Solar Flare Prediction Using Machine Learning

Hussein Yehia, Mohamad Sharafeddine and Farid Abi Doumit

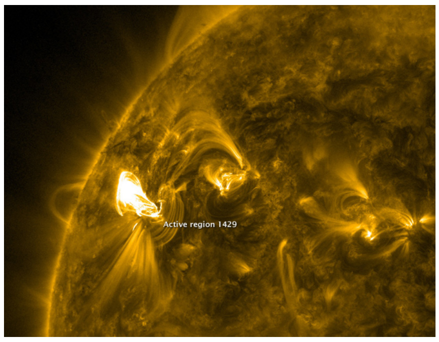
Introduction to Machine Learning, Dr. Joseph Bakarji
Maroun Semaan Faculty of Engineering and Architecture
American University of Beirut

Abstract

Solar flares are sudden releases of electromagnetic energy from the Sun's atmosphere that can disrupt satellites, radio communications, GPS accuracy, and pose radiation hazards to astronauts. Reliable prediction of flare intensity is therefore a key objective for operational space-weather forecasting. In this work, we use quantitative magnetic-field descriptors of solar active regions from NASA's SDO/HMI Active Region Magnetograms for Solar Flare Prediction dataset (2010–2018) to classify upcoming events into four classes: no flare (0), C, M, and X. The main challenge is the imbalance inherent in the dataset because of the infrequency of M and X class flares, which we aim to address by synthetic oversampling and data manipulation. We evaluate several models—Support Vector Machines, fully connected deep neural networks (including residual MLP variants), and tree-based ensembles—under different combinations of class weights and resampling strategies. Across these experiments, an XGBoost classifier trained with Full SMOTE achieves the best trade-off between balanced accuracy and macro-F1, substantially improving detection of M- and X-class flares compared to baseline models. However, performance on the rarest X-class remains limited by overfitting to synthetic samples. Our results confirm the strength of tree-based ensembles for flare-intensity prediction and highlight the importance of carefully controlled oversampling and feature selection when dealing with highly imbalanced heliophysics data.

Introduction

Solar flares are sudden eruptions of electromagnetic radiation that originate in magnetically active regions of the Sun's atmosphere—areas above where intense, twisted magnetic fields store large amounts of energy. When this energy is rapidly released, the resulting flare can emit immense radiation that severely disrupts satellite operation, radio communications, GPS-based navigation, and even power grids on Earth. In late November 2025, a software vulnerability exposed by an intense solar flare affected roughly 6,000 Airbus A320-family aircraft worldwide. A single solar flare event was suspected of corrupting flight-control data and triggering a sudden, uncommanded altitude loss on a commercial flight, forcing regulators and manufacturers to intervene at a global scale. Incidents of this kind highlight the direct operational risk for aviation, satellite-based services, and critical infrastructure that solar flares pose.



These phenomena can be classified based on their intensity, which is measured by peak X-ray brightness. The classes are divided into five categories (A, B, C, M, and X), with each letter representing a class that is 10 times more energetic than its predecessor. Since A- and B-class flares do not pose an imminent danger to our systems on Earth, most efforts focus on C-, M-, and X-class flares, which can significantly impact technological infrastructure.

Reliable, class-specific forecasts can provide early warning to satellite operators, power-grid managers, and airlines, allowing them to take precautionary measures such as reconfiguring orbits, adjusting loads, or rerouting traffic. Developing ML models that can accurately predict the intensity class of a flare before its occurrence in the next 24 hours would allow us to mitigate their disruptive effects. This challenge has been directly addressed in multiple recent studies, but most research utilizes images of the sun's active regions as data to predict the intensity of solar flares. For this reason, this work focuses on a novel approach, which has not been widely explored, where numerical data from the sun's active regions are utilized to predict the occurrence of solar flares in the next 24 hours. This type of data been exploited in a limited number of papers, such as Omitoyin et al., who utilized magnetic features different from those used in this work and achieved 79% accuracy using Random Forests. Goel et al. utilized SHARP parameters which are numeric values that describe magnetic properties of an active region, and compared different models, and concluded that Tree-based models performed the best.

