlight that the Active Regions of the training/validation set do not overlap the test set, and their periods do not overlap.

Table 4.2 shows the RoPS of each fold, the test data, and the entire dataset. Folds 3 and 4 show a discrepancy compared with the other folds, the Test data, and the Entire dataset RoPS.

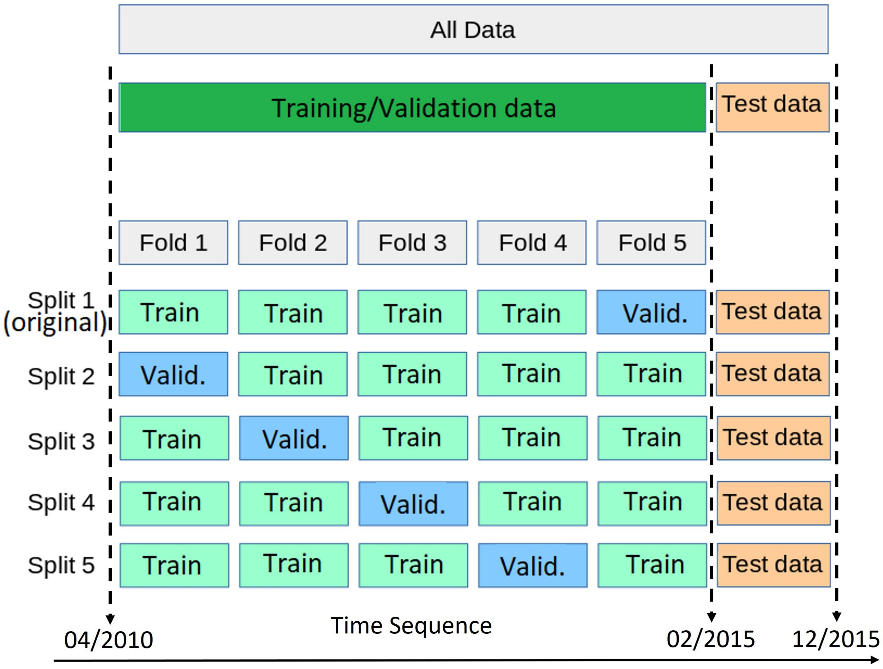


Figure 4.4: Dataset splitting. Source: Grim and Gradvohl (2024).

Table 4.2: RoPS of each fold and test data for M24 and M48 datasets.

Forecasting horizon	Training and Validation sets Fold-1	 Fold-2	 Fold-3	 Fold-4	 Fold-5	Test data	Entire dataset
24 h	2.3%	2.5%	0.9%	5.2%	2.6%	2.1%	2.6%
48 h	4.1%	4.5%	2.7%	8.8%	5.5%	4.4%	4.8%

While Fold-1, -2, -3, and Test data presented RoPS ≈ 2.4% for the 24 h forecasting horizon, Fold-3 presented RoPS = 0.9%, and Fold-4 presented RoPS = 5.2%. In the 48 h forecasting horizon scenario, Fold-1, -2, -3, and Test data presented RoPS ≈ 4.6%, Fold-3 presented RoPS = 2.7%, and Fold-4 presented RoPS = 8.8%.

It shows that each Split of Training/Validation data will present specific challenges due to the RoPS variation in both scenarios, especially Split 4 and Split 5, which will validate with Fold-3 and Fold-4, respectively. Besides being in chronological order, we can also consider the Test data suitable from the RoPS point of view since it is closer to the Entire dataset's RoPS.

Our SHARP dataset, including the images, the magnetogram sequences, and the model developed (described in Section 4.5), is available in our repository (GRIM; GRADVOHL, 2023b).

4.4 Execution environment

This section describes the execution environment in which we ran our experiments, concerning the respective software and hardware resources. We adopted specialized software tools and robust hardware with GPU support.

4.4.1 Software Resources

We built our forecasting models using PyTorch combined with PyTorch Lightning. The scientific community and professionals widely adopt PyTorch and PyTorch Lightning as default frameworks for training deep-learning models.

PyTorch[3] is a deep-learning framework that supports GPU acceleration and automatic backpropagation. PyTorch Lightning[4] simplifies the code by eliminating the explicit “for” loop, a characteristic when training deep-learning models with PyTorch, making the code more organized.

Although installing CUDA[5] and using a GPU is optional in PyTorch, training our models on a CPU would be unfeasible due to the size of the dataset and the transformer-based models we intended to adopt. Therefore, we relied on a high-performance computing environment equipped with GPU-based servers, as described in Section 4.4.2.

4.4.2 Hardware Resources

We trained and tested our models in the high-performance computing Lovelace environment at the Centro Nacional de Processamento de Alto Desempenho em São Paulo (CENAPAD-SP)[6]. The Lovelace computational environment provides five Dell EMC R7525 GPU-based computing nodes. Each node has two NVIDIA Tesla A100 GPU[7] with 40GB HBM2 RAM, two 64-core AMD Epyc 7662 CPUs, and 512 GB RAM.

However, as this is a shared environment, we cannot access all resources available by the computing nodes. We were limited to the resources made available by the queuing system, adapting our models to run on the “umagpu” and “duasgpus” queues via job submission. These queues provide the following features:

[3] Available at https://pytorch.org.

[4] Available at https://www.pytorchlightning.ai.

[5] Available at https://developer.nvidia.com/cuda-toolkit.

[6] Available at https://www.cenapad.unicamp.br.

[7] Available at https://www.nvidia.com/pt-br/data-center/a100.

- The queue “umagpu” provides one computing node with:
  - One NVIDIA Tesla A100 GPU with 40 GB RAM;
  - 16-core on AMD Epyc 7662 CPU;
  - 60 GB RAM;
- The queue “duasgpus” provides one computing node with:
  - Two NVIDIA Tesla A100 GPU, totaling 80 GB RAM;
  - 32-core on AMD Epyc 7662 CPU;
  - 120 GB RAM;

Although we adapted our models to these two queues, we ran most experiments on the “umagpu” queue. Despite our models running faster with two GPUs, the “duasgpus” queue was much more congested, with their jobs taking twice as long or more time to start execution than in the “umagpu” queue.

4.5 Implementation and Configuration of SF_MViTs

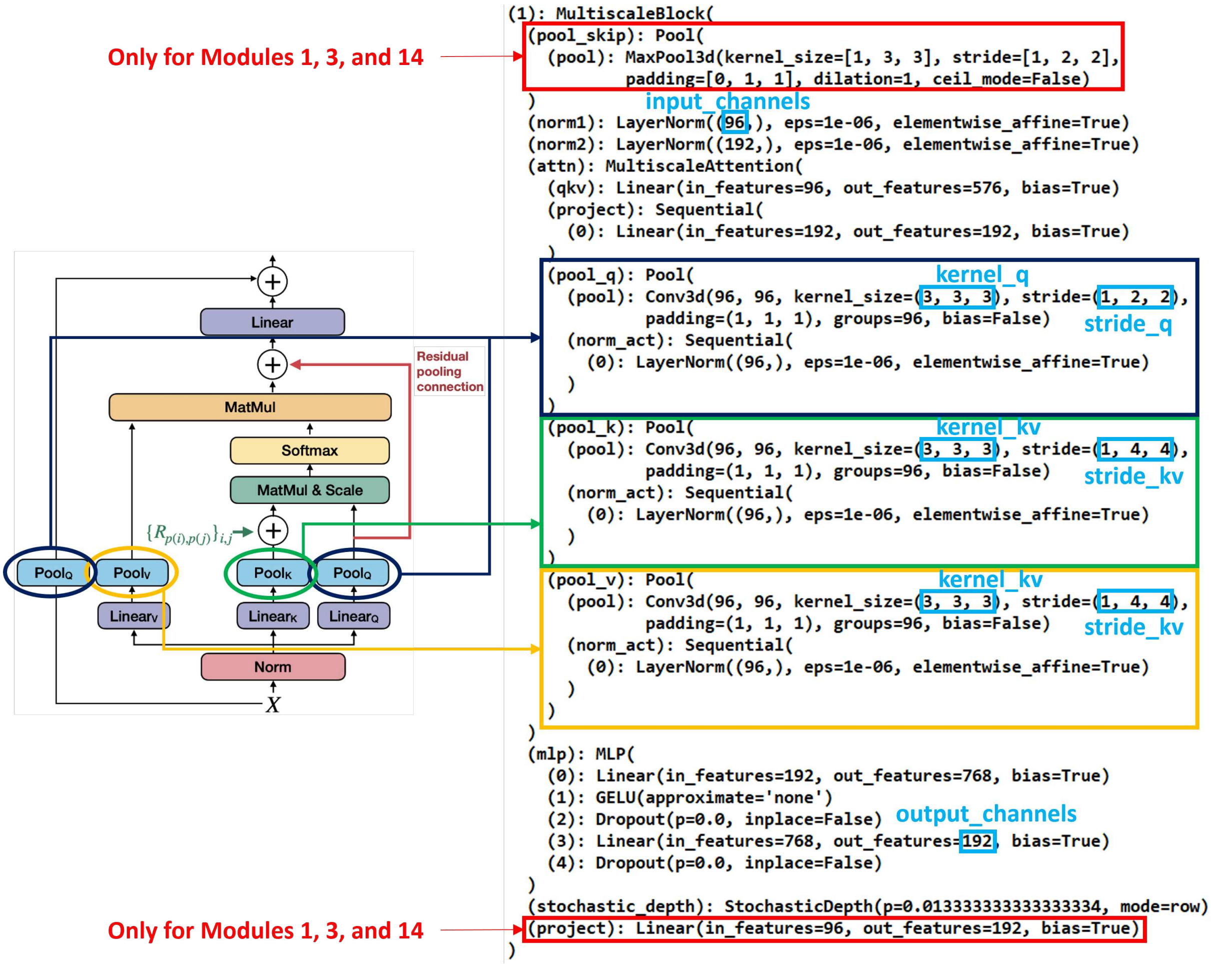
When we started our research, we first considered employing hybrid models (CNN + LSTM) or 3D CNNs for solar flare forecasting, taking magnetogram image sequences as inputs. In the middle of 2022, Guastavino et al. (2022) published a hybrid model joining CNN outputs to a bi-LSTM, achieving a TSS = 0.68. Therefore, we directed our efforts to 3D CNN models.

We even published preliminary results with the ResNet 3D, ResNet MC, and ResNet2D+1 models compared to some traditional 2D CNNs (GRIM; GRADVOHL, 2022) considering the SOLAR-STORM2 dataset from May 4th, 2010 to Dec. 30th, 2015 period. However, our results in this work could have performed better (Max. TSS = 0.48 for ResNet2D+1). Additionally, in late 2022, Pengchao Sun et al. (2022) published a work that applied 3D CNNs to solar flare forecasting. Considering our previous results with 3D CNNs and the innovation criterion, we changed our methodology and adopted transformer-based video models.

We build our SF_MViT models based on the MViTv2_s model loaded from PyTorch’s Torchvision library (Yanghao Li et al. (2022) MViT-S model) with Kinetics-400 weights (KAY et al., 2017). Despite being a small version of MViTv1_b applying the decomposed relative position embedding and the residual pool (see equations 2.9, 2.10, 2.11), it can still be considered a large model compared with the other models mentioned in Chapter 3.

4.5.1 Torchvision's Video MViTv2_s Loaded Structure

Torchvision's Video MViTv2_s model comprises sixteen stacked "Multiscale Block" modules. Figure 4.5 shows the structure of these modules and some details of their related stacked layers, emphasizing the $\mathcal{P}_Q$ (`pool_q`), $\mathcal{P}_K$ (`pool_k`), and $\mathcal{P}_V$ (`pool_v`) layers.



```
(1): MultiscaleBlock(
  (pool_skip): Pool(
    (pool): MaxPool3d(kernel_size=[1, 3, 3], stride=[1, 2, 2],
            padding=[0, 1, 1], dilation=1, ceil_mode=False)
  )
  (norm1): LayerNorm((96,), eps=1e-06, elementwise_affine=True)
  (norm2): LayerNorm((192,), eps=1e-06, elementwise_affine=True)
  (attn): MultiscaleAttention(
    (qkv): Linear(in_features=96, out_features=576, bias=True)
    (project): Sequential(
      (0): Linear(in_features=192, out_features=192, bias=True)
    )
    (pool_q): Pool(
      (pool): Conv3d(96, 96, kernel_size=(3, 3, 3), stride=(1, 2, 2),
              padding=(1, 1, 1), groups=96, bias=False)
      (norm_act): Sequential(
        (0): LayerNorm((96,), eps=1e-06, elementwise_affine=True)
      )
    )
    (pool_k): Pool(
      (pool): Conv3d(96, 96, kernel_size=(3, 3, 3), stride=(1, 4, 4),
              padding=(1, 1, 1), groups=96, bias=False)
      (norm_act): Sequential(
        (0): LayerNorm((96,), eps=1e-06, elementwise_affine=True)
      )
    )
    (pool_v): Pool(
      (pool): Conv3d(96, 96, kernel_size=(3, 3, 3), stride=(1, 4, 4),
              padding=(1, 1, 1), groups=96, bias=False)
      (norm_act): Sequential(
        (0): LayerNorm((96,), eps=1e-06, elementwise_affine=True)
      )
    )
  )
  (mlp): MLP(
    (0): Linear(in_features=192, out_features=768, bias=True)
    (1): GELU(approximate='none')
    (2): Dropout(p=0.0, inplace=False)
    (3): Linear(in_features=768, out_features=192, bias=True)
    (4): Dropout(p=0.0, inplace=False)
  )
  (stochastic_depth): StochasticDepth(p=0.013333333333333334, mode=row)
  (project): Linear(in_features=96, out_features=192, bias=True)
)
```

Figure 4.5: MViTv2_s Multiscale Block Modules. Adapted from Yanghao Li et al. (2022).

Figure 4.5 shows that $\mathcal{P}_Q$, $\mathcal{P}_K$, and $\mathcal{P}_V$ layers (respectively highlighted in dark blue, green, and yellow) have a 3D Convolutional layer instantiated by default (LI, Y. et al., 2022). They are inside the self-attention (`attn`) layer, followed by a Multi-layer Perceptron (`MLP`) that has a Gaussian Error Linear Units (GELU) non-linear function.

Equation 4.6 describes the GELU function, which is the most used in transformer models and can be considered a smoothed ReLU.

$$\text{GELU}(z) = z \times \Phi(z), \tag{4.6}$$

where $\Phi(z)$ is the Cumulative Distribution Function for the Gaussian Distribution.

It is worth noting that Figure 4.5 presents the “Module 1” stacked layers with their respective parameters. However, as highlighted in red in Figure 4.5, the (`pool_skip`) layer at the module’s beginning and the (`project`) layer at the module’s ending are only valid for modules 1, 3, and 14. The other modules do not have these layers.

Each “Multiscale Block” module presents distinct “num_heads”, “input_channels”, and “output_channels” parameters. The 3D Convolutional layers inside of pooling attention $\mathcal{P}_Q$ (`pool_q`), $\mathcal{P}_K$ (`pool_k`), and $\mathcal{P}_V$ (`pool_v`) layers also present distinct “stride_q” and “stride_kv” parameters according to their module.

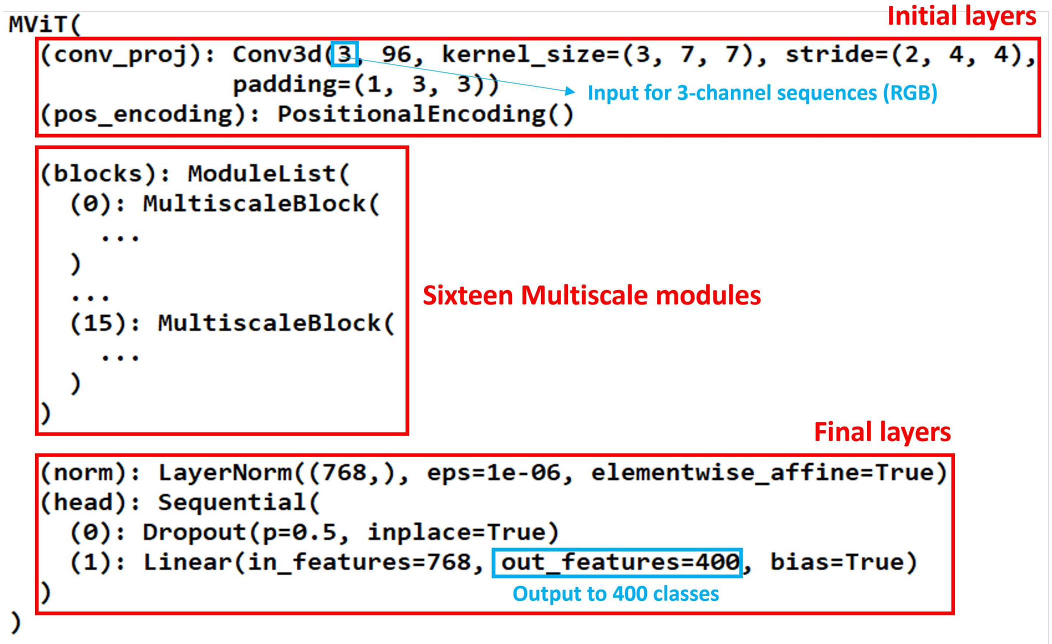
Table 4.3 presents the respective Module and 3D Convolutional layer (3D Conv.)’s parameter values configured on the initialization of the MViTv2_s model. Except for the “num_heads” parameter, which is inside the code and does not appear in the stacked layers, we highlighted the parameters addressed in Table 4.3 in light blue in Figure 4.5.

