Comparative Analysis of Attention-Based Deep Learning Models for Solar Flare Prediction

Shaik Sattar Saif

Abstract—**Solare flares are intense burst of radiations from the sun that can disrupt radio communications and compromise power grids on Earth. Accurately predicting these events remains a challenge due to complexity of solare magnetic field structures and the extreme class imbalance inherent in solar datasets. In this project, we have engineered a machine learning approach to forecast solar flares using a Time Series Dataset of Solar Magnetic Field Parameters obtained from IEEE BigData2019 Kaggle. We implemented a comprehensive pipline including data ingestion, preprocessing, feature engineering and data augmentation followed by a two-stage modeling approach. Initially, a baseline Machine Learning classifier (Logistic Regression, SVM, Random Forest) was selected for both binary and multi class scenarios. Subsequently, we engineered advanced deep learning architectures, specifically Uni-directional and Bi-directional RNNs and LSTMs. Followed, by integrating them with distinct attention mechanisms such as Concatenation, Dot product and Multi-head to focus on crucial temporal features. Experimental results indicate that Bi-directional LSTM with Dot product attention mechanism. outperforming all the models that we have experimented with.**

I. Introduction

Solar activity significantly influences the near Earth environment, this phenomenon collectively is known as space weather. Among the most energetic events are solar flares, which are sudden intense flashes of brightness observed near the Sun's surface. These events are triggered by magnetic reconnection in solar Active Regions (ARs) and can release up to 10^{32} ergs of energy. The impacts of major flares (M and X-class) are profound, and capable of disrupting high frequency radio communications, degrading Global Positioning System (GPS) accuracy, and inducing geomagnetic currents that threaten power grids [9]. Solar flares are primarily classified according to their X-ray brightness in the wavelength ranging from 1 to 8 Angstroms, as measured by the GOES (Geostationary Operational Environmental Satellite). The classification system follows a logarithmic scale with five major categories: A, B, C, M, and X. Each category represents a ten-fold increase in peak flux compared to the preceding one. Therefore, developing accurate and reliable forecasting models is critical for mitigating these risks.

Predicting these events remains a formidable challenge despite having high-resolution data from the Solar Dynamics Observatory (SDO) [1]. The primary difficulties are the complex non-linear evolution of magnetic fields and the extreme class imbalance, where quiet periods vastly outnumber severe flare events. Traditional Machine Learning (ML) classifiers, such as SVMs, often treat samples as static snapshots, neglecting inherent temporal dependencies [1]. While Long Short-Term Memory (LSTM) networks address this by modeling sequential data [8], standard architectures may still struggle to isolate specific predictive time steps within a large window.

In this paper, we present a comprehensive comparative study of solar flare prediction using the IEEE BigData 2019 Kaggle dataset [2]. We propose a robust pipeline that moves beyond baseline classification to advanced sequence modeling. Our primary contributions are as follows:

- Implementation of a complete preprocessing pipeline including data augmentation to mitigate the class imbalance problem.
- Evaluation of baseline machine learning models (Logistic Regression, SVM, Random Forest) for both binary and multi-class classification tasks.
- Development of advanced Deep Learning architectures (Uni- and Bi-directional LSTMs) integrated with multiple attention mechanisms (Concatenation, Dot-product, and Multi-head).
- Demonstration that Bi-directional LSTMs with Dot-product attention provide superior predictive skill (TSS) by effectively focusing on critical precursor signals in the magnetic field time series.

Scope Refinement

Note that the initial proposal for this project included using solar images (SDO/HMI) alongside the magnetic data. However, during the analysis, we discovered a critical issue: the magnetic dataset contains relative time steps (0 to 59) but lacks the specific dates and times required to match them with the corresponding images. Without these timestamps, it is impossible to correctly align the images with the magnetic readings. Due to this technical limitation and time constraints, this report focuses exclusively on analyzing the time-series magnetic parameters using Deep Learning.

II. DATASET AND PREPROCESSING

The framework that are proposed in this paper utilizes the solar magnetic field dataset provided by the IEEE Big Data Cup 2019, derived from the Solar Dynamics Observatory (SDO) Helioseismic and Magnetic Imager (HMI). This section outlines the data acquisition, the resolution of critical integrity issues within the source files, and the feature engineering pipeline employed to prepare the time-series data for deep learning models.