Dataset

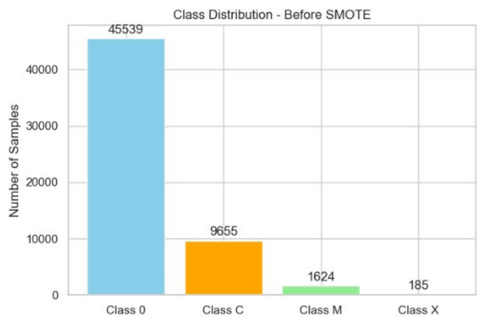
Source

The dataset used in this study is the Active Region Magnetograms for Solar Flare Prediction collection obtained from NASA's SDO/HMI instrument, covering observations from 2010 to 2018. It contains more than 75,000 labeled active-region instances, each represented by 29 quantitative magnetic features that characterize the geometry, complexity, and energetic structure of polarity inversion lines and surrounding magnetic fields. These features include magnetic-flux gradients and their statistics (mean, median, standard deviation, minimum, maximum, skewness, kurtosis), neutral-line properties such as length, fragmentation count, curvature metrics, and bending-energy descriptors, all of which capture the magnetic stress and topological complexity associated with flare productivity. Each instance is paired with a GOES flare classification recorded within a 24-hour forecasting window.

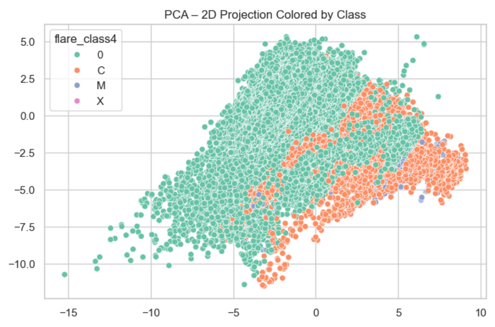
The GOES system categorizes solar flares according to their peak soft X-ray flux measured in the 1–8Å band. Flares are divided into classes (e.g., M2.3 or X1.0) where the first letter represents the classes (A, B, C, M and X) and the number to the right shows the subclass which is based on a logarithmic scale within each class to provide finer resolution of flare strength.

Pre-Processing

Initial inspection reveals a severe class imbalance, with non-flare events dominating the dataset and M- and X-class events representing an extremely small fraction.



Visualization through PCA, t-SNE, and UMAP revealed the complex, non-linearly separable nature of the flare classification problem by displaying complex class boundaries.

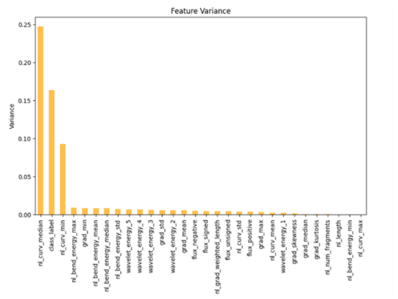


We performed a **stratified split** to ensure samples from all classes were included in train, test and validation sets.

Other preprocessing steps included:

- **Class relabeling**: GOES flare subclasses (e.g., C.1, C.7, M.2) were consolidated into four primary classes—0 (no flare), C, M, and X—to reduce fragmentation and strengthen class separability.
- **Variance-based feature removal**: Features exhibiting extremely low variance (e.g., near-constant gradient statistics) were discarded, as they contribute little information to model decisions.

$$\mathrm{Var}(x) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2,$$



- **Data Imbalance:** To alleviate class imbalance, we applied the Synthetic Minority Oversampling Technique (SMOTE), which generates synthetic minority samples by interpolating between an existing minority instance x_i and one of its k-nearest minority neighbours x_{zi}. The synthetic point is constructed as:

$$\boxed{x_{new} = x_i + \lambda\,(x_{zi} - x_i)}$$

$$with \qquad \lambda \sim \mathcal{U}(0,1)$$

Models and Evaluation Metrics

We consider three model families for solar flare classification: XGBoost, Support Vector Machines

(SVMs), and fully connected Neural Networks. Their performance is assessed using class-sensitive metrics—Precision, Recall, F1-score, and Balanced Accuracy—in order to quantify how well each model identifies the minority flare classes. We report balanced accuracy instead of plain accuracy, since standard accuracy is dominated by the majority (no-flare and C-class) samples and can therefore appear high even when the model performs poorly on the rarer M- and X-class flares.