Table 4.3: Torchvision’s Video MViTv2_s loaded parameters.

Module	Module’s parameters			3D Conv.’s parameters	
	num_heads	input_channels	output_channels	stride_q	stride_kv
0	1	96	96	[1, 1, 1]	[1, 8, 8]
1	2	96	192	[1, 2, 2]	[1, 4, 4]
2	2	192	192	[1, 1, 1]	[1, 2, 2]
3	4	192	384	[1, 2, 2]	[1, 2, 2]
4 ... 13	4	384	384	[1, 1, 1]	[1, 2, 2]
14	8	384	768	[1, 2, 2]	[1, 1, 1]
15	8	768	768	[1, 1, 1]	[1, 1, 1]

We configured the parameters “kernel_q = [3, 3, 3]” and “kernel_kv = [3, 3, 3]” – also highlighted in light blue in Figure 4.5 – on the MViTv2_s initialization with these same values for all modules. The other layers’ parameters not highlighted in Figure 4.5 may vary according to the “num_heads” and “input_channels” or still have the same values between the modules.

Besides the “Multiscale Blocks” represented in Figure 4.5, the MViTv2_s loaded from Torchvision presents a 3D Conv. (`conv_proj`) and a Positional Encoding (`pos_encoding`) layer before the Module List (`blocks`). After the Module List, there is a Normalization layer (`norm`) and the final Output layer (`head`), including a Dropout (`head[0]`) and a fully connected Linear (`head[1]`) layer. Figure 4.6 shows MViTv2_s initial and final layers.



```
MViT(
  (conv_proj): Conv3d(3, 96, kernel_size=(3, 7, 7), stride=(2, 4, 4),
                padding=(1, 3, 3))
  (pos_encoding): PositionalEncoding()
  (blocks): ModuleList(
    (0): MultiscaleBlock(
      ...
    )
    ...
    (15): MultiscaleBlock(
      ...
    )
  )
  (norm): LayerNorm((768,), eps=1e-06, elementwise_affine=True)
  (head): Sequential(
    (0): Dropout(p=0.5, inplace=True)
    (1): Linear(in_features=768, out_features=400, bias=True)
  )
)
```

Initial layers
Input for 3-channel sequences (RGB)
Sixteen Multiscale modules
Final layers
Output to 400 classes

Figure 4.6: MViTv2_s initial and final layers.

After loading Torchvision's MViTv2_s pre-trained with Kinetics-400 weights and its standard parameters, we needed to make some adaptations in the model to work correctly on our binary classification problem for predicting ≥M-class flares within 24 h and 48 h forecasting horizons. Next, Section 4.5.2 presents these issues.

4.5.2 SF_MViTs Adaptations and Configurations

As Figure 4.6 depicts, the Torchvision MViTv2_s model's first layer is a 3D Convolution configured to receive a tensor with three color channels ($D = 3$), conceived to consider RGB videos or sequences. However, the SOLAR-STORM2 dataset comprises grayscale image sequences with a single color channel ($D = 1$).

To tackle this problem, we conducted some tests by reshaping the grayscale images into three channels or modifying the first `conv_proj` layer parameter to accept single-channel input. We found better results and faster training by reshaping the images to three channels ($D = 3$) instead of modifying the model's original input configuration. This situation occurred probably because the loaded MViTv2_s model's PyTorch code was constructed optimized for 3-channel training. Thus, we adopted this approach.

Furthermore, we had to modify the final fully connected layer. Thus, the `head[1]` layer configuration becomes `Linear (in_features=768, out_features=2, bias=True)`, with two output units. The output units of the last fully connected layer represent the number of output classes. This configuration allows us to perform fine-tuning training (i. e., transfer learning) for our binary classification even in models pre-trained with more output classes.

We employed the PyTorch Lightning deep-learning framework to streamline our code, eliminating the training's main "for" explicit loop and automating specific hyperparameters. We still adopted the legacy hooks from the `LightningModule` class, which work correctly on the previous versions "0.x" to "1.x" since we started developing our codes on PyTorch Lightning "1.2". In our code, we adopt the following hooks:

- `training_step()`: to load the training data batches, make the model's forward pass, compute the batch training loss, compute the actual mean training loss, and return the two losses computed.
- `training_epoch_end()`: to compute the mean training loss at the epoch's end and return it.
- `validation_step()`: to load the validation data batches, make the model's forward pass, compute the batch validation loss and accuracy, and return the computed values.
- `validation_epoch_end()`: to compute the average epoch validation loss and accuracy and store them in the log.
- `test_step()`: to load the test data batches, make the model's forward pass, compute the batch testing loss and accuracy, and return the testing batch loss and accuracy.
- `test_epoch_end()`: to compute the average epoch testing loss and accuracy and store them in the log.
- `configure_optimizers()`: to configure the loss optimizer, which updates the models' weights in the backward pass and the scheduler to change them.

We highlight that the loss backward pass is not present in any hook because it is executed implicitly at the end of the `training_step` hook. PyTorch Lightning also helps us find the initial Learning Rate (LR) to implement the LR monitor, the training checkpoint control, and the early stop method through their callbacks. Table 4.4 outlines our initial hyperparameter settings concerning the model's training.

Table 4.4: Hyperparameters' initial configuration.

Hyperparameter	Value
Input dimensions (L)	$T \times H \times W$
Images in sequence (T)	16 images
Images' height (H)	224 pixels
Images' width (W)	224 pixels
Initial channel width (D)	3 channels
Batch size	10 samples
Maximum Epochs	100 epochs
Initial Learning Rate	10^{-7}
Learning Rate decay	10^{-1}
Min. Learning Rate	10^{-10}
Optimizer	Adam
Cost function	Weighted Cross-Entropy Loss
Scheduler	Reduce LR on Plateau
Scheduler Patience	2 epochs
Early Stopping	By Val_Loss
Early Stopping Patience	10 epochs

The inputs consisted of batches of ten samples, each with $T = 16$ sequenced images, reshaped to three channels ($D = 3$), resized to 224×224 pixels, and normalized from 0 to 1. We evaluated the PyTorch Lightning Auto LR Finder to suggest the initial LR. Most results presented values between 10^{-7} and 8×10^{-7}. However, in some tests, the Auto LR Finder produced values outside this range, occasionally returning an error.

Thus, we set the initial LR to 10^{-7}, applying the scheduler known as "Reduce LR on Plateau", with a decay factor of 10^{-1} if the Validation Loss (Val_Loss) did not improve after two epochs (Scheduler Patience). Additionally, we implemented the early stopping method to halt training if the Val_Loss did not improve after ten epochs (Early Stopping patience).

To address the issue of highly imbalanced classes, we implemented the Weighted Cross-Entropy Loss (Wloss) cost function, following the approach of Nishizuka, Sugiura, Kubo, Den, and Ishii (2018) and Xuebao Li et al. (2020), as specified in Equation 4.7.

$$\text{Wloss} = \sum_{c=0}^{C-1} \omega_c \times y_c \times \log(\hat{y}_c), \tag{4.7}$$

where C represents the number of classes, ω_c is the weight related to each class, y_c indicates the expected output, and $\hat{y}_c$ represents the predicted output.

We calculated the weights ω_c according to the inverse proportion of class occurrences, considering the training and validation sets separately. Thus, we had distinct cost functions for the training and validation steps, each with its class weight. Additionally, we trained oversampled models to address the class imbalance, as detailed in Section 4.6.

4.6 Oversampling SHARP Magnetogram Sequences

Table 4.1 shows that the SOLAR-STORM2 dataset's RoPS in the period we adopted was 2.56% for ≥M-class flares within the 24 h forecasting horizon and 4.78% within the 48 h forecasting horizon. These low percentages may hinder learning from positive samples, even when fine-tuning combined with a Wloss cost function is adopted since class imbalance significantly impacts the training of classification models (JAPKOWICZ; STEPHEN, 2002).

A common approach to addressing the class imbalance in deep-learning models is oversampling the minority-class samples through data augmentations. Image data augmentation can include techniques such as adding new samples with rotation, flipping, random cuts, scale and zoom, color changes, noise, image mixing, and synthetic data generation. Due to the particularities of our dataset, we do not consider the following data augmentation techniques:

- Random cuts: It can cut off important parts of the Active Regions in the sample;
- Scale and zoom: Some samples present more than one Active Region, which can be located near the image border;
- Color changes: Our dataset is originally single-channel grayscale images;
- Noise / Image mixing: Considering we are adopting image sequences, we believe that the same pattern of noise and mixing would be applied to the sequence;
- Synthetic data generation: It can be prejudicial to the learning process.

To apply noise, image mixing, and synthetic data generation to the magnetogram sequences, we would need to build functions requiring a considerable complex degree in our code to be not prejudicial to the learning process. Thus, we consider implementing only sequences flipping and rotations for the minority class samples, in our case, positive samples that represent magnetogram sequences that generate one or more solar flares ≥ M-class within the next 24 h or 48 h.

Image flipping and rotation already adopted by Xuebao Li et al. (2020) and Deng et al. (2021), together with the downsampling of the majority-class samples (no-flaring samples or related to <M-class flares). However, downsampling majority-class samples can negatively affect learning from those examples. Therefore, we oversampled the sequences of positive samples exclusively, as illustrated in Figure 4.7.

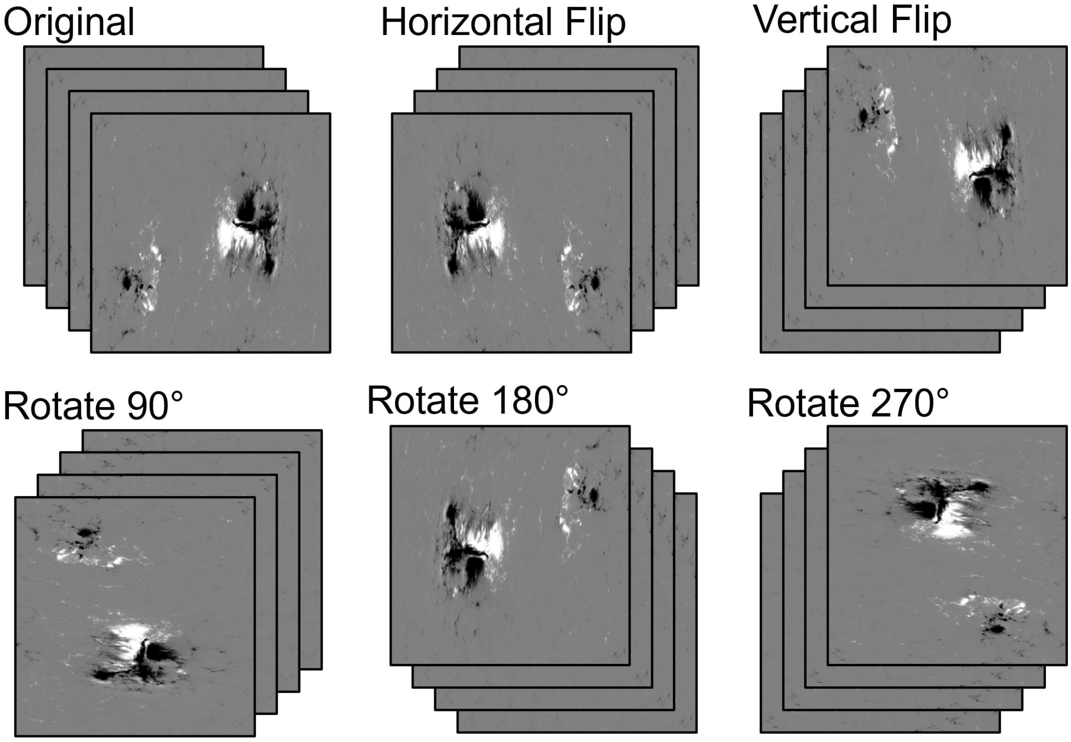


Figure 4.7: Oversampling ≥M-class flares sequences. Source: Grim and Gradvohl (2024).

Adopting this procedure, we created two artificial samples by applying vertical and horizontal flipping, and three artificial samples by rotating the magnetogram sequences by 90°, 180°, and 270°, totaling five artificial samples for each ≥M-class flare positive sample. This process increased the number of these samples by six times.

Table 4.5 presents the RoPS in each training/validation fold for the "M24" dataset, which corresponds to ≥M-class flares within the 24 h forecasting horizon, and the "M48" dataset, representing the ≥M-class flares within the 48 h forecasting horizon. Table 4.5 also presents the RoPS of their respective oversampled versions: the "M24_Over" and "M48_Over" datasets.

Table 4.5: RoPS of each fold for M24, M48, M24_Over, and M48_Over datasets.

	Training and Validation sets				
	Fold-1	Fold-2	Fold-3	Fold-4	Fold-5
M24 dataset	2.3%	2.5%	0.9%	5.2%	2.6%
M24_Over dataset	13.9%	15.2%	5.7%	31.4%	15.9%
M48 dataset	4.1%	4.5%	2.7%	8.8%	5.5%
M48_Over dataset	24.7%	26.9%	16.5%	52.8%	33.1%

To apply the oversample techniques in our models, we first called a function to create an inflated first-level dataset plain text file (see Section 4.2.1). For each positive sample (label = 1), we created five new samples with the same sequence reference and gave each one a specific flag. The original samples receive the flag “nf”, while the created samples receive the flags “hf”, “vf”, “hf”, “r1”, “r2”, and “r3”.

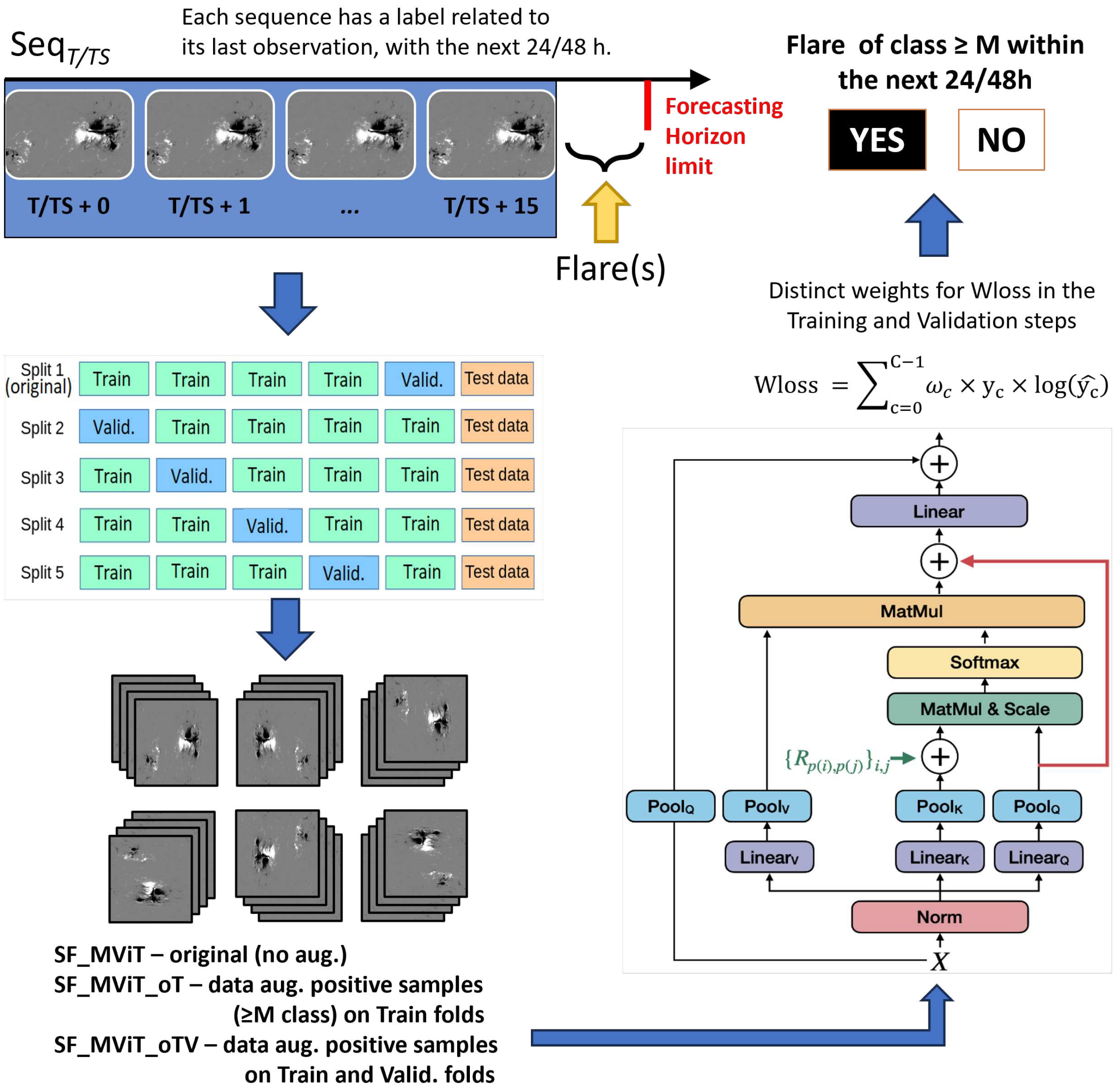
Then, we created our `Mflare_Dataset` class, based on PyTorch’s `Dataset` class, in which we apply the respective transformations in the flagged sequences as follows: horizontal flip for the “hf” flagged, vertical flip for the “vf” flagged, 90° rotation for the “r1” flagged, 180° rotation for the “r2” flagged, and 270° rotation for the “r3” flagged.