A. Dataset Description

The dataset consists of multivariate time-series data representing Solar Active Regions (ARs). Collectively amounting to approximately 17 GB, the data is distributed across three training partitions. Each sample is identified by a unique *record_id* and contains a temporal sequence of 60 steps, representing the evolution of physical parameters (e.g., Total Magnetic Flux, Magnetic Shear) over a specific observation window. The target variable is the flare intensity class (Q, B, C, M, X) associated with the AR.

Data Limitation: It is important to note that the IEEE BigData 2019 dataset provides magnetic parameters as anonymous time sequences. The data lacks absolute timestamps (e.g., ”2014-05-01 12:00:00”). Since solar images are indexed by specific dates and times, we could not merge the two datasets for this study. Therefore, the SDO image dataset was excluded, and our work relies solely on the tabular magnetic features.

B. Data Ingestion and Optimization

The original dataset consisted of large JSON files totaling approximately 17 GB. This was too big to load into the computer's memory (RAM) all at once. To fix this, we used a library called **Dask** [5], [6]. Dask allows us to process the data in small chunks instead of trying to open the whole file at the same time.

We also converted the data from the slow JSON format into **Parquet** format [7]. Parquet is a special file type that compresses the data, making it much smaller and faster to read. This step allowed us to clean and process the massive dataset efficiently without crashing the system.

C. Data Integrity Correction

During the initial exploratory data analysis, a severe data integrity issue was identified involving *record_id* collisions across the distributed partition files. Approximately 40,000 unique time-series observations were incorrectly assigned duplicate identifiers across disjoint files. This corruption caused distinct temporal sequences to merge during aggregation, resulting in erroneous records with 120 time steps instead of the standardized 60. Crucially, this corruption disproportionately affected the minority classes; over 35% of the rare X-class and M-class flares were compromised. To resolve this, we implemented a stateful ”Rename on Collision” strategy. The preprocessing pipeline tracks global IDs and assigns a unique, sequential integer to any subsequent occurrence of a preexisting ID. This restoration process ensured that all samples retained their correct temporal dimension (60 steps) and preserved vital minority class samples that would otherwise have been discarded.

D. Data Augmentation Strategy

Solar flare datasets exhibit extreme class imbalance. In the raw training partition, the ratio of majority (Q) to minority (X) class samples exceeded 450:1. The severity of the initial class imbalance is illustrated in Fig. 1, where the minority classes are barely visible compared to the quiet class.

To mitigate this bias, a three stage pipeline was applied.

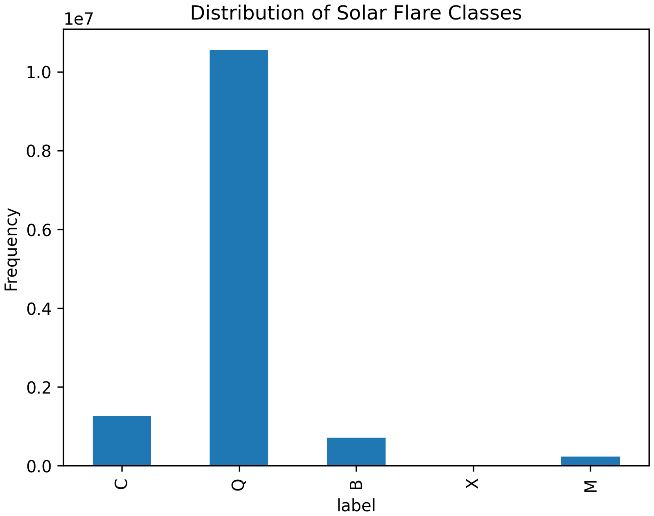


Fig. 1: Distribution of Solar Flare classes in the raw training set, highlighting the extreme class imbalance.

1) Synthetic Generation (X and M Class): For the rarest classes, simple duplication leads to overfitting. Instead, we generated synthetic samples using two time-series modification techniques:

- **Magnitude Scaling:** Simulates variations in peak intensity. A scalar $\alpha \sim \mathcal{U}[0.9, 1.1]$ is applied to the series: $\mathbf{x}'_t = \alpha \cdot \mathbf{x}_t$.
- **Magnitude Warping:** Simulates non-linear variations in flux evolution (e.g., faster rise times). A smooth curve $w(t)$ is generated via Cubic Splines and element-wise multiplied with the series: $\mathbf{x}'_t = \mathbf{x}_t \cdot w(t)$. This ensures the generated samples remain smooth and physically plausible, avoiding the unnatural spikes introduced by adding random noise.