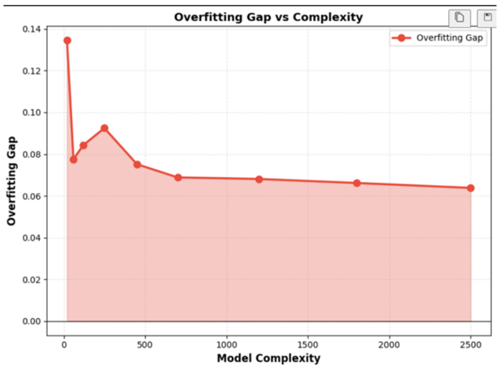
- **Precision**:
$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}.$$
- **Recall**:
$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}.$$
- **F1-score**:
$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}.$$
- **Balanced accuracy** (multi-class):
$$\text{Balanced Accuracy} = \frac{1}{K} \sum_{k=1}^{K} \frac{\text{TP}_k}{\text{TP}_k + \text{FN}_k},$$
where K is the number of classes and TP_k, FN_k are the true positives and false negatives for class k.

XGBoost

The choice of XGBoost was motivated by its ability to iteratively correct mistakes on hard-to-classify samples through gradient-boosted decision trees. Because our core challenge is severe class imbalance—particularly predicting the extremely rare M and X class solar flares—the model's architecture, which sequentially fits trees to the residual errors of previous trees, would allow the model to focus on improving its predictive ability on the misclassified minority-sample classes: the later trees concentrate on the samples the earlier ones struggled with. Its built-in regularization (L1, L2, and shrinkage) also helps prevent excessive overfitting to oversampled or synthetic data

Baseline XGBoost

We began by evaluating a baseline XGBoost classifier without any resampling or weighting adjustments, and progressively introduced modifications aimed at improving the representation of minority classes. The baseline model already exhibited very high training balanced accuracy (>95%) but substantially lower test balanced accuracy (55–57%), and the same was seen in terms of the loss difference between training and loss as shown by the large vertical gap between the blue (training) and orange (test) curves in Figure 1 indicating clear overfitting. After increasing the complexity- maxdepth, nestimators- and tuning the hyperparameters, subsample, colsample by tree, and regularization terms which are inherent in the XGBoost model (λ for L2 and α for L1)—the test scores improved only marginally to about 63% balanced test accuracy. The figures below proved that after a certain complexity, the model hits a ceiling accuracy above which it does not break by increased complexity. This meant that the model does not gain new generalizable information through more non-linearity, mainly because its minority-class understanding is limited not by model capacity but by lack of representative data.



Controlled Smote

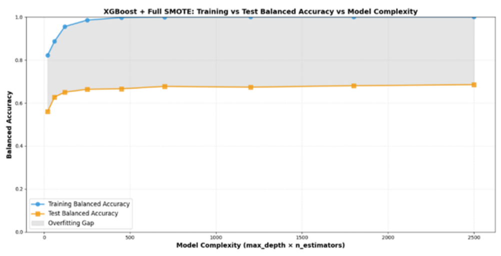
To tackle this lack of data, we introduced SMOTE. We applied a controlled form of SMOTE that expanded only the M and X classes to 9000 samples each. Although this increased their nominal presence in the dataset, the performance metrics (precision, recall, F1, balanced accuracy) barely changed, and overfitting persisted on the M and X samples. Based on the equation of SMOTE, we know that the synthetic samples are the result of a linear interpolation between the existing samples. With that in mind, we observe that because the original minority samples already form tight clusters in feature space, the synthetically generated points will be concentrated near these clusters, making them more dense, but not expanding the underlying support of the distribution or adding valuable information for the model to generalize on. Also, because the dataset is still dominated by the majority 0-class flares, we hypothesized that most splits are influenced by majority-class patterns, so even if XGboost trees aimed to learn from the X and M flares, these are still low compared to the 0-class flares. In conclusion, controlled SMOTE did not provide the model with more information about the non-linearity in the minority samples, but only more repetitions of the original dataset, and the imbalance in the data remained, although less strong now. These factors all point to the results that indicate no improvement in the model's ability to generalize.