4.7 Methodology Overview

Based on this methodology, we developed three distinct models implementing oversampling in different parts throughout the dataset to evaluate the impact of the oversampling techniques adopted. We performed a full fine-tuning training in all models’ layers. First, we trained the Solar Flare MViT (SF_MViT) model solely with the original sequences, without oversampling.

Next, we trained two oversampled models: the Solar Flare MViT over Train (SF_MViT_oT), applying oversampling only on the training set, and the Solar Flare MViT over Train and Validation (SF_MViT_oTV), employing oversampling on the training and validation sets. The test set remains without modifications.

Figure 4.8 shows an overview of our methodology from building our magnetogram sequences to training our three proposed models.


$\mathrm{Seq}_{T/TS}$
Each sequence has a label related to its last observation, with the next 24/48 h.
T/TS + 0
T/TS + 1
...
T/TS + 15
Forecasting Horizon limit
Flare(s)
Flare of class ≥ M within the next 24/48h
YES
NO
Distinct weights for Wloss in the Training and Validation steps
SF_MViT – original (no aug.)
SF_MViT_oT – data aug. positive samples (≥M class) on Train folds
SF_MViT_oTV – data aug. positive samples on Train and Valid. folds
Linear
MatMul
Softmax
MatMul & Scale
$\{R_{p(i),p(j)}\}_{i,j}$
Pool_Q
Pool_V
Pool_K
Pool_Q
Linear_V
Linear_K
Linear_Q
Norm
X

Split 1 (original)	Train	Train	Train	Train	Valid.	Test data
Split 2	Valid.	Train	Train	Train	Train	Test data
Split 3	Train	Valid.	Train	Train	Train	Test data
Split 4	Train	Train	Valid.	Train	Train	Test data
Split 5	Train	Train	Train	Valid.	Train	Test data

$$\mathrm{Wloss} = \sum_{c=0}^{C-1} \omega_c \times y_c \times \log(\hat{y}_c)$$

Figure 4.8: Methodology overview.

4.8 Concluding Remarks

This chapter describes the methodology used to train our SF_MViTs models. First, we presented the base dataset we adopted (TIANCHI, 2020) and how we created the SHARP magnetogram sequences to act as input for video recognition models. We also showed the hybrid split we adopted to perform cross-validation in the training and validation set, isolating the test set.

Due to the size of the dataset and the transformer-based models we intended to develop, we needed to consider high-performance GPU-based computing servers and specialized software tools. Then, we presented the hardware and software resources we used to execute our experiments, including the CENAPAD-SP Lovelace computing environment and combining PyTorch with PyTorch Lightning software tools.

Following, we explained the development of our models. We used PyTorch to load Torchvision's MViTv2_s model trained with the Kinetics-400 dataset. Then, we modified the loaded model to receive magnetogram sequences as inputs (video data) to predict ≥M-class flares within the 24 h and 48 h forecasting horizons, employing fine-tuning training. We also presented the hyperparameter's configurations.

Our methodology innovated in applying fine-tuning training on a pre-trained transformer-based model for general purposes. Instead, most related works applied models built specifically for solar flare forecasting trained from scratch. From our knowledge, we are the first work to employ transformer-based models for solar flare forecasting that receive only SHARP magnetogram image sequences as inputs.

Although Kaneda et al. (2023) also developed a transformer-based approach for solar flare forecasting, they trained their models with magnetogram numerical parameters and image sequences. They also considered only the 24 h forecasting horizon, while we considered 24 h and 48 h forecasting horizons.

In addition, we developed three distinct models that applied data augmentation in different parts of the dataset, from the training to the validation sets, to deal with class imbalance. Therefore, we can evaluate the impact of the minority class oversampling in each model, attesting whether data augmentation can significantly improve their performance.

Next, Chapter 5 discusses the results obtained, including comparisons between our models and the correlated works.

Chapter 5

Results and Discussions

This chapter presents the results and discussions. Sections 5.1 and 5.2 address our models' training convergence for the M24 and M48 datasets, respectively. Section 5.3 presents our results for the test step, and Section 5.4 compares them with state-of-the-art works.

5.1 M24 Dataset Training Convergence

We tracked the models' convergence during training by monitoring the evolution of Validation Loss (Val_Loss) and Validation Accuracy (Val_ACC). Figures 5.1, 5.2, and 5.3 show the Val_Loss and the Val_ACC of the models training on the M24 dataset, recorded every five epochs. They show the individual results for each split and the means between all splits.

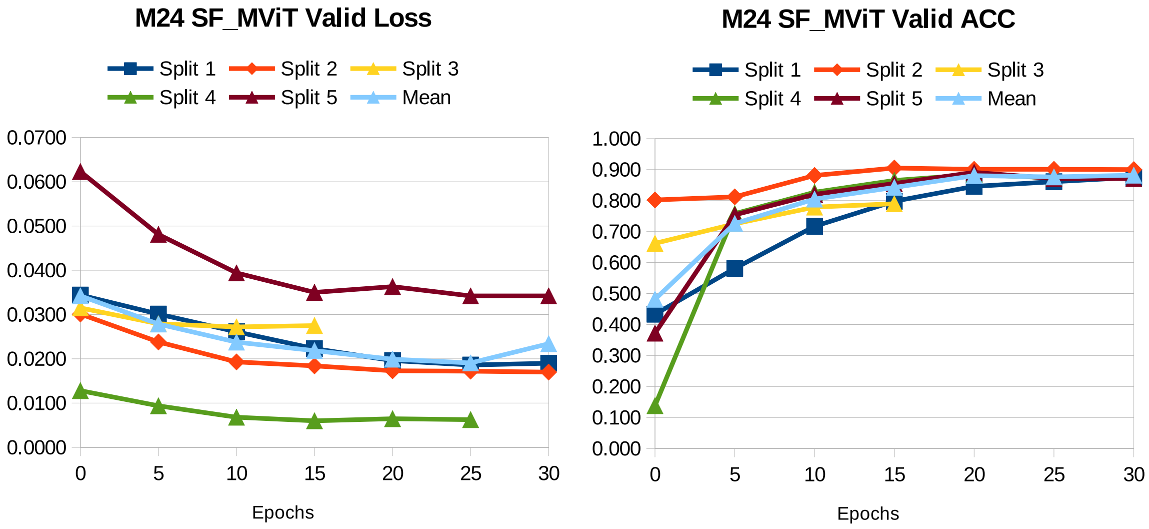


Figure 5.1: SF_MViT convergence in the M24 dataset. Source: Grim and Gradvohl (2024).

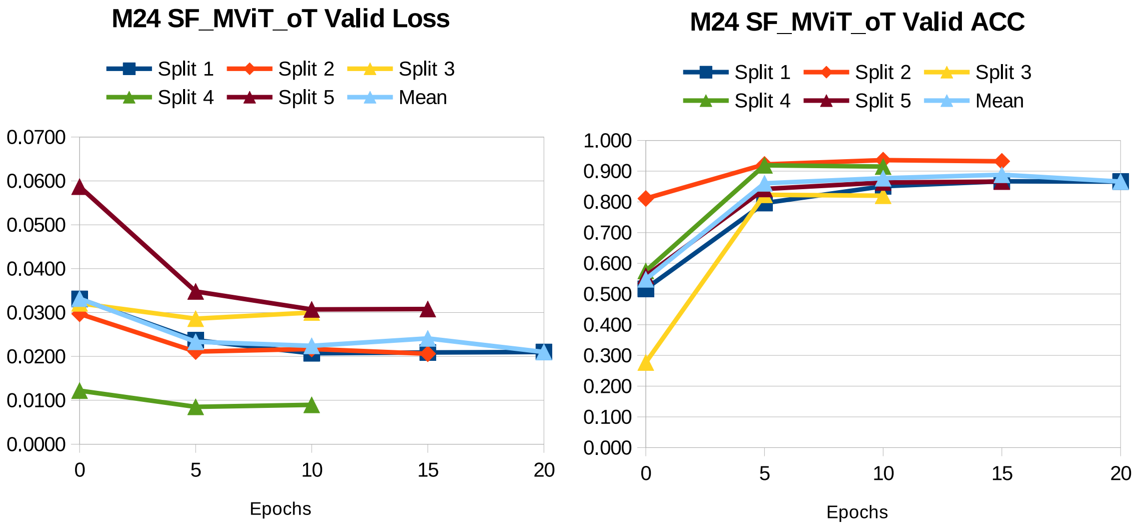


Figure 5.2: SF_MViT_oT convergence in the M24 dataset. Source: Grim and Gradvohl (2024).

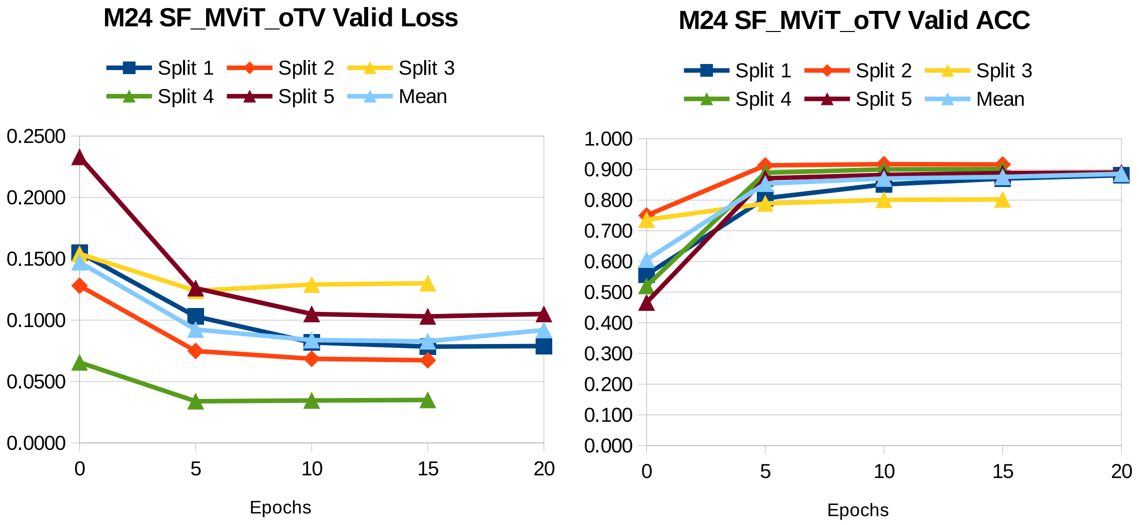


Figure 5.3: SF_MViT_oTV convergence in the M24 dataset. Source: Grim and Gradvohl (2024).

Analyzing figures 5.1, 5.2, and 5.3, we observed that the oversampled models (SF_MViT_oT and SF_MViT_oTV) achieved faster convergence than SF_MViT. Specifically, the Val_Loss and the Val_ACC for the oversampled models stabilized closer to epoch 10, whereas SF_MViT stabilized between epochs 15 and 25.

Moreover, Table 5.1 presents the Val_Loss and the Val_ACC for the best epoch, defined as the epoch with the lowest Val_Loss, from the models trained on the M24 dataset. It details the individual results for each Split, the means, and their respective Standard Deviation (SD) across the splits.

Table 5.1: Convergence status of the best epoch on the M24 dataset.

	SF_MViT model			SF_MViT_oT model			SF_MViT_oTV model		
Split	Best Epoch	Val_ Loss	Val_ ACC	Best Epoch	Val_ Loss	Val_ ACC	Best Epoch	Val_ Loss	Val_ ACC
1	25	0.019	0.86	10	0.021	0.85	16	0.078	0.88
2	30	0.017	0.90	8	0.019	0.93	9	0.067	0.92
3	9	0.027	0.78	3	0.027	0.80	5	0.124	0.79
4	17	0.006	0.87	4	0.008	0.90	6	0.033	0.89
5	29	0.034	0.87	9	0.031	0.84	14	0.103	0.89
Means	22	0.021	0.86	7	0.021	0.86	10	0.081	0.87
SD	±9	±0.011	±0.05	±3	±0.009	±0.05	±5	±0.035	±0.05

Table 5.1 clarified that oversampled models converge faster than SF_MViT. The mean Best Epoch for SF_MViT_oT was 7, with a corresponding mean Val_Loss = 0.021, while the mean Best Epoch for SF_MViT was 22, with the same mean Val_Loss (0.021) of SF_MViT.

We can directly compare the Val_Losses of SF_MViT and SF_MViT_oT because they have the same validation step's weights in their respective cost functions, as neither model applied oversampling in the validation step. Conversely, SF_MViT_oTV applies oversampling in the training and validation steps, resulting in different weights on its Val_Loss cost function.

Still, according to Table 5.1, the mean Best Epoch for SF_MViT_oTV was 10, with a higher Val_Loss = 0.081 but a similar Val_ACC to the other models. This discrepancy is due to the different weights on the SF_MViT_oTV model's Val_Loss.

5.1.1 M24 Dataset Training Times

Table 5.2 presents the models' Mean Epoch Training Time, Mean Total Epochs, Estimated Split Training Time, and Estimated Total Training Time, remembering that we trained each model in five splits. Due to our Early Stopping Method, all models continue training for ten epochs after the best epoch in our implementation.

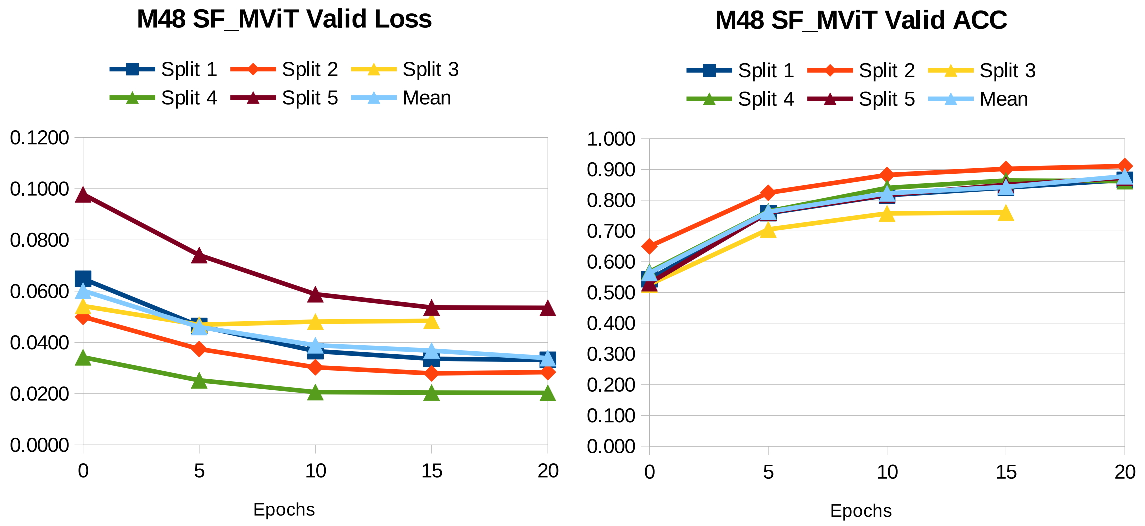
Despite this delay in stopping the training, the SF_MViT_oT and SF_MViT_oTV models are advantageous in Estimated Split Training Time and Total Training Time. Table 5.2 shows that, although SF_MViT_oT and SF_MViT_oTV converge faster, they take slightly longer per epoch due to the additional augmented samples.

Table 5.2: Estimated times to train the models in the M24 dataset.

	Mean Epoch Training Time	Mean Total Epochs	Estimated Split Training Time	Estimated Total Training Time
SF_MViT model	0:25:45	32	13:44:00	68:40:00
SF_MViT_oT model	0:32:24	17	9:10:48	45:54:00
SF_MViT_oTV model	0:30:08	20	10:02:40	50:13:20

5.2 M48 Dataset Training Convergence

Figures 5.4, 5.5, and 5.6 display the models' Val_Loss and the Val_ACC trained on the M48 dataset every five epochs, with individual results and the means between the splits.



M48 SF_MViT Valid Loss

M48 SF_MViT Valid ACC

Figure 5.4: SF_MViT convergence in the M48 dataset. Source: Grim and Gradvohl (2024).

According to figures 5.4, 5.5, and 5.6, the oversampled models (SF_MViT_oT and SF_MViT_oTV) converge faster than SF_MViT on the M48 dataset. However, the difference is less pronounced than on the M24 dataset. For the M48 dataset, SF_MViT_oT and SF_MViT_oTV stabilize their Val_Loss and Val_ACC closer to epoch 10, while SF_MViT stabilizes closer to epoch 15. Thus, SF_MViT stabilizes training on the M48 dataset more quickly than on the M24 dataset, where it stabilizes around epoch 20.