2) Resampling Strategy: For intermediate classes (B, C), naive oversampling was used to increase representation. Conversely, the majority class (Q) was randomly undersampled by 60% to reduce dataset dominance. This combined strategy reduced the imbalance ratio from ∼450:1 to 9:1 (Table I).

This strategy reduced the imbalance ratio to approximately 9:1, allowing the model to learn discriminative features for high-energy events.

TABLE I: Class Distribution Before and After Augmentation

Class	Original Count	Augmented Count
Q	172,396	68,958
C	20,604	21,634
B	11,638	12,802
M	3,785	11,355
X	**377**	**7,917**
Total	**208,800**	**122,666**

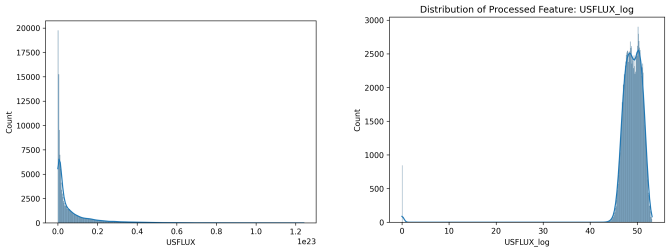
E. Feature Engineering

Raw solar magnetic parameters are often provided as static values per time step. However, solar flares are dynamic events driven by the evolution of magnetic energy over time. To allow our models to capture these trends, we implemented a feature engineering pipeline that transforms raw physical parameters into temporal descriptors.

1) Logarithmic Transformation: Key magnetic parameters, such as the total unsigned flux (USFLUX) and Total Photospheric Magnetic Energy (TOTPOT), exhibit a "heavy-tailed" distribution where extreme values are orders of magnitude larger than the mean. This skewness causes instability in gradient-based learning models. To normalize these distributions, we applied a log-transformation:

$$x' = \log(x + 1) \tag{1}$$

This compression allows the model to differentiate between small variations in weak active regions while still capturing the massive energy of X-class regions. Fig. 2 demonstrates the effect of the transformation, the raw distribution (Left) is heavily right-skewed, while the log-transformed feature (Right) approximates a normal distribution, stabilizing model training.



(a) Raw USFLUX (b) Log-Transformed

Fig. 2: Impact of Logarithmic Transformation on feature distribution.

2) Temporal Feature Extraction: Since a single snapshot of an Active Region is insufficient to predict an eruption, we generated features that describe the history and stability of the region over a sliding window. For every time step t, grouped by $record\_id$, and using standard time-series methods [3], [4], we computed:

- **Rolling Mean (Trend)**: A 5-step moving average to smooth out sensor noise and highlight the underlying energy trend.
- **Rolling Volatility (Instability)**: A 5-step moving standard deviation. A sudden spike in volatility often indicates magnetic flux emergence, a precursor to flaring.
- Rate of Change (Delta): The difference in one step $(x_t - x_{t-1})$. This measures how quickly energy builds up or dissipates.
- Lag Features (Memory): Values of 3 time steps prior (x_{t-3}). This explicitly provides the model with a short-term historical context.

3) Feature Selection: The raw dataset contained highly correlated features (multicollinearity). As shown in Fig. 3, several magnetic parameters exhibited high multicollinearity. For instance, Total Flux and Total Energy had a correlation coefficient ¿ 0.95, justifying the removal of redundant terms

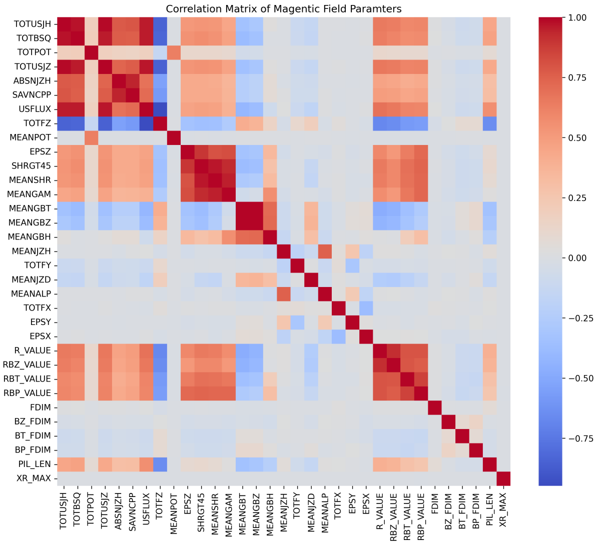


Fig. 3: Correlation Matrix of magnetic parameters. Darker regions indicate high redundancy, justifying feature selection.