Table 1: *Classification Report: XGBoost + Controlled SMOTE*

Class	Precision	Recall	F1-score
0	1.00	1.00	1.00
C	0.84	0.81	0.83
M	0.32	0.74	0.45
X	0.00	0.00	0.00
Macro Avg	0.54	0.64	0.57
Weighted Avg	0.92	0.93	0.92

Full SMOTE

Next, Full SMOTE brought all classes up to the size of the largest class, meaning even the abundant C-class samples was oversampled. Surprisingly, results improved slightly compared to controlled SMOTE (e.g., test balanced accuracy increasing from 0.638 to 0.674), despite the clear overfitting.



We hypothesized that with Full SMOTE, the XGBoost model became less biased towards the majority classes, especially the M class on which recall improved compared to controlled SMOTE, which reduced the model's tendency to ignore the minority classes. This translated directly into improved balanced accuracy, but the improvement is still not much because the synthetic data fails to expand the original minority class samples in a meaningful way that allows the model to capture new patterns it could not observe in the original dataset. The clusters of M and X classes led to overfitting on these classes, but that was less than before.

Table 2: *Classification Report: XGBoost + Full SMOTE*

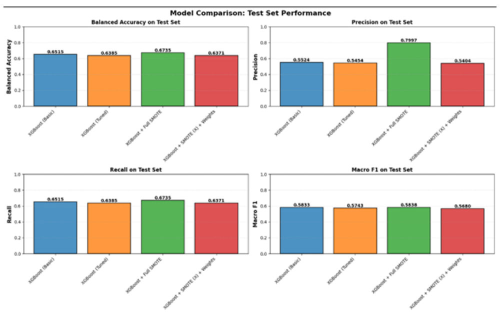
Class	Precision	Recall	F1-score
0	1.00	1.00	1.00
C	0.86	0.79	0.82
M	0.34	0.89	0.49
X	1.00	0.01	0.02
Macro Avg	0.80	0.67	0.58
Weighted Avg	0.95	0.93	0.92

Class weights

Given that oversampling did not resolve the issue of overfitting, we introduced class weights. The rationale behind that is to increase the penalty for misclassifying minority-class events. This would directly force the model to pay more attention to M and X classes, and increase the gradient contributions from errors on these classes. Despite that, class weights coupled with SMOTE did not perform better than Full SMOTE. This is mainly related to the fact that the previous models are already able to capture the patterns in the minority classes without class weights, but the data imbalance prevents improvement since even strong weights cannot create new geometric regions to learn from.

Best XGBoost model, comparative analysis

Conducting a comparative analysis, we observed that XGBoost with Full SMOTE performed better than all other models that were finetuned on every measure: precision, recall and balanced accuracy. However, the model also overfitted on the M and X classes as evidenced by the precision and recall results and the gap between test balanced accuracy and training balanced accuracy. Further finetuning of the XGBoost model has shown not to be effective, which is why our approach was to transition to other architectures discussed next.



SVM

A Support Vector Machine (SVM) is a supervised classifier that finds the optimal decision boundary with the maximum margin—the greatest distance between the boundary and the nearest data points from each class (support vectors). For nonlinearly separable data, it uses the kernel trick to implicitly map data into a higher dimensional space, enabling linear separation without direct transformation. Its behavior is tuned by key hyperparameters:

- **C (Regularization):** Balances margin width against classification accuracy (low C = wide margin, high C = strict fitting).