Table 5.3 presents Best Epoch's Val_Loss and Val_ACC from models trained on the M48 dataset, detailing each Split's results, the means, and the SD across the splits.

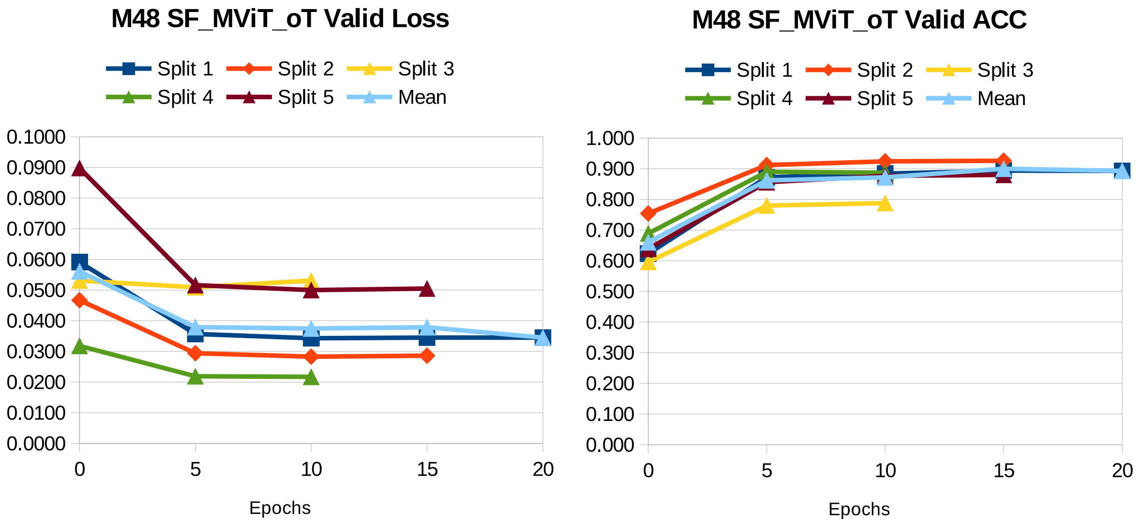


Figure 5.5: SF_MViT_oT convergence in the M48 dataset. Source: Grim and Gradvohl (2024).

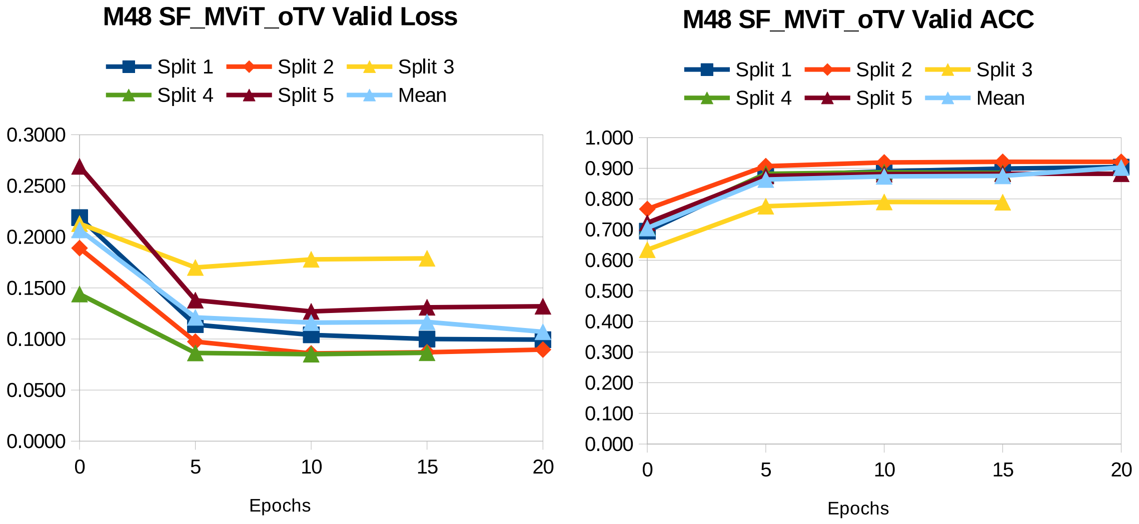


Figure 5.6: SF_MViT_oTV convergence in the M48 dataset. Source: Grim and Gradvohl (2024).

Table 5.3 showed that the mean Best Epoch for SF_MViT_oT was 7 for the M48 dataset, with a corresponding mean Val_Loss = 0.036, while for SF_MViT was 15 with the same mean Val_Loss of SF_MViT_oT. The mean Best Epoch for SF_MViT_oTV was 12, with a higher Val_Loss = 0.113. However, the Val_ACC for SF_MViT_oTV remains closer to the other models. As in the M24 dataset scenario, this behavior occurred due to the different weights adopted in the Val_Loss of the training step's cost function of the SF_MViT_oTV model.

Table 5.3: Convergence status of the best epoch in the M48 dataset.

	SF_MViT model			SF_MViT_oT model			SF_MViT_oTV model		
Split	Best Epoch	Val_ Loss	Val_ ACC	Best Epoch	Val_ Loss	Val_ ACC	Best Epoch	Val_ Loss	Val_ ACC
1	19	0.033	0.86	12	0.034	0.89	20	0.099	0.90
2	18	0.027	0.90	8	0.028	0.92	16	0.083	0.92
3	6	0.047	0.73	3	0.049	0.75	5	0.170	0.78
4	13	0.020	0.83	4	0.021	0.87	9	0.084	0.88
5	17	0.053	0.85	7	0.050	0.87	10	0.127	0.88
Means	15	0.036	0.84	7	0.036	0.86	12	0.113	0.87
SD	±5	±0.014	±0.06	±4	±0.013	±0.06	±6	±0.037	±0.06

5.2.1 M48 Dataset Training Times

Table 5.4 shows the Mean Epoch Training Time, the Estimated Split Training Time, and the Total Training for the models trained in the M48 dataset.

Table 5.4: Estimated times to train the models in the M48 dataset.

	Mean Epoch Training Time	Mean Total Epochs	Estimated Split Training Time	Estimated Total Training Time
SF_MViT model	00:32:28	25	13:31:40	67:38:20
SF_MViT_oT model	01:00:21	17	17:05:57	85:29:45
SF_MViT_oTV model	01:02:08	22	22:46:56	113:54:40

According to Table 5.4, despite converging faster, SF_MViT_oT and SF_MViT_oTV take longer to train an epoch than SF_MViT. It contrasts with the situation in Table 5.2, where the M48 dataset has a higher RoPS and more oversampled positive samples than the M24 dataset. Consequently, the oversampled models are not advantageous considering the estimated training times in the M48 dataset scenario compared to SF_MViT.

5.3 Results Obtained in the Test Step

We loaded the models' weights from the Best Epoch at the end of each split's training process. Then, we performed the test step and evaluated their performance. We highlight that the test set remains original without oversampling and is unknown to the models.

Most of the metrics that we adopt to evaluate performance in the test step are the same metrics that we already addressed in Chapter 3, such as Accuracy (ACC), Probability of Detection (POD), False-Alarm Ratio (FAR), Critical Success Index (CSI), Heidke Skill Score (HSS), and True Skill Statistic (TSS). We also add the Precision (PRE) and the F1 Score (F1) to better compare with correlated works. We explain all these metrics in Appendix A.

After executing all splits for each model, we compute their metrics' means and SDs. Tables 5.5, 5.6, and 5.7 present the individual results for each split, along with the means and SDs for the SF_MViT, SF_MViT_oT, and SF_MViT_oTV models within the 24 h forecasting horizon (FH). Supported by the consensus prevailing since Bobra and Couvidat (2015), our analysis considers the TSS as the primary evaluation metric.

For all tables in this chapter, the values in bold mean the best performing, while the underlined values mean the worst performing, always considered within the same metric. The Rate of Positive Samples (RoPS) on the M24 test set is 2.08%.

Table 5.5: SF_MViT test results for ≥M-class flares within the 24 h forecasting horizon.

SF_MViT results for M24 Test								
Split	F1	ACC	POD	PRE	FAR	CSI	HSS	TSS
1	0.24	0.89	0.80	0.14	0.86	0.14	0.21	0.70
2	0.22	0.89	0.77	0.13	0.87	0.13	0.20	0.67
3	0.19	0.85	0.82	0.10	0.90	0.10	0.15	0.67
4	0.25	0.90	0.85	0.15	0.85	0.14	0.22	**0.74**
5	0.24	0.90	0.76	0.15	0.85	0.14	0.22	0.67
Means	0.23	0.89	0.80	0.13	0.87	0.13	0.20	0.69
SD	±0.02	±0.02	±0.04	±0.02	±0.02	±0.02	±0.03	±0.03

Tables 5.5, 5.6, and 5.7 showed that our models reached TSS ≈ 0.70 in M24 test set. Recall that TSS values range from -1 to 1, with 1 being optimal, near zero indicating random hits and negative values showing no forecasting competence (HANSSEN; KUIPERS, 1965). The largest

Table 5.6: SF_MViT_oT test results for ≥M-class flares within the 24 h forecasting horizon.

SF_MViT_oT results for M24 Test								
Split	F1	ACC	POD	PRE	FAR	CSI	HSS	TSS
1	0.27	0.91	0.84	0.16	0.84	0.16	0.25	**0.75**
2	0.27	0.92	0.76	0.17	0.83	0.16	0.25	0.68
3	0.21	0.88	0.77	0.12	0.88	0.12	0.18	0.65
4	0.28	0.91	0.81	0.17	0.83	0.16	0.25	0.72
5	0.25	0.90	0.82	0.15	0.85	0.14	0.23	0.72
Means	0.26	0.90	0.80	0.15	0.85	0.15	0.23	0.70
SD	±0.03	±0.02	±0.03	±0.02	±0.02	±0.02	±0.03	±0.04

Table 5.7: SF_MViT_oTV test results for ≥M-class flares within the 24 h forecasting horizon.

SF_MViT_oTV results for M24 Test								
Split	F1	ACC	POD	PRE	FAR	CSI	HSS	TSS
1	0.27	0.92	0.73	0.17	0.83	0.16	0.25	0.66
2	0.29	0.92	0.79	0.18	0.82	0.17	0.27	0.72
3	0.25	0.90	0.82	0.15	0.85	0.15	0.23	0.72
4	0.28	0.91	0.84	0.17	0.83	0.16	0.25	**0.75**
5	0.28	0.92	0.75	0.18	0.82	0.17	0.26	0.67
Means	0.27	0.91	0.79	0.17	0.83	0.16	0.25	0.70
SD	±0.02	±0.01	±0.05	±0.01	±0.01	±0.01	±0.01	±0.04

SD was 0.05 for the SF_MViT_oTV's POD metric, indicating the models' stability. Therefore, we can consider our fine-tuned models valid approaches for solar flare forecasting.

Except for the POD metric, the oversampled models outperform the SF_MViT in most cases across their corresponding splits, reflecting directly in their means. Figure 5.7 illustrates the improvements made to the oversampled models compared to SF_MViT.

We highlight notable improvements for the SF_MViT_oTV comparing with the SF_MViT in the following metric means: HSS by +0.05, F1 and PRE by +0.04, CSI by +0.03, ACC by +0.02, and TSS by +0.01. Additionally, the SF_MViT_oTV decreases the FAR by −0.03, which is favorable since a lower FAR indicates fewer false positives, and zero is the optimal value.

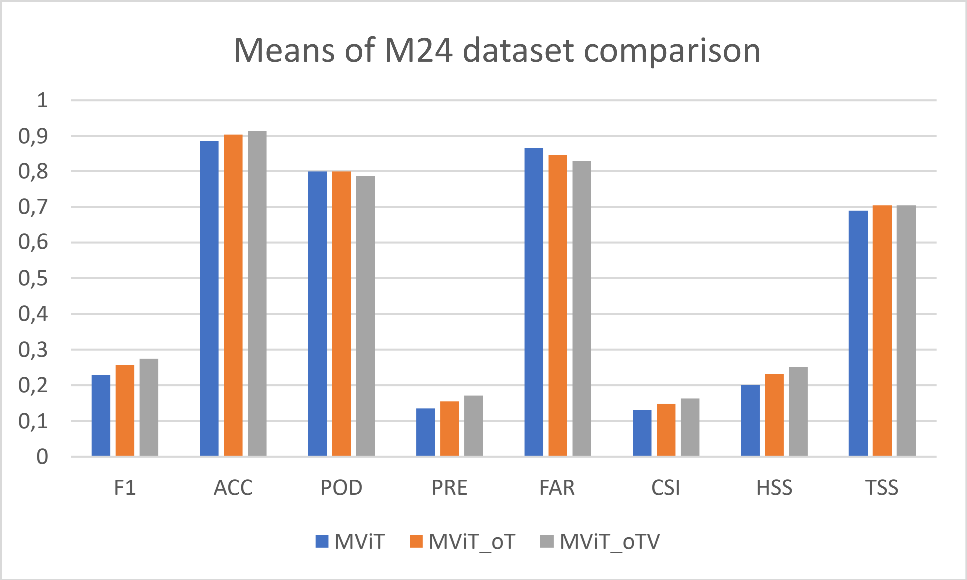


Figure 5.7: Models' performance means on the M24 dataset. Source: Grim and Gradvohl (2024).

Tables 5.8, 5.9, and 5.10 present the individual results, the means, and the SD for the M48 test set. Our models achieved better results in the M48 test set, with TSS ≈ 0.8, compared to the M24 test set. The other secondary performance metrics also present significant improvements.

Table 5.8: SF_MViT test results for ≥M-class flares within the 48 h forecasting horizon.

SF_MViT results for M48 Test								
Split	F1	ACC	POD	PRE	FAR	CSI	HSS	TSS
1	0.49	0.92	0.90	0.34	0.66	0.32	0.45	**0.82**
2	0.49	0.92	0.91	0.34	0.66	0.32	0.46	**0.82**
3	0.33	0.84	0.92	0.20	0.80	0.20	0.28	0.76
4	0.43	0.90	0.91	0.29	0.71	0.28	0.39	0.80
5	0.43	0.90	0.90	0.28	0.72	0.28	0.39	0.80
Means	0.43	0.90	0.91	0.29	0.71	0.28	0.39	0.80
SD	±0.07	±0.03	±0.01	±0.06	±0.06	±0.05	±0.07	±0.02

The M48 test set's RoPS is 4.40%, more than double that of the M24 test set's RoPS, which can have contributed to these performance improvements since it also extends to training and validation sets, facilitating the learning process from positive samples in comparison with M24 dataset. Beyond TSS, we noticed significant improvements (approximately +0.10) in all other metrics for the M48 test set, compared to the same models in the M24 test set, except for ACC. The most notable SD was 0.08 for the HSS on the SF_MViT_oT model.

Table 5.9: SF_MViT_oT test results for ≥M-class flares within the 48 h forecasting horizon.

SF_MViT_oT results for M48 Test								
Split	F1	ACC	POD	PRE	FAR	CSI	HSS	TSS
1	0.54	0.93	0.90	0.39	0.61	0.37	0.51	**0.83**
2	0.54	0.93	0.87	0.39	0.61	0.37	0.51	0.80
3	0.37	0.87	0.88	0.23	0.77	0.23	0.32	0.74
4	0.54	0.93	0.90	0.38	0.62	0.37	0.51	**0.83**
5	0.50	0.92	0.89	0.34	0.66	0.33	0.46	0.81
Means	0.50	0.92	0.89	0.35	0.65	0.33	0.46	0.80
SD	±0.07	±0.03	±0.01	±0.07	±0.07	±0.06	±0.08	±0.04

Table 5.10: SF_MViT_oTV test results for ≥M-class flares within the 48 h forecasting horizon.

SF_MViT_oTV results for M48 Test								
Split	F1	ACC	POD	PRE	FAR	CSI	HSS	TSS
1	0.51	0.94	0.73	0.39	0.61	0.34	0.48	0.68
2	0.48	0.93	0.75	0.36	0.64	0.32	0.45	0.69
3	0.50	0.92	0.89	0.35	0.65	0.33	0.46	**0.82**
4	0.54	0.93	0.86	0.39	0.61	0.37	0.51	0.80
5	0.52	0.93	0.87	0.37	0.63	0.35	0.49	0.80
Means	0.51	0.93	0.82	0.37	0.63	0.34	0.48	0.76
SD	±0.02	±0.01	±0.07	±0.02	±0.02	±0.02	±0.02	±0.07

Figure 5.8 illustrates the models' performance means trained on the M48 dataset. We noticed the improvements in SF_MViT_oT and the SF_MViT_oTV compared to SF_MViT. Analyzing tables 5.8, 5.9, and 5.10 and comparing Figure 5.8 to Figure 5.7, we observed a similar trend in the improvements on the M48 test set, where the oversampled models outperformed SF_MViT in most means. However, the following points deserve attention:

- The prominent decrease at mean SF_MViT_oTV POD metric (−0.09).
- The decrease at mean SF_MViT_oTV TSS metric (−0.04).
- The significant decrease at SF_MViT_oTV POD (−0.17) and TSS (−0.14) metrics in Split 1.