Using correlation analysis, we removed redundant variables and retained only the most discriminative physical parameters (e.g. magnetic shear, polarity inversion line length). The final dataset consists of 14 engineered features, summarized in Table II.

TABLE II: Final Engineered Feature Set

Feature Group	Count	Physical Significance
Log-Transformed	3	USFLUX, TOTPOT, PIL_LEN. Represents the total magnetic energy and size of the region.
Raw Physical	4	MEANSHR, TOTFZ, EPSZ, R_VALUE. Represents magnetic shear and complexity near the polarity line.
Rolling Stats	6	Moving Average (Trend) and Standard Deviation (Volatility) calculated on Flux and Energy.
Temporal Delta	1	Rate-of-change for Total Potential Energy (TOTPOT).

III. PROPOSED METHODOLOGY

A. Mathematical Modeling

The solar flare prediction task is modeled as a time-series classification problem. Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$ represent a sequence of magnetic field parameters, where $T = 60$ is the sequence length and $\mathbf{x}_t \in \mathbb{R}^D$ represents the $D = 14$

features at time step t. The objective is to learn a mapping function $f : \mathbf{X} \rightarrow y$, where $y \in \{Q, C, M, X\}$ represents the maximum flare intensity within the forecast window. Note that B-class events were excluded from the dataset as they are too weak to impact Earth's technology, such as satellites, radio communications, or power grids.

B. Baseline Classifiers

To establish a performance benchmark, we implemented standard Machine Learning algorithms for both binary (Flare vs. No-Flare) and multi-class classification:

- **Logistic Regression:** A linear model serving as the simplest baseline.
- **Support Vector Machine (SVM):** Implemented with a Radial Basis Function (RBF) kernel to capture non-linear decision boundaries in the magnetic feature space.
- **Random Forest (RF):** An ensemble method used to handle feature interactions and provide robustness against overfitting.

C. Deep Learning Architectures

While baseline models treat the input as a static vector, deep recurrent architectures are designed to capture temporal dependencies. Recent reviews in time series classification [8] highlight the superiority of these models for sequential data.

1) Recurrent Neural Networks (RNN): We first implemented standard RNNs, which maintain a hidden state $\mathbf{h}_t$ to capture sequential information. However, standard RNNs suffer from the vanishing gradient problem, limiting their ability to learn long-term dependencies in the 60-step sequence.

2) Long Short-Term Memory (LSTM): To mitigate the vanishing gradient issue, we employed LSTM networks. The LSTM cell introduces a gating mechanism (Input, Forget, and Output gates) that regulates the flow of information, allowing the model to retain relevant magnetic history over longer durations [1].

3) Bi-Directional LSTM (Bi-LSTM): Standard LSTMs only process information in the forward direction ($t = 1 \rightarrow T$). However, the context of a magnetic fluctuation often depends on both past and future evolution within the window. We utilized Bi-LSTMs, which process the sequence in both forward and backward directions, concatenating the hidden states:

$$\mathbf{h}_t = [\overrightarrow{\mathbf{h}}_t; \overleftarrow{\mathbf{h}}_t] \tag{2}$$

This results in a richer representation of the magnetic field evolution.

D. Attention Mechanisms

In a standard LSTM, the final hidden state is often the bottleneck, as it must compress the entire sequence. To address this, we integrated Attention Mechanisms, which allow the model to "focus" on specific time steps (e.g., a sharp rise in flux) that are most predictive of a flare.

1) Dot-Product Attention: This mechanism computes a compatibility score between hidden states using a simple dot product. It is computationally efficient and effective for measuring global similarity.

$$\text{score}(\mathbf{h}_t, \mathbf{h}_s) = \mathbf{h}_t^\top \mathbf{h}_s \tag{3}$$

2) Concatenation (Additive) Attention: Here, the alignment score is learned through a feed-forward neural network, allowing the model to learn complex non-linear relationships between time steps.

3) Multi-Head Attention: Inspired by Transformer architectures, we implemented Multi-Head Attention. This allows the model to jointly attend to information from different representation subspaces at different positions. For example, one "head" might focus on Magnetic Shear volatility, while another focuses on Flux Magnitude changes.