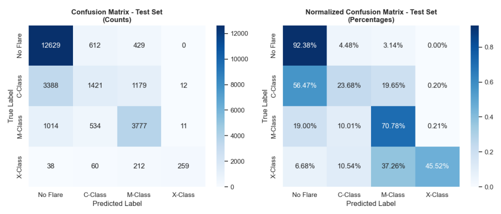
- **Kernel:** A kernel defines the shape of the decision boundary. A linear kernel creates a straight hyperplane and assumes classes can be separated by a line, while an RBF kernel creates flexible, curved boundaries that can model intricate, non-linear relationships in the data.
- **Class Weight:** Adjusts misclassification penalties for imbalanced data ('balanced' weights minority classes higher).
- **Gamma (RBF):** Controls the influence radius of each data point (low = smooth boundaries, high = tight fitting).

Baseline Model

Our development journey began by establishing a baseline using a linear SVM kernel on the original imbalanced dataset, achieving an overall accuracy of 0.67 with severe class disparities. While the model showed high recall for the majority "No Flare" class (93%), it critically failed to detect rare but important flares—X-Class recall was only 18%, and C-Class just 20%. This baseline revealed both the linear kernel's inability to capture complex feature relationships and the detrimental effects of class imbalance, making the model unsuitable for operational deployment where detecting rare solar flares is essential.

Introducing Non-Linear Separation

To address the linear kernel's limitations, we transitioned to an RBF kernel for non-linear decision boundaries. This model achieved the highest accuracy of 0.71, but analysis revealed severe bias: X-Class recall was only 46% despite 92% precision—meaning it missed over half of critical flares. This confirmed that without balancing mechanisms, the RBF kernel still favored the majority class.



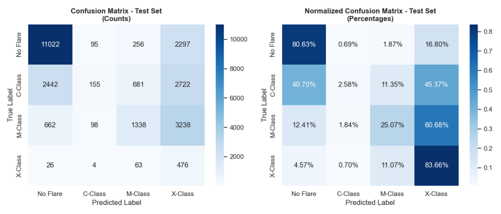
Addressing Class Imbalance Using SMOTE

Recognizing that class imbalance remained the core issue, we shifted to data-level intervention with SMOTE, oversampling X-Class to 4,000 samples. SMOTE with an RBF kernel (C=1) failed completely, predicting 99.71% of instances as "No Flare" and achieving 0% X-Class recall, as the RBF kernel treated the distinct, synthetic clusters as an "outlier" region separate from the main data distribution, defaulting to the original majority class.

Table 3: *SMOTE with RBF Kernel Performance*

Class	Precision	Recall	F1-Score
No Flare	0.54	1.00	0.70
C-Class	0.43	0.01	0.02
M-Class	1.00	0.00	0.00
X-Class	0.00	0.00	0.00
Acc			0.54

Attempting to address overfitting to synthetic samples, we applied strong regularization with C=0.1. This proved catastrophic, the model defaulted to predicting everything as "No Flare". Conversely, SMOTE with a linear kernel (C=1) over-predicted minority class, yielding 84% X-Class recall but only 5% precision, because its simplistic decision boundary could not distinguish genuine solar flare patterns from artificial synthetic clusters in the now-balanced feature space.



All three experiments demonstrated that synthetic oversampling created SVM models that ignored minority classes or generated false positives, prompting a strategic pivot back to the original dataset with class weighting.