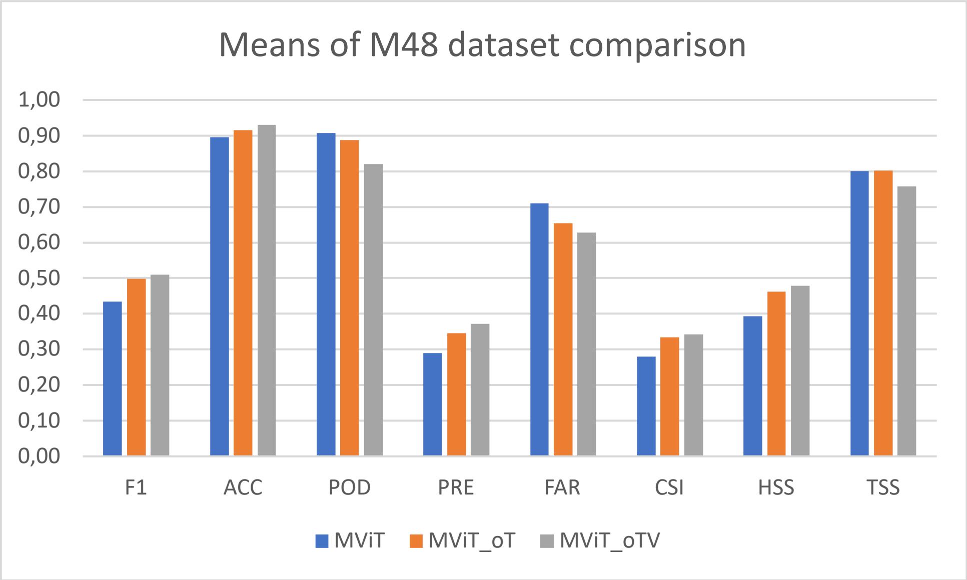


Figure 5.8: Models' performance means on the M48 dataset. Source: Grim and Gradvohl (2024).

Regarding the M24 dataset, Table 5.7 shows a pronounced drop in SF_MViT_oTV's POD and TSS metrics in Split 1 compared to SF_MViT. The Split 1 represents the original data's arrangement in chronological order. As Tang, Liao, et al. (2021) discussed, the POD values impact the TSS, especially when the TSS ≥ 0.4. As a result, applying oversampling to both the training and validation sets (SF_MViT_oTV model) enhanced the performance of most metrics. However, these improvements came at the expense of decreasing the POD and TSS.

On the other hand, employing oversampling only on the training set (SF_MViT_oT model) enhanced most of the performance metrics while keeping the TSS stable compared to SF_MViT. According to tables 5.8 and 5.9, these improvements in the M48 dataset were HSS and F1 by $+0.07$, PRE by $+0.06$, FAR by -0.06, CSI by $+0.05$, and ACC and POD by $+0.02$. While, according to tables 5.5 and 5.6 in the M24 dataset, the improvements were HSS and F1 by $+0.03$, PRE and CSI by $+0.02$, FAR by -0.02, and ACC by $+0.01$.

We concluded that adopting oversampling was essential for our performance improvement. Past works, such as Xuebao Li et al. (2020) and Deng et al. (2021), have adopted oversampling for ≥M-class flares samples in conjunction with subsampling for <C-class flares samples. However, they also resampled the test set, considering the entire dataset, which could potentially lead to biased results. Furthermore, they did not evaluate the isolated impact of the resampling techniques they adopted.

In this section, we conducted experiments carefully to consider the application of oversampling and the potential impacts on the performance metrics analyzed. Our approach of selectively applying oversampling only in the training set while maintaining the original validation and test sets provided a balanced method to enhance the SF_MViT_oT model performance in almost all analyzed metrics, except for POD, without compromising the reliability evaluation of the TSS.

5.3.1 Inference Testing Times

Despite training and testing our models on the CENAPAD-SP environment described in Section 4.4.2, we chose to consider the inference times based on our dedicated desktop located at the School of Technology (FT) of the University of Campinas (UNICAMP). This computer has an 8-core Intel(R) Core(TM) i7-10700 CPU @ 2.90GHz, 16 GB DDR4 RAM, and 1 TB SATA HDD. It also has one NVIDIA GeForce RTX 3060 GPU[1] with 12GB GDDR6 RAM.

We imported the training weights of each model's best-performing split in our desktop to run the inference mode in GPU and CPU processing. Table 5.11 shows the GPU Total Time, the GPU Time per Sample, the Total CPU Time, and the CPU Time per Sample of our models testing in the M24 and M48 datasets. Remembering that the total testing samples are 9,384.

Table 5.11: Inference testing times (in seconds).

Dataset	Model	Best Split	GPU Total Time	GPU Time per Sample	CPU Total Time	CPU Time per Sample
M24	SF_MViT	4	403	0.04	11,793	1.26
	SF_MViT_oT	0	411	0.04	11,824	1.26
	SF_MViT_oTV	3	409	0.04	11,592	1.24
M48	SF_MViT	2	401	0.04	12,287	1.31
	SF_MViT_oT	1	403	0.04	9,518	1.01
	SF_MViT_oTV	3	403	0.04	12,952	1.38

Despite the long training times our models experienced in a high-performance computing environment presented in tables 5.2 and 5.4, the inference testing times presented in Table 5.11 show that our models can run in inference mode on modest desktop computers and are suitable even for near real-time data fluxes.

[1] Available at https://www.nvidia.com/en-us/geforce/graphics-cards/30-series/rtx-3060-3060ti/

5.4 Literature Comparison

We assessed our findings by comparing them to related works adopting line-of-sight magnetograms for forecasting ≥M-class flares. Our initial goal was to directly compare the performance of our models with Tang, Liao, et al. (2021)'s results since both of us adopted the SOLAR-STORM2 dataset within the same period. Tang, Liao, et al. (2021) enhanced their models by focusing on improving specific performance metrics, creating optimized models for the TSS (TSS_FFM) and F1 (F1_FFM) scores.

We selected the Split 1 results of our models, which maintain the dataset in the original chronological order, to compare with the preliminary results of Tang, Liao, et al. (2021). They considered its F1_FFM model trained for predicting ≥M-class flares within the 48 h forecasting horizon, employing the same data splitting of our Split 1 in their preliminary tests: the first 50,000 samples for training, the following 10,000 samples for validation, and the final 9,384 samples for testing. However, as Table 5.12 shows, we observed a minor discrepancy in the RoPS within the test set.

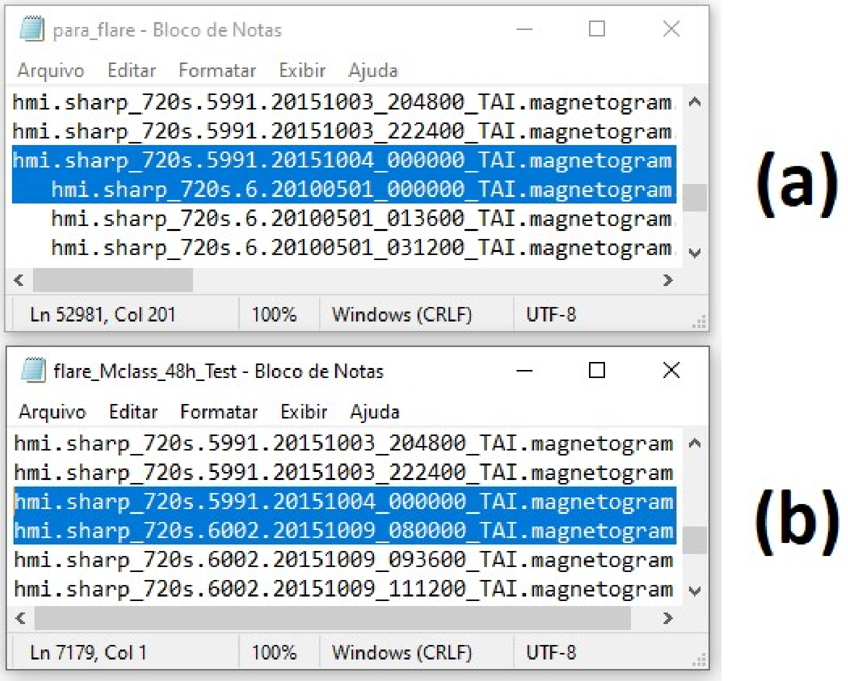
Table 5.12: Performance comparison with Tang, Liao, et al. (2021) results in the same context.

Model	RoPS	F1	ACC	POD	PRE	FAR	CSI	HSS	TSS
SF_MViT*	4.40%	0.49	0.92	0.90	0.34	0.66	0.32	0.45	0.82
SF_MViT_oT*	4.40%	0.54	0.93	0.90	0.39	0.61	0.37	0.51	**0.83**
SF_MViT_oTV*	4.40%	0.51	0.94	0.73	0.39	0.61	0.34	0.48	0.68
CNN F1_FFM$^{\diamond}$	3.80%	0.41	0.96	0.41	0.42	0.58	0.26	0.39	0.38
DNN F1_FFM$^{\diamond}$	3.80%	0.54	0.95	0.71	0.44	0.56	0.37	0.52	0.67
biLSTM F1_FFM$^{\diamond}$	3.80%	0.45	0.96	0.47	0.44	0.56	0.29	0.43	0.44
Fusion F1_FFM$^{\diamond}$	3.80%	**0.57**	0.97	0.48	0.69	0.31	0.39	0.55	0.47

* Our model uses Split 1. Results refer to our M48 dataset.

$^{\diamond}$ Values computed according to Table 5 of Tang, Liao, et al. (2021).

Although we both followed the same splitting method, Tang, Liao, et al. (2021) reported 357 positive samples in their test set, while ours presented 413 positive samples. This discrepancy is evident in RoPS in Table 5.12. This divergence arises because, despite the SOLAR-STORM2 dataset appearing to be originally ordered by SHARP-IDs, we discovered cases of disordered samples, highlighted in blue in the transitions depicted in Figure 5.9-(a).


para_flare - Bloco de Notas
Arquivo Editar Formatar Exibir Ajuda
Ln 52981, Col 201
100%
Windows (CRLF)
UTF-8
(a)
flare_Mclass_48h_Test - Bloco de Notas
Arquivo Editar Formatar Exibir Ajuda
Ln 7179, Col 1
100%
Windows (CRLF)
UTF-8
(b)

```
hmi.sharp_720s.5991.20151003_204800_TAI.magnetogram
hmi.sharp_720s.5991.20151003_222400_TAI.magnetogram
hmi.sharp_720s.5991.20151004_000000_TAI.magnetogram
    hmi.sharp_720s.6.20100501_000000_TAI.magnetogram
    hmi.sharp_720s.6.20100501_013600_TAI.magnetogram
    hmi.sharp_720s.6.20100501_031200_TAI.magnetogram
```

```
hmi.sharp_720s.5991.20151003_204800_TAI.magnetogram
hmi.sharp_720s.5991.20151003_222400_TAI.magnetogram
hmi.sharp_720s.5991.20151004_000000_TAI.magnetogram
hmi.sharp_720s.6002.20151009_080000_TAI.magnetogram
hmi.sharp_720s.6002.20151009_093600_TAI.magnetogram
hmi.sharp_720s.6002.20151009_111200_TAI.magnetogram
```

Figure 5.9: Forced timestamp reorder in the original dataset. Source: Grim and Gradvohl (2024)

Then, we reordered the original dataset considering the crescent SHARP-ID order, as illustrated in Figure 5.9-(b). In most cases, the solar active regions are numbered with crescent SHARP-IDs according to how the instruments detect them. The numbers follow a chronological order, bringing our dataset closer to reality. We believe this reordering may have caused the mentioned sample's divergence.

Now that we have clarified this point, we can make the appropriate comparisons. Table 5.12 presents the results from Tang, Liao, et al. (2021), whose most models optimized for the F1 score, except the DNN F1_FFM, significantly penalized the TSS score results.

In contrast to optimizing the models for a performance metric, we optimized our models by monitoring the weighted Val_loss, selecting the epoch that presented the lowest Val_Loss to perform the test step. As a result, our models achieved more balanced outcomes, obtaining the highest TSS in Table 5.12. The F1 and other metrics showed values closer to Tang, Liao, et al. (2021)'s results, except for PRE and FAR, which performed worse in our models.

Considering the models with the best TSS of each author, our top-performing model, SF_MViT_oT, matched the F1 and CSI of Tang, Liao, et al. (2021) DNN F1_FFM model. Additionally, it showed a higher POD of +0.19 and TSS of +0.16, while having slightly lower ACC of −0.02, PRE of −0.05, FAR of −0.05, and HSS of −0.01, as detailed in Table 5.12.

5.4.1 Indirectly Compared Works

As highlighted in most related literature, it is challenging to make a quantitative performance comparison of different solar flare forecasting models due to variations in datasets, methods, and analysis techniques adopted (HUANG, X.; WANG, H.; XU; LIU, J., et al., 2018; NISHIZUKA; SUGIURA; KUBO; DEN; ISHII, 2018; TANG; LIAO, et al., 2021).

To tackle this problem, we followed the approach commonly seen in related works: comparing models that used similar datasets and methods. Given the substantial class imbalance of positive M- and X-class flares samples, we adopted the test set's RoPS as the basic parameter for comparing our works with the literature.

The degree of the test set's class imbalance can directly impact performance evaluation. Hence, we presented the RoPS for each study. Tables 5.13 and 5.14 compare our findings with similar state-of-the-art works adopting line-of-sight magnetograms for predicting ≥M-class flares with closer RoPS. In our results, we considered the means across the splits. Table 5.13 shows the comparison for the 48 h forecasting horizon.

Table 5.13: Comparison for predicting ≥M-class flares within the 48 h forecasting horizon.

Model	RoPS	F1	ACC	POD	PRE	FAR	CSI	HSS	TSS
SF_MViT[⋆]	4.40%	0.43	0.90	0.91	0.29	0.71	0.28	0.39	**0.80**
SF_MViT_oT[⋆]	4.40%	0.50	0.92	0.89	0.35	0.65	0.33	0.46	**0.80**
SF_MViT_oTV[⋆]	4.40%	0.51	0.93	0.82	0.37	0.63	0.34	0.48	0.76
F1_FFM[⋄]	4.75%	0.52	0.96	0.49	0.58	—	—	—	0.48
TSS_FFM[⋄]	4.75%	0.42	0.89	0.83	0.28	—	—	—	0.72
Huang−2018[§]	4.11%	0.26	0.81	0.81	0.16	0.84	0.14	0.20	0.62

[⋆] Our model. Considering the means between all splits.

[⋄] Tang, Liao, et al. (2021) obtained their result with the cross-validation method, according to tables 7, 8, and 9 of their paper. They only mentioned the RoPS from the entire dataset in their Table 9. They did not mention the FAR, CSI, and HSS metrics.

[§] Values computed through Table 4 of Xin Huang, Huaning Wang, Xu, Jinfu Liu, et al. (2018).

Table 5.13 shows that, in this context, our models outperformed the most metrics, with SF_MViT_oT achieving the highest TSS = 0.80 compared with Tang, Liao, et al. (2021) TSS_FFM model (TSS increased +0.08) and Xin Huang, Huaning Wang, Xu, Jinfu Liu, et al.

(2018) (TSS increased +0.18). Although we both adopted the same dataset, Tang, Liao, et al. (2021) obtained their results by adopting the 4-fold cross-validation method for training and evaluating their models. Beyond our work, only Xin Huang, Huaning Wang, Xu, Jinfu Liu, et al. (2018) and Tang, Liao, et al. (2021) have considered using line-of-sight magnetogram image-based as inputs to predict ≥M-class flares within the 48 h forecasting horizon.

Within the 24 h forecasting horizon, our oversampled models (SF_MViT_oT and SF_MViT_oTV) outperformed Kaneda et al. (2023) (TSS increased +0.17) and Xin Huang, Huaning Wang, Xu, Jinfu Liu, et al. (2018) (TSS increased +0.04), as Table 5.14 shows. Our models also performed comparably to Tang, Liao, et al. (2021) TSS_FFM (TSS decreased −0.02) and Guastavino et al. (2022) (TSS increased +0.02), even though they present the lowest RoPS.

Table 5.14: Comparison for predicting ≥M-class flares within the 24 h forecasting horizon.

Model	RoPS	F1	ACC	POD	PRE	FAR	CSI	HSS	TSS
SF_MViT[⋆]	2.08%	0.23	0.89	0.80	0.13	0.87	0.13	0.20	0.69
SF_MViT_oT[⋆]	2.08%	0.26	0.90	0.80	0.15	0.85	0.15	0.23	0.70
SF_MViT_oTV[⋆]	2.08%	0.27	0.91	0.79	0.17	0.83	0.16	0.25	0.70
F1_FFM[⋄]	2.54%	0.38	0.97	0.37	0.41	—	—	—	0.36
TSS_FFM[⋄]	2.54%	0.23	0.84	0.88	0.13	—	—	—	0.72
Huang–2018[§]	2.42%	0.18	0.81	0.85	0.10	0.90	0.10	0.14	0.66
Deep Flare Net (DeFN)[†]	3.25%	0.30	0.86	0.95	0.18	0.82	0.18	0.26	**0.80**
Guastavino et al. (2022)[‡]	3.21%	—	—	—	—	—	—	—	0.68
Kaneda et al. (2023)[⊙]	8.54%	—	—	—	—	—	—	—	0.53

[⋆] Our model. Considering the means between all splits.