E. Implementation Details

All models were implemented using PyTorch. To further address class imbalance, we employed a **Weighted Cross-Entropy Loss**:

$$L = -\sum_{c=1}^{C} w_c y_c \log(\hat{y}_c) \tag{4}$$

where w_c is the inverse class frequency. This penalizes the model more heavily for misclassifying rare events (X, M classes). We utilized the Adam optimizer with a learning rate of $1e-3$ and a batch size of 256.

IV. Experimental Results

In this section, we present the evaluation of the proposed solar flare prediction models. Given the extreme class imbalance, standard accuracy is an unreliable metric. Therefore, our primary evaluation criteria are the Matthews Correlation Coefficient (MCC) and Macro-averaged F1-Score, as these metrics provide a balanced measure of performance across all flare classes (Q, C, M, X).

A. Baseline Model Performance

We established baselines using Logistic/Softmax Regression and Random Forest classifiers. Table III summarizes their performance on the validation set.

TABLE III: Performance of Baseline Classifiers

Model	**Type**	**Accuracy**	**F1-Macro**	**MCC**
Logistic Regression	Binary	0.80	0.69	0.4756
Random Forest	Binary	0.94	0.82	0.6619
SVM	Binary	*Did not converge (Computational Limit)*		
Softmax Regression	Multi	0.90	0.44	0.4682
Random Forest	Multi	0.93	0.72	0.6190
SVM	Multi	*Did not converge (Computational Limit)*		

- **Linear Models**: Logistic Regression achieved a high Recall for the binary task but suffered from low Precision

TABLE IV: Comparison of Deep Learning Architectures (Sorted by MCC)

Model Configuration	Cell	Dir.	Attention	F1-Weighted	F1-Macro	MCC
LSTM_Bi_dot	**LSTM**	**Bi**	**Dot Product**	**0.9254**	**0.7845**	**0.7080**
LSTM_Bi_concat	LSTM	Bi	Concat	0.9085	0.7537	0.6634
LSTM_Uni_dot	LSTM	Uni	Dot Product	0.8675	0.6992	0.5587
LSTM_Uni_concat	LSTM	Uni	Concat	0.8345	0.5527	0.4674
LSTM_Uni_NoAtt	LSTM	Uni	None	0.8285	0.5684	0.4628
LSTM_Bi_NoAtt	LSTM	Bi	None	0.8261	0.4733	0.4318
RNN_Uni_NoAtt	RNN	Uni	None	0.8232	0.3956	0.4025
RNN_Bi_dot	RNN	Bi	Dot Product	0.8251	0.4626	0.3934
RNN_Uni_concat	RNN	Uni	Concat	0.8077	0.4516	0.3929

(0.36), indicating a high false-positive rate. In the multi-class setting, Softmax Regression failed completely to classify the minority X-class (F1-score of 0.00).

- **Random Forest**: This model significantly outperformed linear baselines, achieving a binary MCC of 0.6619. It successfully identified X-class flares with a precision of 0.98, though its recall remained moderate (0.50).
- **Support Vector Machines (SVM)**: While SVMs are powerful classifiers, they exhibit quadratic time complexity $O(n^2)$ with respect to the number of samples. Due to the large size of the augmented dataset ($\approx 123,000$ samples) and the high dimensionality of the feature space, SVM training failed to converge within practical computational limits.

B. Deep Learning Comparative Analysis

We performed an extensive ablation study to evaluate the impact of Cell Type (RNN vs. LSTM), Directionality (Uni vs. Bi), and Attention Mechanism (None, Dot, Concat). The results are presented in Table IV.

- **LSTM vs. RNN**: Across all configurations, LSTMs consistently outperformed standard RNNs. The best RNN model achieved an MCC of only 0.40, whereas LSTM variants consistently exceeded 0.60. This confirms that the gating mechanisms in LSTMs are essential for capturing long-term dependencies in the 60-step magnetic field sequences.
- **Impact of Directionality**: Bi-directional architectures provided a substantial boost over Uni-directional ones. For instance, `LSTM_Bi_dot` (MCC 0.708) significantly outperformed `LSTM_Uni_dot` (MCC 0.559), suggesting that the context of both past and future evolution within the observation window is critical.
- **Attention Mechanisms**: The integration of attention mechanisms improved performance by allowing the model to focus on specific temporal precursors. The Dot-Product Attention emerged as the most effective strategy, yielding the highest MCC (0.7080) and F1-Macro (0.7845) scores among all experimented models. Interestingly, the Concatenation attention, while effective, slightly underperformed compared to the simpler Dot-product approach.