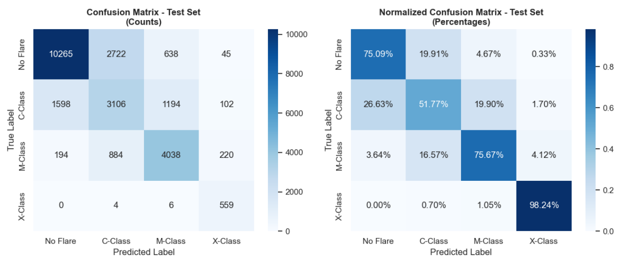
Class Weighting Instead of SMOTE

Learning from the failed SMOTE experiments, we pivoted to class weighting within the original dataset using `class_weight='balanced'`. This adjusted the penalty for misclassifying minority classes, directing the algorithm's focus toward X-Class and M-Class instances during training.

The RBF model with class weighting showed marked improvement: X-Class recall surged to 96% (from 46%), with a 96.31% true positive rate visible in the confusion matrix. M-Class recall improved to 72%, and overall accuracy reached 0.68. While X-Class detection dramatically improved, its precision dropped to 48%, revealing an increase in false positives—a trade-off necessitating further tuning.

Table 4: *RBF with Class Weighting*

Class	Precision	Recall	F1-Score
No Flare	0.84	0.75	0.79
C-Class	0.44	0.47	0.45
M-Class	0.65	0.72	0.68
X-Class	0.48	0.96	0.64
Acc			0.68

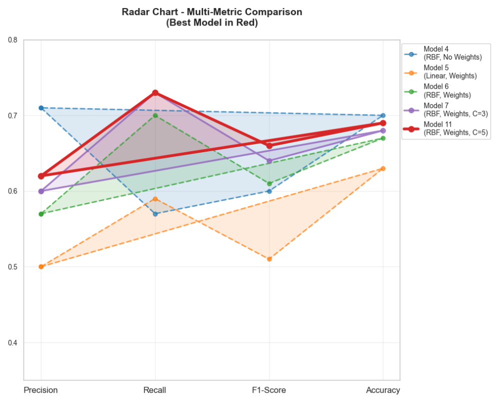


Fine Tuning the Regularization Parameter (C)
To further optimize performance after establishing class weighting as effective, I tuned the regularization parameter C which balances margin maximization against classification error. Increasing to C=3 improved X-Class recall to 98% with 97.54% true positive rate and M-Class recall to 75%, while X-Class precision rose to 56%, reducing false positives. Optimizing at C=5 produced the best-balanced model, achieving 98% X-Class recall with 60% precision (98.24% true positive rate), 76% M-Class recall, 52% C-Class recall, and 0.70 overall accuracy with a 0.71 weighted F1-score.

Table 5: *Optimized RBF (C=5)*

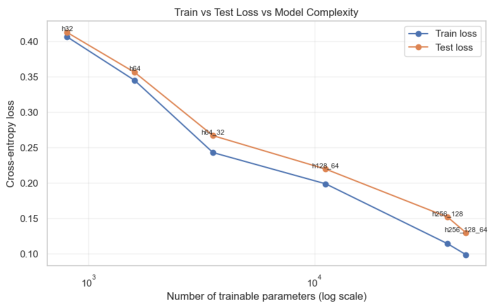
Class	Precision	Recall	F1-Score
No Flare	0.85	0.75	0.80
C-Class	0.46	0.52	0.49
M-Class	0.69	0.76	0.72
X-Class	0.60	0.98	0.75
Acc			0.70

Best SVM Model
The optimization culminated in Model 11 (RBF, C=5, `class_weight='balanced'`), which achieved the optimal balance: 98% X-Class recall with 60% precision, 76% M-Class recall, and 0.70 overall accuracy with a 0.71 weighted F1-score. As shown in the multi-metric radar chart (Best Model in Red), Model 11 outperforms all previous configurations across Precision, Recall, F1-Score, and Accuracy, establishing it as the most reliable classifier for operational solar flare forecasting.