[⋄] Tang, Liao, et al. (2021) obtained their result with the cross-validation method, according to tables 7, 8, and 9 of their paper. They only mentioned the RoPS from the entire dataset in their Table 9. They did not mention the FAR, CSI, and HSS metrics.

[§] Values computed through Table 4 of Xin Huang, Huaning Wang, Xu, Jinfu Liu, et al. (2018).

[†] Model from Nishizuka, Sugiura, Kubo, Den, and Ishii (2018). Values from their Table 3.

[‡] We considered the mean TSS, including all samples, from Table 2 and Figure 5 of Guastavino et al. (2022) for M1+ flares forecasting. They mentioned only the approximated RoPS of ≥M-class flares ($p_M \approx 3.21\%$) of the entire dataset in Section 2.2. They did not mention confusion matrices or other performance metrics besides the TSS score.

[⊙] We considered the mean TSS mentioned in Table 4 of Kaneda et al. (2023). They only mentioned the RoPS of ≥M-class flares concerning the entire dataset in Section 5.1 of their paper. They also considered the Gandin–Murphy–Gerrity Score (GMGS) and Brier Skill Score (BSS).

According to Table 5.14, our models performed worse than the DeFN model from Nishizuka, Sugiura, Kubo, Den, and Ishii (2018) (TSS decreased approximately −0.10). We must emphasize that the testing period they adopted was the entire year of 2015, which is close to ours (from February 2015 to December 2015). However, they downsampled their dataset frequency to 1 h, while we downloaded the SOLAR-STORM2 dataset already downsampled by 1.5 h (TIANCHI, 2020).

As a result, according to Figure 5 of Nishizuka, Sugiura, Kubo, Den, and Ishii (2018), their test set includes 31,336 samples with RoPS of 3.25%, while our test set contains 9,384 samples with RoPS of 2.08%. They also did not adopt a validation set, taking the best model with the TSS optimized directly on the test set, which may have favored the DeFN model's performance.

Regarding Tang, Liao, et al. (2021)'s F1_FFM models, tables 5.13 and 5.14 clearly show how these models penalize the TSS score despite reaching the best F1. Regardless of any metrics optimizations, our oversampled models (SF_MViT_oT and SF_MViT_oTV) achieved F1 = 0.50, values closer to Tang, Liao, et al. (2021) F1_FFM model (F1 = 0.52) while also attaining the best TSS = 0.80 within the 48 h forecasting horizon, according to Table 5.13.

5.4.2 Biased Results Comparison

As mentioned earlier, Xuebao Li et al. (2020) and Deng et al. (2021) adopted oversampling for positive ≥M-class flares and undersampling for negative < M-class flares samples across the entire dataset, which increased the RoPS throughout their dataset, including the test set.

Although Pengchao Sun et al. (2022) did not use oversampling, their dataset selected only certain active regions without flares comprising the period from 2010 to 2019, leading to a RoPS of 13.29% for ≥M-class flares within the 24 h forecasting horizon. However, according to Guastavino et al. (2022), the RoPS in the HMI archive from September 2012 to September 2017 was approximately 3.20%, considering the same scenario.

Test sets with higher RoPS can lead to an optimistic bias in the results. To confirm this issue in practice, we created a new model that applied our oversampling techniques for the entire M24 and M48 datasets, including their test sets. Table 5.15 presents the results for this model, referred to as "SF_MViT_oTV Test", comparing with SF_MViT_oT and correlated works. It shows that oversampling our test sets increased the RoPS from 2.08% to 11.29% in the M24 and 4.40% to 21.64% in the M48 test sets. This approach resulted in a biased improvement across all SF_MViT_oTV Test's performance metrics, compared to SF_MViT_oT.

Table 5.15: Comparison for ≥M-class flares with biased test sets.

Model	RoPS	F1	ACC	POD	PRE	FAR	CSI	HSS	TSS
M24 SF_MViT_oT ⋆	2.08%	0.26	0.90	0.80	0.15	0.85	0.15	0.23	0.70
M24 SF_MViT_oTV Test ⋆⋆	11.29%	0.69	0.91	0.86	0.57	0.43	0.52	0.63	0.77
M48 SF_MViT_oT ⋆	4.40%	0.50	0.92	0.89	0.35	0.65	0.33	0.46	0.80
M48 SF_MViT_oTV Test ⋆⋆	21.64%	0.85	0.93	0.90	0.79	0.21	0.73	0.80	**0.84**
Deng et al. (2021)×	49.70%	0.90	0.88	0.88	0.91	—	—	0.76	0.77
Xuebao Li et al. (2020)⊗	38.94%	0.85	0.89	0.93	0.89	0.11	—	0.76	0.75
Pengchao Sun et al. (2022)⊙	13.29%	0.72	0.90	0.93	0.59	0.10	—	0.67	0.83

⋆ Our model. Considering the means between the splits.

⋆⋆ Our model. Considering the split's means with the test set artificially oversampled.

× We considered the metrics from column '≥M-class (M_1)' of Deng et al. (2021)'s Table 8. We computed the RoPS from their Table 4, considering the M-class and X-class flares as positive samples and the mean of all test sets. They did not mention the FAR and CSI.

⊗ We considered the metrics from Table 2 of Xuebao Li et al. (2020). We computed the RoPS according to their Figure 2, considering the M-class and X-class flares as positive samples and the mean of all test sets. They did not mention the CSI score.

⊙ We considered the metrics from Table 5 of Pengchao Sun et al. (2022). We calculated the rate RoPS according to their Table 1.They did not mention the CSI score.

According to Table 5.15, the most impacted metrics (F1, PRE, FAR, CSI, and HSS) improved by more than 0.30 between SF_MViT_oT and "SF_MViT_oTV Test" in both scenarios. The ACC remained stable, increasing by 0.01 points in both scenarios. The POD and TSS increased by 0.06 and 0.07, respectively, in the M24 dataset and by 0.02 and 0.04 in the M48 dataset.

These results supported our hypothesis, showing us that resampling the test set biases the test results toward better performance. They also highlight the problem of comparing studies with vastly different RoPSs. Table 5.15 illustrates the discrepancies between the SF_MViT_oT and "SF_MViT_oTV Test" results, showing how the latter are closer to the results of Xuebao Li et al. (2020) and Deng et al. (2021), and Pengchao Sun et al. (2022).

Owing to the reduced variation concerning the class data imbalance, the TSS has been considered the primary metric for evaluating solar flare forecasting performance in classification problems since Bobra and Couvidat (2015). Even so, most authors usually evaluate the solar flare forecasting models' performance, considering the other performance metrics covered in this thesis as secondary, together with the TSS.

5.5 Concluding Remarks

The results from our models (SF_MViT, SF_MViT_oT, and SF_MViT_oTV) indicate that the oversampled models (SF_MViT_oT and SF_MViT_oTV) achieved faster stabilization of Val_Loss and Val_ACC. The SF_MViT model without oversampling required more epochs to converge, indicating that data augmentation positively influences training speed and model stability. The oversampling techniques implemented in our models demonstrated their effectiveness in accelerating the learning process by reducing the Estimated Total Training Time in the M24 dataset.

During the test step, our oversampled models showed improved performance metrics compared to the SF_MViT model, particularly regarding F1, PRE, FAR, CSI, and HSS. These improvements were more pronounced in the 48 h forecasting horizon, suggesting that data augmentation techniques can enhance model performance in the solar flare forecasting domain for extended prediction periods. The consistency of the results across multiple splits in two datasets underscores the robustness of our models.

Despite the long training times experienced in a high-performance computing environment, the inference testing times obtained on a desktop with GPU show that our models can run in inference mode on desktop computers, and are suitable even for near real-time data fluxes.

Compared with state-of-the-art works, our results highlight that our models, particularly SF_MViT_oT, are competitive. SF_MViT_oT model achieved the highest TSS values (TSS = 0.80) in the 48 h forecasting horizon, outperforming the related works. We believe the following factors were crucial to our performance:

- The adoption of SHARP line-of-sight magnetogram sequences as inputs;
- The adoption of the MViT-S based model, which is much larger than those compared;
- The use of pre-trained weights with the KINETICS-400 dataset, instead from training from scratch;
- The fine-tuning training methods adopted, such as learning rate decrease rule and weighted cross-entropy loss with distinct weights in training and validation;
- And the adoption of data augmentation in positive samples of magnetogram sequences.

In the 24 h forecasting horizon, our oversampled models performed closer to the most correlated works (TSS ≈ 0.70 ± 2), except for Nishizuka, Sugiura, Kubo, Den, and Ishii (2018), which performed better. However, as mentioned earlier, they presented higher RoPS and optimized the DeFN model directly in the testing set, which may have favored their performance.

Furthermore, although we do not recommend oversampling the test set, we trained a model using oversampling also in the test set. In this model, referred to as the "SF_MViT_oTV Test", the results showed an artificial and significant improvement (≥ 0.10) in most performance metrics analyzed, with less impact only on the ACC (+0,01), POD (+0.04), and TSS (+0.04), comparing with their respective SF_MViT_oT model, in the 24 h and 48 h forecasting horizons.

Chapter 6

Conclusions

Our work introduced Solar Flare Forecasting MViT models (SF_MViTs), developed by applying fine-tuning on modified Torchvision's Multiscale Vision Transformers (MViTv2_s). Derived from Improved MViT-S (LI, Y. et al., 2022), our developed models can learn spatiotemporal correlations in sequences of images or videos. In the solar flare forecasting domain, these models can learn from Active Region's (AR) magnetogram image sequences implicit parameters about the organization and structure of sunspots in the spatial axis and parameters related to sunspots' evolution in the temporal axis.

Since these parameters may have correlations with solar flare occurrences in the future, we take observations of line-of-sight magnetogram image sequences, with a correspondent label indicating if it will generate a solar flare, as inputs to train our three developed models: SF_MViT with original data, SF_MViT_oT with oversampling of positive ≥M-class flare samples on the training set, and SF_MViT_oTV with oversampling of positive ≥M-class flare samples on both training and validation sets.

We built our magnetogram image sequences from the SOLAR-STORM2 dataset (TIANCHI, 2020; HUANG, X.; WANG, H.; XU; LIU, J., et al., 2018) considering the same period adopted by Tang, Liao, et al. (2021), from 2010 to 2015. Initially, we adopted the SOLAR-STORM2 dataset for predicting ≥M-class flares within the 48 h forecasting horizon. Additionally, we created the 24 h forecasting horizon by relabeling the samples according to the active regions classified by NOAA inside each SHARP-ID.

Our approach contrasts with most correlated works, which typically focus on training deep-learning models built specifically for solar flare forecasting from scratch. Instead, we loaded MViTv2, already trained with a general-purpose dataset, and adapted it by applying fine-tuning training (also known as transfer learning) through our SHARP magnetogram sequenced dataset. By adopting transfer learning, we were able to converge the training of a deeper transformer compared to the models presented in our literature survey with a proportionally modest dataset.

Given our results, with a mean True Skill Statistic (TSS) of approximately 0.70 for forecasting ≥M-class flares within 24 h forecasting horizon, a mean TSS of around 0.80 for forecasting ≥M-class flares within the 48 h forecasting horizon, and a TSS standard deviation of about 0.04 in both scenarios, we concluded that our fine-tuned models are a valid and stable approach for solar flare forecasting. Comparing our three models, we found that adopting oversampling in both the training and validation sets (SF_MViT_oTV) enhanced the performance of most metrics. However, it decreased the Probability of Detection (POD) metric, which had consequences for the TSS. Using oversampling only in the training set (SF_MViT_oT) improved most metrics while keeping the TSS stable.

Our models outperformed the state-of-the-art works for forecasting ≥M-class flares within the 48 h forecasting horizon. Our models also presented positive results for the 24 h forecasting horizon, even though they performed worse than the DeFN model from Nishizuka, Sugiura, Kubo, Den, and Ishii (2018). Additionally, our results demonstrated the benefits of using a weighted loss function instead of a metric-optimized loss, providing more balanced results between F1 and TSS scores than the Tang, Liao, et al. (2021)'s models.

As widely discussed in this text and the most relevant works, it is challenging to quantitatively compare the performance of such different solar flare forecasting models, given the variations in methods, analysis techniques, and datasets adopted. Thus, we aimed to compare our work with studies adopting similar datasets and methods. Given the significant imbalance level of ≥M-class flares, we considered the test set's Rate of Positive Samples (RoPS) as the basis for comparison because it directly impacts the performance metrics.

We proved this by applying oversampling in the test set, which increased the test set's RoPS and led to a biased improvement in all SF_MViT_oTV Test's performance metrics compared to the SF_MViT_oT model. The most affected metrics were F1, PRE, FAR, CSI, and HSS, improving by over 0.30 points in all scenarios. The TSS score increased by 0.07 within the 24 h and 0.04 within the 48 h forecasting horizon, showing slightly more variation than the other performance metrics.

6.1 Related Publications

We published the results of this work in the Solar Physics Journal (GRIM; GRADVOHL, 2024). We also make the entire dataset available to facilitate other researchers' understanding and reproduction of our work. The base dataset, image-sequence reference, source code, input, output, and checkpoint files are available in our data repository (GRIM; GRADVOHL, 2023b).

Our source code repository contains a clean project version containing only the image-sequence reference, input, source code, and submission scripts (GRIM; GRADVOHL, 2023a).

6.2 Future Works

For future works, we suggest that the solar flare forecasting community provide a standard dataset to facilitate comparison of works. The dataset should contain numerical parameters and line-of-sight magnetograms accompanied by their respective labels for the 24 h and 48 h forecasting horizons, for ≥ C- and X-class flares.

We consider improving our models by adopting datasets with data from 2015 onwards to fit the operational scenario described by Cinto, Gradvohl, et al. (2020a). With the incoming peak of Solar Cycle 25, we recommend an approach that performs training and validation with data from Solar Cycle 24 and tests with data from Solar Cycle 25.

It would also be possible to build an ensemble with the five trained versions for each Split of our SF_MViT, SF_MViT_oT, and SF_MViT_oTV models by adding a majority voting mechanism or a fusion layer at the end, similar to Tang, Liao, et al. (2021) work.

Another interesting proposal would be to analyze the adoption of data augmentation techniques not considered in this thesis due to their implementation complexity, which would be the adoption of noise, image mixing, and synthetic data generation.

Due to Torchvision's MViT_v2 model's load limitation, we could only use a sequence with 16 magnetograms. We believe adopting an extensive sequence can favor parameter learning in training. However, this would lead to adopting computing environments with greater processing capacity, including next-generation GPUs, sufficient RAM, and multi-GPU approaches.

Even with a sequence of 16 magnetograms, we faced limitations considering the computational environment we used, such as RAM overflows, especially with the model SF_MViT_oTV. To face these problems, we developed a version of our models capable of running on multiple GPUs, which is also available in our source code repository (GRIM; GRADVOHL, 2023a). However, we experienced lengthy delays in starting job execution on the CENAPAD queue for two GPUs (“duasgpus” queue) because it was much congested, as we detail in Appendix B. Thus, our Multi-GPU version of SF_MViT, SF_MViT_oT, and SF_MViT_oTV models has the potential to be much explored and improved.

Another proposal is to develop a forecasting system integrating deep-learning models from distinct image-processing tasks. For example, from a continuous input of full-disk solar images (magnetograms or images of the upper layers of the solar atmosphere), we would have a system capable of detecting and segmenting solar ARs that could serve as inputs for solar flare forecasting models such as the ones we developed.

This procedure is necessary since the magnetic classification process of solar ARs is a once-a-day process, with the magnetic classifications being published at 00:30 UTC each day based on information from the previous day. As the dynamics of solar ARs can vary on timescales of a few hours, this information becomes outdated during periods of maximum solar activity cycle or more active periods. Then, we would need to create a dataset with the known AR position descriptions to train a model for AR detection.

Additionally, to segment ARs, the respective known pixel masks would be necessary. In this sense, other works can employ models based on R-CNN, such as Mask R-CNN (MINAEE et al., 2021) or YOLO (REDMON, J.; FARHADI, 2018), combined with U-Net (RONNEBERGER; FISCHER; BROX, 2015) networks for ARs’ detection and segmentation.

References

ABED, A. K.; QAHWAJI, R.; ABED, A. The automated prediction of solar flares from SDO images using deep learning. **Advances in Space Research**, COSPAR, v. 67, n. 8, p. 2544–2557, Apr. 2021. ISSN 02731177. DOI: 10.1016/j.asr.2021.01.042.

AHMED, O. W. et al. Solar Flare Prediction Using Advanced Feature Extraction, Machine Learning, and Feature Selection. **Solar Physics**, v. 283, n. 1, p. 157–175, Mar. 2013. ISSN 0038-0938. DOI: 10.1007/s11207-011-9896-1.

BABCOCK, H. W. The Topology of the Sun's Magnetic Field and the 22-Year Cycle. **The Astrophysical Journal**, v. 133, p. 572, Mar. 1961. DOI: 10.1086/147060.