C. Multi-Head Attention Analysis

We further evaluated Multi-Head Attention (MHA) architectures, hypothesized to capture diverse temporal features simultaneously. As shown in Table V, the best MHA configuration (2 Layers, 8 Heads) achieved an MCC of 0.6430. While this outperforms the baselines, it did not surpass the simpler `LSTM_Bi_dot` model. This suggests that the increased model complexity (evident in the training time of 67+ minutes) may have led to optimization difficulties or slight overfitting compared to the more direct Dot-product mechanism.

TABLE V: Multi-Head Attention (MHA) Performance

Model	Heads	Hidden	Time (min)	F1-Macro	MCC
MH_Bi_LSTM (L2)	8	128	67.2	0.7136	0.6430
MH_Bi_LSTM (L3)	4	256	243.3	0.6674	0.6084
MH_Bi_LSTM (L2)	4	256	173.5	0.5500	0.4653
MH_Bi_LSTM (L1)	8	128	46.9	0.4642	0.3991

V. CONCLUSION AND FUTURE WORK

In this work, we presented a robust deep learning framework for the short-term forecasting of solar flares, addressing the critical challenges of extreme class imbalance and non-linear magnetic field evolution. By implementing a comprehensive pipeline including ”Rename-on-Collision” data correction, magnitude warping augmentation, and log-transformed feature engineering we successfully stabilized the training distribution, reducing the imbalance ratio from 450:1 to 9:1.

Our experimental results demonstrate the limitations of static baseline classifiers (Logistic Regression, Random Forest) and the necessity of sequence modeling. The proposed Bi-directional LSTM with Dot-Product Attention emerged as the superior architecture, achieving a Matthews Correlation Coefficient (MCC) of 0.7080 and a Macro F1-score of 0.7845. This represents a significant improvement over standard Recurrent Neural Networks (MCC 0.40). Furthermore, the ablation study confirmed that the attention mechanism effectively allows the model to isolate critical temporal precursors within the 60-step observation window, filtering out noise from the quiet sun background.

Future Work

While the current approach utilizes handcrafted physical parameters, valuable spatial information is lost when condensing Active Regions into tabular data. Future iterations of this project will aim to:

- **Multimodal Alignment:** The main barrier to using solar images was the missing timestamp information in the provided CSV files. Future work will involve using pattern matching to recover the original dates or contacting the dataset authors. Once the timestamps are found, we can align the SDO images with the magnetic data to build a stronger Multimodal model (CNN + LSTM).
- **Operational Deployment**: Optimize the model for real-time inference to provide continuous probability forecasts for space weather monitoring systems.

REFERENCES

[1] R. A. Bobra and S. Couvidat, ”Solar flare prediction using SDO/HMI vector magnetic field data with a machine-learning algorithm,” *The Astrophysical Journal*, vol. 798, no. 2, p. 135, 2015.

[2] IEEE BigData Cup 2019, ”Solar Flare Prediction from Time Series of Magnetic Field Parameters,” [Online]. Available: https://dmlab.cs.gsu.edu/solar/

[3] R. Holla, ”Advanced feature engineering for time series data,” *Medium*, [Online]. Available: https://medium.com/@rahulholla1/advanced-feature-engineering-for-time-series-data-5f00e3a8ad29

[4] dotData Team, ”Practical Guide for Feature Engineering of Time Series Data,” *dotData Blog*, [Online]. Available: https://dotdata.com/blog/practical-guide-for-feature-engineering-of-time-series-data/

[5] Dask Developers, ”Dask Tutorial: Dataframes and Arrays,” [Online]. Available: https://tutorial.dask.org/

[6] Dask Blog, ”Dask for Parallel Computing: GroupBy, Chunk Sizes, and Performance,” [Online]. Available: https://blog.dask.org/

[7] M. Rocklin, ”Use Parquet for faster I/O,” *Dask Blog*, Jun. 2017. [Online]. Available: https://blog.dask.org/2017/06/28/use-parquet

[8] H. I. Fawaz, et al., ”Deep learning for time series classification: a review,” *arXiv preprint arXiv:2003.03878*, 2020. Available: https://arxiv.org/abs/2003.03878

[9] M. A. Author et al., ”Solar Flare Prediction using Machine Learning,” *Atmosphere*, vol. 15, no. 8, p. 930, 2024. [Online]. Available: https://www.mdpi.com/2073-4433/15/8/930