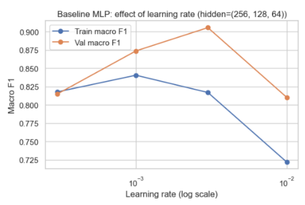


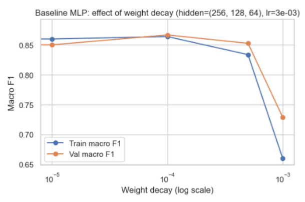
Deep Networks

Neural and Deep Networks I
Our first approach involved training baseline neural and deep networks of fully connected layers with **ReLU** activations. We tried varying architectures to establish a performance foundation before increasing model complexity. We conducted three systematic ablation studies to understand the impact of model complexity, learning rate, and regularization. Our first step was to systematically vary hidden layers from 1 layer to 3 layers, training each configuration with identical hyperparameters (learning rate=1e-3, weight decay=1e-4). The individual learning curves for each architecture reveal a clear progression. Simpler architectures (h32 (1 hidden layer of 32 neurons), h64) exhibit **underfitting** characteristics, meaning that both training and test losses converge at high values with F1 scores plateauing around 0.60-0.65. This underfitting occurs because these models lack sufficient capacity to capture the complex non-linear relationships between features and flare classes. The training and validation curves remain closely aligned, indicating that the models are not overfitting but rather failing to learn adequate features. As complexity increases, we observe a clear improvement in performance. It demonstrates a decreasing curve for test and training losses where it reaches the minimum on the model **h256_128_64**.



Having identified the optimal architecture (h256_128_64), we next investigated the **learning rate**, which critically affects convergence speed and final performance. We tested values [3e-4, 1e-3, 3e-3, 1e-2], maintaining the optimal architecture constant. The results demonstrate that learning rate 0.003 achieves optimal validation F1 of 0.9056. Lower and higher learning rates caused training instability and lower F1 scores. With the optimal architecture and learning rate established, we investigated **weight decay** values [0.0, 1e-5, 1e-4, 5e-4, 1e-3] to determine the impact of explicit L2 regularization. Surprisingly, weight_decay=0.0 achieved optimal F1 score.

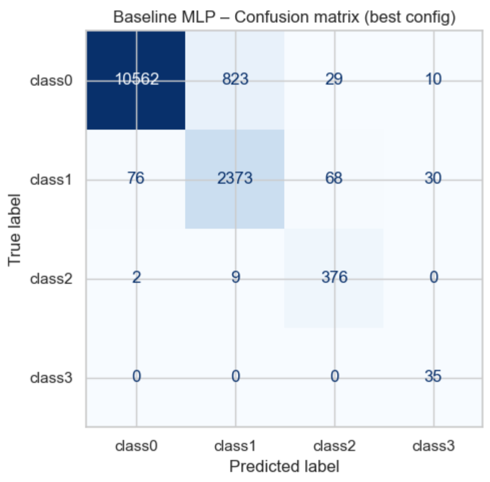




The optimal baseline configuration (h256_128_64, lr=0.003, wd=0.0) achieved a test macro F1 of 0.8408. The confusion matrix reveals excellent performance on class 0 (precision=0.993, recall=0.925) but demonstrates the model's difficulty distinguishing C and M classes, with significant off-diagonal misclassifications. X-class flares achieve perfect recall (1.000) but low precision (0.467), indicating **over-prediction** to avoid missing critical events. This performance establishes a strong baseline, but we observed that the model struggles with **class separation**, particularly between C and M flares.

Table 6: *Classification Report: 256_128_64, lr=0.003, wd=0.0*

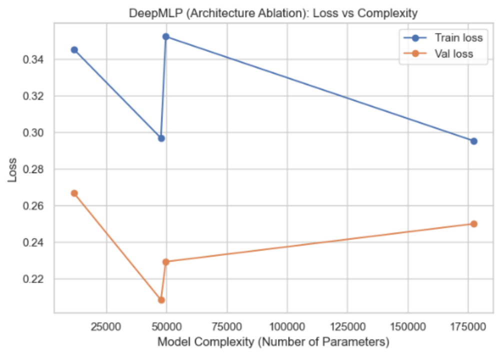
Class	Precision	Recall	F1-score
0	0.993	0.925	0.957
C	0.740	0.932	0.825
M	0.795	0.972	0.874
X	0.467	1.00	0.636
Macro Avg	0.749	0.957	0.823
Weighted Avg	0.941	0.927	0.931