BARNES, G. et al. A COMPARISON OF FLARE FORECASTING METHODS. I. RESULTS FROM THE "ALL-CLEAR" WORKSHOP. **The Astrophysical Journal**, v. 829, n. 2, p. 89, Sept. 2016. ISSN 1538-4357. DOI: 10.3847/0004-637X/829/2/89.

BARNES, L. R. et al. CORRIGENDUM: False Alarm Rate or False Alarm Ratio? **Weather and Forecasting**, v. 24, n. 5, p. 1452–1454, Oct. 2009. ISSN 1520-0434. DOI: 10.1175/2009WAF2222300.1.

BLOOMFIELD, D. S.; HIGGINS, P. A.; MCATEER, R. T.; GALLAGHER, P. T. Toward reliable benchmarking of solar flare forecasting methods. **Astrophysical Journal Letters**, v. 747, n. 2, 2012. ISSN 20418205. DOI: 10.1088/2041-8205/747/2/L41.

BOBRA, M. G.; COUVIDAT, S. Solar flare prediction using SDO/HMI vector magnetic field data with a machine-learning algorithm. **Astrophysical Journal**, v. 798, n. 2, 2015. ISSN 15384357. DOI: 10.1088/0004-637X/798/2/135.

BOBRA, M. G.; SUN, X., et al. The Helioseismic and Magnetic Imager (HMI) Vector Magnetic Field Pipeline: SHARPs – Space-Weather HMI Active Region Patches. **Solar Physics**, v. 289, n. 9, p. 3549–3578, Sept. 2014. ISSN 0038-0938. DOI: 10.1007/s11207-014-0529-3.

CALDANA, I.; SILVA, A. E. A. da; COELHO, G. P.; GRADVOHL, A. L. S. Using X-Ray Flux Time Series for Solar Explosion Forecasting. In: 2017 Brazilian Conference on Intelligent Systems (BRACIS). Uberlândia: IEEE, Oct. 2017. P. 204–209. ISBN 978-1-5386-2407-4. DOI: 10.1109/BRACIS.2017.31.

CAMPOREALE, E. The Challenge of Machine Learning in Space Weather: Nowcasting and Forecasting. **Space Weather**, v. 17, n. 8, p. 1166–1207, 2019. ISSN 15427390. DOI: 10.1029/2018SW002061.

CINTO, T.; GRADVOHL, A. L. S.; COELHO, G. P.; SILVA, A. E. A. da. A framework for designing and evaluating solar flare forecasting systems. **Monthly Notices of the Royal

Astronomical Society, Oxford University Press, v. 495, n. 3, p. 3332–3349, 2020. ISSN 0035-8711. DOI: 10.1093/mnras/staa1257.

CINTO, T.; GRADVOHL, A. L. S.; COELHO, G. P.; SILVA, A. E. A. da. Solar Flares Forecasting Using Time Series and Extreme Gradient Boosting Ensembles. **Solar Physics**, Springer Nature B.V., 2020. ISSN 0038-0938. DOI: 10.1007/s11207-020-01661-9.

CINTO, T. **Solar flare forecasting: a methodology to automate the design of classifiers for events of diverse classes**. 2020. S. 180. PhD thesis – Faculdade de Tecnologia da Universidade Estadual de Campinas, Limeira, SP.

COLAK, T.; QAHWAJI, R. Automated McIntosh-Based Classification of Sunspot Groups Using MDI Images. **Solar Physics**, v. 248, n. 2, p. 277–296, Apr. 2008. ISSN 0038-0938. DOI: 10.1007/s11207-007-9094-3.

COLAK, T.; QAHWAJI, R. Automated Solar Activity Prediction: A hybrid computer platform using machine learning and solar imaging for automated prediction of solar flares. **Space Weather**, v. 7, n. 6, June 2009. ISSN 1542-7390. DOI: 10.1029/2008SW000401.

CUNNINGHAM, P.; CORD, M.; DELANY, S. J. Supervised Learning. In: **Machine Learning Techniques for Multimedia: Case Studies on Organization and Retrieval**. Ed. by Matthieu Cord and Pádraig Cunningham. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. P. 21–49. ISBN 978-3-540-75171-7. DOI: 10.1007/978-3-540-75171-7_2.

DENG, Z. et al. Fine-grained Solar Flare Forecasting Based on the Hybrid Convolutional Neural Networks*. **The Astrophysical Journal**, IOP Publishing, v. 922, n. 2, p. 232, Dec. 2021. ISSN 0004-637X. DOI: 10.3847/1538-4357/ac2b2b.

DOSOVITSKIY, A. et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv, 2021. DOI: 10.48550/ARXIV.2010.11929. eprint: 2010.11929 (cs.CV).

ECHER, E. et al. Introduction to space weather. **Advances in Space Research**, v. 35, n. 5, p. 855–865, Jan. 2005. ISSN 02731177. DOI: 10.1016/j.asr.2005.02.098.

FAN, H. et al. Multiscale Vision Transformers. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV). Montreal, Canada: IEEE, 2021. P. 6804–6815. DOI: 10.1109/ICCV48922.2021.00675.

FANG, T.-W. et al. Space Weather Environment During the SpaceX Starlink Satellite Loss in February 2022. **Space Weather**, John Wiley and Sons Inc, v. 20, 11 Nov. 2022. ISSN 1542-7390. DOI: 10.1029/2022SW003193.

FANG, Y.; CUI, Y.; AO, X. Deep Learning for Automatic Recognition of Magnetic Type in Sunspot Groups. **Advances in Astronomy**, v. 2019, p. 1–10, Aug. 2019. ISSN 1687-7969. DOI: 10.1155/2019/9196234.

GÉRON, A. **Hands-On Machine Learning with Scikit-Learn**. Ed. by Nicole Tache. First Edit. Sebastopol, CA: O'Reilly Media, 2017. P. 566. ISBN 9781491962299.

GOODFELLOW, I. J. et al. Generative Adversarial Networks. **Communications of the ACM**, v. 63, n. 11, p. 139–144, June 2014. ISSN 15577317. DOI: 10.1145/3422622.

GRIM, L. F. L.; GRADVOHL, A. L. S. Solar Flare Forecasting Based on Magnetogram Sequences Learning with Multiscale Vision Transformers and Data Augmentation Techniques. **Solar Physics**, Springer Science and Business Media B.V., v. 299, p. 33, 3 Mar. 2024. ISSN 0038-0938. DOI: 10.1007/s11207-024-02276-0.

GRIM, L. F. L.; GRADVOHL, A. L. S. **Solar flare forecasting based on magnetogram sequences learning with multiscale vision transformers and data augmentation techniques**. Dec. 2023. Available from: <https://github.com/lfgrim/SFF_MagSeq_MViTs>. Visited on: 11 July 2024.

GRIM, L. F. L.; GRADVOHL, A. L. S. Automatic Analysis of Magnetogram Sequences for Solar Flares Forecasting. **Journal of Production and Automation**, v. 5, p. 2–9, 2 Dec. 2022. DOI: 10.5281/zenodo.7504243.

GRIM, L. F. L.; GRADVOHL, A. L. S. **Solar flare forecasting based on magnetogram sequences learning with MViT and data augmentation**. Version 0.1. Genebra: Zenodo, Dec. 2023. DOI: 10.5281/zenodo.10246577.

GUASTAVINO, S. et al. Implementation paradigm for supervised flare forecasting studies: A deep learning application with video data. **Astronomy and Astrophysics**, EDP Sciences, v. 662, a105, June 2022. ISSN 0004-6361. DOI: 10.1051/0004-6361/202243617.

HALE, G. E.; ELLERMAN, F.; NICHOLSON, S. B.; JOY, A. H. The Magnetic Polarity of Sun-Spots. **The Astrophysical Journal**, v. 49, n. 26, p. 153, 1919. ISSN 0004-637X. DOI: 10.1086/142452.

HANSSEN, A. W.; KUIPERS, W. J. A. **On the relationship between the frequency of rain and various meteorological parameters.(With reference to the problem of objective forecasting).** Koninklijk: Nederlands Meteorologisch Instituut, 1965.

HATHAWAY, D. H. The Solar Cycle. **Living Reviews in Solar Physics**, Max Planck Institute for Solar System Research, v. 12, p. 4, 1 Dec. 2015. ISSN 2367-3648. DOI: 10.1007/lrsp-2015-4.

HAUTALUOMA, G. et al. **Solar Cycle 25 Is Here. NASA, NOAA Scientists Explain What That Means**. 2020. Available from: <https://www.nasa.gov/press-release/solar-cycle-25-is-here-nasa-noaa-scientists-explain-what-that-means>. Visited on: 10 Nov. 2021.

HAYKIN, S. **Redes Neurais: Princípios e Prática**. Porto Alegre: Bookman Editora, 2001. P. 898. ISBN 9788577800865.

HE, K.; GKIOXARI, G.; DOLLÁR, P.; GIRSHICK, R. Mask R-CNN. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 42, n. 2, p. 386–397, Mar. 2020. ISSN 19393539. DOI: 10.1109/TPAMI.2018.2844175.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep Residual Learning for Image Recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas: IEEE, 2016. P. 770–778. DOI: 10.1109/CVPR.2016.90.

HEIDKE, P. Calculation of the success and goodness of strong wind forecasts in the storm warning service. **Geogr. Ann. Stockholm**, v. 8, n. 1926, p. 301–349, 1926.

HELIOVIEWER.ORG. **Solar and heliospheric image visualization tool**. 2021. Available from: <https://helioviewer.ias.u-psud.fr/>. Visited on: 10 Nov. 2021.

HENTZAU. **How Is Sunspot Formed?** The Scientific Gamer. 2012. Available from: <https://scientificgamer.com/how-is-sunspot-formed/>. Visited on: 9 July 2024.

HIGGINS, P.; GALLAGHER, P.; MCATEER, R.; BLOOMFIELD, D. Solar magnetic feature detection and tracking for space weather monitoring. **Advances in Space Research**, v. 47, n. 12, p. 2105–2117, June 2011. ISSN 02731177. DOI: 10.1016/j.asr.2010.06.024.

HOWARD, R.; LABONTE, B. J. The sun is observed to be a torsional oscillator with a period of 11 years. **The Astrophysical Journal**, v. 239, p. l33–l36, July 1980. DOI: 10.1086/183286.

HUANG, G.; LIU, Z.; VAN DER MAATEN, L.; WEINBERGER, K. Q. Densely Connected Convolutional Networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Los Alamitos, CA, USA: IEEE Computer Society, July 2017. P. 2261–2269.

HUANG, X.; WANG, H.; XU, L.; LIU, J., et al. Deep Learning Based Solar Flare Forecasting Model. I. Results for Line-of-sight Magnetograms. **The Astrophysical Journal**, IOP Publishing, v. 856, n. 1, p. 7, 2018. ISSN 1538-4357. DOI: 10.3847/1538-4357/aaae00.

HUANG, X.; WANG, H.; XU, L.; SUN, W. Learning Solar Flare Forecasting Model from Magnetograms. In: IEEE Visual Communications and Image Processing. New York, USA: IEEE, Dec. 2017. 2018-Janua, p. 1–4. ISBN 978-1-5386-0462-5. DOI: 10.1109/VCIP.2017.8305095.

IOFFE, S.; SZEGEDY, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In: BACH, F.; BLEI, D. (Eds.). **Proceedings of the 32nd International Conference on Machine Learning**. Lille, France: PMLR, July 2015. v. 37. (Proceedings of Machine Learning Research), p. 448–456. Available from: <https://proceedings.mlr.press/v37/ioffe15.html>.

JAPKOWICZ, N.; STEPHEN, S. The Class Imbalance Problem: A Systematic Study. **Intell. Data Anal.**, v. 6, p. 429–449, Nov. 2002. DOI: 10.3233/IDA-2002-6504.

KANEDA, K. et al. Flare Transformer: Solar Flare Prediction Using Magnetograms and Sunspot Physical Features. In: PROCEEDINGS of Computer Vision – ACCV 2022. Cham: Springer Nature Switzerland, 2023. P. 442–457. DOI: 10.1007/978-3-031-26284-5_27.

KAY, W. et al. The Kinetics Human Action Video Dataset. **CoRR**, 2017. Available from: <https://api.semanticscholar.org/CorpusID:27300853>. Visited on: 9 July 2024.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. ImageNet Classification with Deep Convolutional Neural Networks. **Communications of the ACM**, ACM, New York, NY, USA, v. 60, n. 6, p. 84–90, May 2017. ISSN 0001-0782. DOI: 10.1145/3065386.

LANG, K. R. **Sun, Earth and Sky**. Ed. by Kenneth R. Lang. New York, NY: Springer New York, 2006. P. 284. ISBN 978-0-387-30456-4. DOI: 10.1007/978-0-387-33365-6.

LANZEROTTI, L. J. Space Weather: Historical and Contemporary Perspectives. **Space Science Reviews**, The Author(s), v. 212, n. 3-4, p. 1253–1270, Sept. 2017. ISSN 0038-6308. DOI: 10.1007/s11214-017-0408-y.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **Nature**, v. 521, n. 7553, p. 436–444, May 2015. ISSN 0028-0836. DOI: 10.1038/nature14539.

LEKA, K. D. et al. A Comparison of Flare Forecasting Methods. III. Systematic Behaviors of Operational Solar Flare Forecasting Systems. **The Astrophysical Journal**, v. 881, 2 2019. ISSN 1538-4357. DOI: 10.3847/1538-4357/ab2e11.

LI, X.; ZHENG, Y.; WANG, X.; WANG, L. Predicting Solar Flares Using a Novel Deep Convolutional Neural Network. **The Astrophysical Journal**, IOP Publishing, v. 891, n. 1, p. 10, 2020. ISSN 1538-4357. DOI: 10.3847/1538-4357/ab6d04.

LI, Y. et al. MViTv2: Improved Multiscale Vision Transformers for Classification and Detection. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). New Orleans, LA, USA: IEEE, June 2022. P. 4804–4814. DOI: 10.1109/CVPR52688.2022.00476.

LIN, T.-Y. et al. Microsoft COCO: Common Objects in Context. In: COMPUTER Vision – ECCV 2014. Cham: Springer International Publishing, 2014. P. 740–755. DOI: 10.1007/978-3-319-10602-1_48.

LIU, C.; DENG, N.; WANG, J. T. L.; WANG, H. Predicting Solar Flares Using SDO /HMI Vector Magnetic Data Products and the Random Forest Algorithm. **The Astrophysical Journal**, IOP Publishing, v. 843, n. 2, p. 104, July 2017. ISSN 1538-4357. DOI: 10.3847/1538-4357/aa789b.

LOTUFO, R. d. A. **RegressaoLinearOtimizacao.png**. 2019. Available from: <https://github.com/robertoalotufo/files/blob/master/figures/RegressaoLinear_Otimizacao.png>. Visited on: 12 Nov. 2021.

MACHOL, J.; VIERECK, R.; PECK, C.; III, J. M. **GOES X-ray Sensor (XRS) Operational Data**. [S.l.: s.n.], June 2022. P. 1–15. Available from: <www.ngdc.noaa.gov/stp/satellite/goes-r.html.>.

MCATEER, R. T. J.; GALLAGHER, P. T.; CONLON, P. A. Turbulence, complexity, and solar flares. **Advances in Space Research**, v. 45, n. 9, p. 1067–1074, 2010. ISSN 0273-1177. DOI: 10.1016/j.asr.2009.08.026.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The Bulletin of Mathematical Biophysics**, v. 5, p. 115–133, 4 Dec. 1943. ISSN 0007-4985. DOI: 10.1007/BF02478259.

MCINTOSH, P. S. The classification of sunspot groups. **Solar Physics**, v. 125, n. 2, p. 251–267, 1990. ISSN 00380938. DOI: 10.1007/BF00158405.

MESSEROTTI, M. et al. Solar weather event modelling and prediction. **Space Science Reviews**, v. 147, n. 3-4, p. 121–185, 2009. ISSN 00386308. DOI: 10.1007/s11214-009-9574-x.

MINAEE, S. et al. Image Segmentation Using Deep Learning: A Survey. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, IEEE, p. 1–22, Feb. 2021. ISSN 19393539. DOI: 10.1109/TPAMI.2021.3059968.

MINSKY, M.; PAPERT, S. **Perceptrons: an Introduction to Computational Geometry**. Massachusetts: MIT Press, 1969. ISBN 9780262630221. Available from: <https://books.google.com.br/books?id=Ow1OAQAAIAAJ>.

MINSKY, M. L. **Computation**. Englewood Cliffs: Prentice-Hall, 1967.

MURANUSHI, T. et al. UFCORIN: A fully automated predictor of solar flares in GOES X-ray flux. **Space Weather**, v. 13, n. 11, p. 778–796, Nov. 2015. ISSN 1542-7390. DOI: 10.1002/2015SW001257.

NASA. **Space Weather**. 2024. Available from: <https://science.nasa.gov/heliophysics/focus-areas/space-weather/>. Visited on: 11 Apr. 2024.

NASA. **The Sun**. 2013. Available from: <https://www.nasa.gov/image-article/sun/>. Visited on: 6 Apr. 2024.

NASA GODDARD SPACE FLIGHT CENTER. **A Powerful Sequence of Flares Start September 2017**. 2017. Available from: <https://svs.gsfc.nasa.gov/12706#X9.3>. Visited on: 11 Nov. 2021.

NASA SOLAR DYNAMICS OBSERVATORY. **NASA's SDO Observes Largest Sunspot of the Solar Cycle**. 2014. Available from: <https://web.archive.org/web/20210718234919/https://www.nasa.gov/content/goddard/sdo-observes-largest-sunspot-of-the-solar-cycle/>. Visited on: 19 Sept. 2024.

NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION. **GOES X-Ray Flux**. 2017. Available from: <ftp://ftp.swpc.noaa.gov/pub/warehouse/2017/2017_plots/xray/20170907_xray.gif>. Visited on: 11 Nov. 2021.

NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION. **Solar Cycle Progression**. 2024. Available from: <https://www.swpc.noaa.gov/products/solar-cycle-progression>. Visited on: 10 May 2024.

NATIONAL SOLAR OBSERVATORY. **First Image from NSF's Inouye Solar Telescope!** 2021. Available from: <https://nso.edu/telescopes/dkist/first-light-full-image/>. Visited on: 9 Nov. 2021.

NISHIZUKA, N.; SUGIURA, K.; KUBO, Y.; DEN, M.; ISHII, M. Deep Flare Net (DeFN) Model for Solar Flare Prediction. **The Astrophysical Journal**, v. 858, n. 2, 2018. ISSN 1538-4357. DOI: 10.3847/1538-4357/aab9a7.

NISHIZUKA, N.; SUGIURA, K.; KUBO, Y.; DEN, M.; WATARI, S., et al. Solar Flare Prediction Model with Three Machine-learning Algorithms using Ultraviolet Brightening and Vector Magnetograms. **The Astrophysical Journal**, IOP Publishing, v. 835, n. 2, p. 156, Jan. 2017. ISSN 1538-4357. DOI: 10.3847/1538-4357/835/2/156.

OCEANIC, N.; ADMINISTRATION, A. **Solar Flares (Radio Blackouts)**. 2024. Available from: <https://www.swpc.noaa.gov/phenomena/solar-flares-radio-blackouts>. Visited on: 27 Sept. 2024.

OCEANIC, N.; CENTER, A. A. S. W. P. **NOAA Space Weather Scales**. 2024. Available from: <https://www.swpc.noaa.gov/noaa-scales-explanation>. Visited on: 27 Sept. 2024.

OLIVEIRA, L. S.; GRADVOHL, A. L. S. Automatic analysis of magnetograms for identification and classification of active regions using Deep Learning. **Revista Brasileira de Computação Aplicada**, v. 12, p. 67–79, 2020. DOI: 10.5335/rbca.v12i2.10531.

PARK, E. et al. Application of the Deep Convolutional Neural Network to the Forecast of Solar Flare Occurrence Using Full-disk Solar Magnetograms. **The Astrophysical Journal**, IOP Publishing, v. 869, n. 2, p. 91, 2018. ISSN 1538-4357. DOI: 10.3847/1538-4357/aaed40.

PARKER, E. N. Hydromagnetic Dynamo Models. **The Astrophysical Journal**, v. 122, p. 293, Sept. 1955. DOI: 10.1086/146087.

RADFORD, A.; METZ, L.; CHINTALA, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. **4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings**, p. 1–16, Nov. 2015. arXiv: 1511.06434.

RAWAT, W.; WANG, Z. Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. **Neural Computation**, v. 29, n. 9, p. 2352–2449, Sept. 2017. ISSN 0899-7667. DOI: 10.1162/neco_a_00990.

REDMON, J.; FARHADI, A. YOLOv3: An Incremental Improvement, Apr. 2018. arXiv: 1804.02767.

REDMON, R. J. et al. September 2017's Geoeffective Space Weather and Impacts to Caribbean Radio Communications During Hurricane Response. **Space Weather**, v. 16, n. 9, p. 1190–1201, 2018. DOI: https://doi.org/10.1029/2018SW001897.

RIBEIRO, F. **Técnicas de aprendizado de máquina aplicadas à previsão de explosões solares**. 2020. S. 138. Dissertação de Mestrado – Universidade Estadual de Campinas, Limeira, SP. DOI: 10.47749/T/UNICAMP.2020.1149462.

RIBEIRO, F.; GRADVOHL, A. L. S. Machine learning techniques applied to solar flares forecasting. **Astronomy and Computing**, v. 35, p. 100468, Apr. 2021. ISSN 22131337. DOI: 10.1016/j.ascom.2021.100468.

RONNEBERGER, O.; FISCHER, P.; BROX, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. **IEEE Access**, v. 9, p. 16591–16603, May 2015. ISSN 2169-3536. arXiv: 1505.04597.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. **Psychological Review**, v. 65, p. 386–408, 6 1958. ISSN 1939-1471. DOI: 10.1037/h0042519.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. **Nature**, v. 323, p. 533–536, 6088 Oct. 1986. ISSN 0028-0836. DOI: 10.1038/323533a0.

SCHOU, J. et al. Design and Ground Calibration of the Helioseismic and Magnetic Imager (HMI) Instrument on the Solar Dynamics Observatory (SDO). **Solar Physics**, v. 275, n. 1-2, p. 229–259, Jan. 2012. ISSN 0038-0938. DOI: 10.1007/s11207-011-9842-2.

SHIBATA, K.; MAGARA, T. Solar Flares: Magnetohydrodynamic Processes. **Living Reviews in Solar Physics**, v. 8, 2011. ISSN 1614-4961. DOI: 10.12942/lrsp-2011-6.

SILVA, C. W. da. **Uma nova abordagem para a extração de características das linhas do campo magnético da coroa solar utilizando a transformada de Hough**. 2013. S. 159. PhD thesis – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, SP.

SIMONYAN, K.; ZISSERMAN, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In: p. 1–14. arXiv: 1409.1556.

SUN, P. et al. Solar Flare Forecast Using 3D Convolutional Neural Networks. **The Astrophysical Journal**, v. 941, p. 1, 1 Dec. 2022. ISSN 0004-637X. DOI: 10.3847/1538-4357/ac9e53.

SZEGEDY, C. et al. Going Deeper with Convolutions. In: IEEE Conference on Computer Vision and Pattern Recognition. Boston: IEEE, 2014. P. 1–9. ISBN 978-1-4673-6964-0. DOI: 10.1109/CVPR.2015.7298594.

TAN, M.; LE, Q. V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In: CHAUDHURI, K.; SALAKHUTDINOV, R. (Eds.). **36th International Conference on Machine Learning**. [S.l.]: PMLR, 2019. v. 97. (Proceedings of Machine Learning Research), p. 6105–6114.

TANG, R.; LIAO, W., et al. Solar Flare Prediction Based on the Fusion of Multiple Deep-learning Models. **The Astrophysical Journal Supplement Series**, IOP Publishing, v. 257, n. 2, p. 50, Dec. 2021. ISSN 0067-0049. DOI: 10.3847/1538-4365/ac249e.

TANG, R.; ZENG, F., et al. The Comparison of Predicting Storm-Time Ionospheric TEC by Three Methods: ARIMA, LSTM, and Seq2Seq. **Atmosphere**, v. 11, n. 4, p. 316, Mar. 2020. ISSN 2073-4433. DOI: 10.3390/atmos11040316.

TIANCHI. **Solar Flare Forecasting Dataset**. 2020. Available from: <https://tianchi.aliyun.com/dataset/dataDetail?dataId=74780>. Visited on: 11 June 2024.

TRAN, D. et al. A Closer Look at Spatiotemporal Convolutions for Action Recognition. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, June 2018. P. 6450–6459. ISBN 978-1-5386-6420-9. DOI: 10.1109/CVPR.2018.00675.

VASWANI, A. et al. Attention Is All You Need. In: GUYON, I. et al. (Eds.). **Advances in Neural Information Processing Systems**. Long Beach: Curran Associates, Inc., 2017. v. 30.

WANG, X. et al. Predicting Solar Flares with Machine Learning: Investigating Solar Cycle Dependence. **The Astrophysical Journal**, IOP Publishing, v. 895, n. 1, p. 3, May 2020. ISSN 1538-4357. DOI: 10.3847/1538-4357/ab89ac.

WOODCOCK, F. The Evaluation of Yes/No Forecasts for Scientific and Administrative Purposes. **Monthly Weather Review**, v. 104, n. 10, p. 1209–1214, Oct. 1976. ISSN 0027-0644. DOI: 10.1175/1520-0493(1976)104<1209:TEOYFF>2.0.CO;2.

WU, Z. et al. Modeling Spatial-Temporal Clues in a Hybrid Deep Learning Framework for Video Classification. In: 11. 23RD ACM international conference on Multimedia. New York: ACM, Oct. 2015. v. 20, p. 461–470. ISBN 9781450334594. DOI: 10.1145/2733373.2806222.

YI, K.; MOON, Y.-J.; SHIN, G.; LIM, D. Forecast of Major Solar X-Ray Flare Flux Profiles Using Novel Deep Learning Models. **The Astrophysical Journal**, IOP Publishing, Bristol, v. 890, n. 1, p. l5, Feb. 2020. ISSN 2041-8213. DOI: 10.3847/2041-8213/ab701b.

YOUDEN, W. J. Index for rating diagnostic tests. **Cancer**, v. 3, n. 1, p. 32–35, 1950. ISSN 0008-543X. DOI: 10.1002/1097-0142(1950)3:1<32::AID-CNCR2820030106>3.0.CO;2-3.

YU, D.; HUANG, X.; WANG, H.; CUI, Y. Short-Term Solar Flare Prediction Using a Sequential Supervised Learning Method. **Solar Physics**, v. 255, n. 1, p. 91–105, Mar. 2009. ISSN 0038-0938. DOI: 10.1007/s11207-009-9318-9.

ZHENG, Y.; LI, X.; WANG, X. Solar Flare Prediction with the Hybrid Deep Convolutional Neural Network. **The Astrophysical Journal**, IOP Publishing, v. 885, n. 1, p. 73, Nov. 2019. ISSN 0004-637X. DOI: 10.3847/1538-4357/ab46bd.

Appendix A

Metrics to evaluate the model's performances

This appendix presents the correlated performance metrics referenced in this thesis and the work discussed in Chapter 3. Most of these works address binary classification problems, so we calculate all metrics from a Confusion Matrix (or Contingency Table).

The Confusion Matrix presents the number of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) relative to a set of samples with known outputs y and predicted outputs $\hat{y}$ as illustrated in Table A.1, in which the occurrence of a solar flare is considered a positive event (1), and the non-occurrence of a negative event (0).

Table A.1: Outputs of a confusion matrix.

Real Output	Predicted Output	
	$\hat{y} = 1$	$\hat{y} = 0$
$y = 1$	TP	FN
$y = 0$	FP	TN

From the results obtained in the Confusion Matrix, we can calculate several metrics, starting with Accuracy (ACC), one of the best-known and most popular performance metrics for classifiers, defined in Equation A.1.

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{TN} + \mathrm{FN}} \tag{A.1}$$

ACC calculates the percentage of correctly classified samples concerning the total number of samples. However, ACC is not a good performance metric for solar flare forecasting, usually characterized as an unbalanced classification problem (BOBRA; COUVIDAT, 2015). In this scenario, a classifier that misses all minority predictions – the positive events – can still present a high accuracy (RIBEIRO; GRADVOHL, 2021), depending on the imbalance ratio.

Therefore, for unbalanced classification problems, it is crucial to emphasize metrics that focus on the positive class. The first of them is the Probability of Detection, also known as Recall (POD / Recall), defined in Equation A.2, which is also called True Positive Rate (ABED, A. K.; QAHWAJI; ABED, A., 2021), Recall or Sensitivity (BOBRA; COUVIDAT, 2015).

$$\mathrm{POD} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{A.2}$$

POD represents the ratio between correctly predicted positive samples (TP) and the total number of positive samples. Its range varies between 0 and 1, with 1 being the optimal value. On the other hand, False Alarm Ratio (FAR), defined in Equation A.3, measures the ratio between incorrectly predicted positive samples (FP) concerning the total positive predictions with an interval also from 0 to 1, with 0 being the optimal value (BARNES, L. R. et al., 2009).

$$\mathrm{FAR} = \frac{\mathrm{FP}}{\mathrm{TP} + \mathrm{FP}} \tag{A.3}$$

The Critical Success Index (CSI) is more balanced than previously discussed metrics once it includes the FAR in its formula (ABED, A. K.; QAHWAJI; ABED, A., 2021; PARK et al., 2018). Equation A.4 describes the CSI. Its range also varies from 0 to 1, with 1 being the optimal value, as well as POD and ACC.

$$\mathrm{CSI} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{A.4}$$

Some authors (ZHENG; LI, X.; WANG, X., 2019; LI, X. et al., 2020; DENG et al., 2021) consider Precision (PRE) instead of CSI. The PRE represents the proportion of correctly predicted samples concerning all predictions given as positive, as defined in Equation A.5.

$$\mathrm{PRE} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{A.5}$$

However, although POD, FAR, and CSI give greater weight to correctly classified positive events, the most effective method for evaluating the performance of an unbalanced classifier is using a skill score (BOBRA; COUVIDAT, 2015). The most used in solar flare forecasting works is the Heidke Skill Score (HSS) (HEIDKE, 1926; WOODCOCK, 1976), defined in Equation A.6.

$$\mathrm{HSS} = \frac{2 \times (\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN})}{(\mathrm{TP} + \mathrm{FN}) \times (\mathrm{FN} + \mathrm{TN}) + (\mathrm{TP} + \mathrm{FP}) \times (\mathrm{FP} + \mathrm{TN})} \tag{A.6}$$

The HSS score range is from $-\infty$ to 1, with 1 being the optimal value. Values close to zero indicate that the predictions were produced mainly by chance (ABED, A. K.; QAHWAJI; ABED, A., 2021), and values less than zero do not represent any forecasting skill (ZHENG; LI, X.; WANG, X., 2019). Although HSS is the most used in solar flare forecasting works, it is still influenced by the samples' unbalanced ratio.

Finally, Bloomfield et al. (2012) formed the consensus that the True Skill Statistic (TSS), also known as Hanssen and Kuipers (1965) index or Youden (1950) index, is the most suitable metric for solar flare forecasting classifiers. The reason is that the class imbalance ratio has less influence on the TSS than on the other metrics (WOODCOCK, 1976). Bobra and Couvidat (2015) also corroborate this consensus. Since their work, TSS has been considered the primary performance metric for solar flare forecasting classifiers. Equation A.7 defines the TSS.

$$\mathrm{TSS} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} - \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \tag{A.7}$$

The TSS range varies from -1 to 1, with 1 being the optimal value. As in HSS, values close to zero indicate that the hits are random, and negative values indicate no prediction competence.

Some works also consider the F1 Score, defined as the harmonic mean between PRE and POD, according to Equation A.8. The F1 range is from 0 to 1, with 1 being the optimal value.

$$\mathrm{F1} = \frac{2 \times \mathrm{PRE} \times \mathrm{POD}}{\mathrm{PRE} + \mathrm{POD}} \tag{A.8}$$

Appendix B

Operational Challenges Faced

We started our research training and testing our models on our dedicated desktop, described in Section 5.3.1, where we get the inference testing times. However, as we progressed with the experiments' execution, the GPU of our desktop started experiencing constant RAM overflows when training our models, especially when we decided to use MViTs-based models. This situation forced us to change to a more robust hardware environment.

Thus, we requested an account on the Centro Nacional de Processamento de Alto Desempenho em São Paulo (CENAPAD-SP) to train the models in the high-performance computing Lovelace environment. This computational environment was much more powerful, and we described its configuration and queue parameters in Section 4.4.2.

As we already mentioned, although we adapted our models to run in queues with one GPU ("umagpu" queue) and two GPUs ("duasgpus" queue), we ran most experiments with only one GPU because the queue with 2 GPUs was much more congested, with their jobs taking twice as long or more time to start execution than in the queue with one GPU.

Besides PyTorch and PyTorch Lightning, the following list summarizes other software we initially used in our research. However, we need to exclude from our models due to the CENAPAD-SP hardware environment limitations:

- Jupyter Notebook[1]: We started developing our experiments in the interactive computing format of Jupyter Labs (`.ipynb` files).
- Neptune[2]: Enables real-time metrics monitoring during model training, validation, and testing. It also allows the monitoring of hardware resources.

[1] Available at `https://jupyter.org`

[2] Available at `https://neptune.ai`

To run our models on the CENAPAD-SP Lovelace environment, we needed to port our model codes from Jupyter Lab format (`.ipynb`) to pure Python code (`.py`) and disable Neptune live logging because Lovelace's computing nodes did not support interactive execution and did not provide Internet communication. Thus, we submitted the Python codes of our Solar Flare MViT (SF_MViT) models as jobs through Lovelace's queuing system.

This particularity was time-consuming since we lost instant access to the online logs at runtime. We experienced long response times regarding the convergence and effectiveness of the models' performance compared to our previous interactive environment with Jupyter and Neptune on our dedicated GPU desktop.