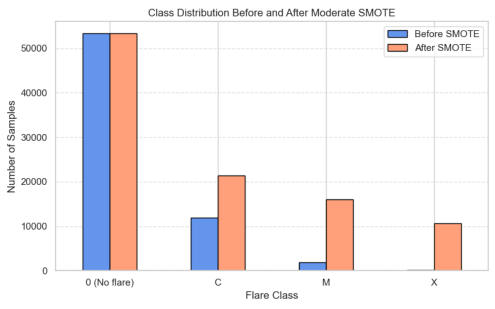
True \ Predicted	class0	class1	class2	class3
class0	10562	823	29	10
class1	76	2373	68	30
class2	2	9	376	0
class3	0	0	0	35

Deep Networks II

Based on the previous trials and tests, we hypothesized that deeper architecture layers could capture more complex features. Hence, we shifted to deep networks that incorporate **BatchNorm** and **Dropout** (p=0.3) between each fully connected layer. We expected that normalization and regularization would allow us to train deeper networks that could learn more decision boundaries. However, our hypothesis is **disproven**, as the deep network with identical architecture (256, 128, 64) achieved only 0.6858 validation F1 and 0.6519 test F1, significantly worse than baseline. The complexity ablation for the deep network reveals that increased depth does not improve performance. The loss versus complexity plot shows **larger gaps** between losses compared to baseline, indicating worse generalization. Thus, this led us to reconsider our approach.



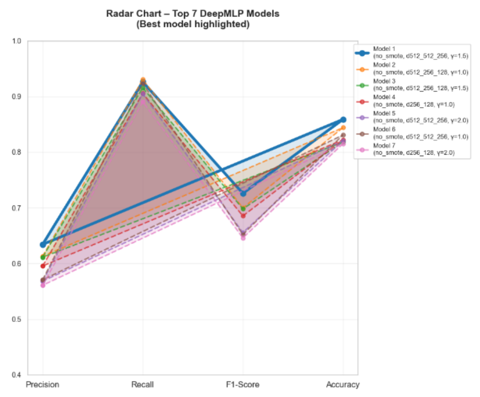
We recognized that the fundamental challenge was not model architecture complexity, but rather the severe **class imbalance** in our dataset. Thus we tested four variants of **SMOTE**: (1) no SMOTE (baseline), (2) full SMOTE (all classes balanced to equal proportions), (3) partial SMOTE (minority classes at 50% of majority), and (4) controlled SMOTE (C: 40%, M: 30%, X: 20% of majority).

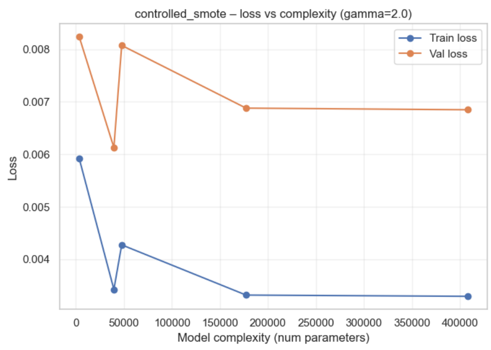


Our original no-SMOTE setup was the strongest approach as the model misclassified class 0 almost entirely and over-predicted minority classes, which confirms that the synthetic points introduced **harmful patterns**. This critical finding demonstrates that synthetic oversampling might not always be beneficial. Therefore, we investigated **Focal Loss** as an adjunct approach to SMOTE to tackle class imbalance. Focal Loss modifies the loss function to focus learning on hard examples and minority classes. It cross-entropy loss to downweight well-classified examples, with the formula:

$$\boxed{\mathrm{FL}(p_t) = -(1-p_t)^{\gamma}\log(p_t)}$$

where p_t is the model's predicted probability for the true class. Additionally, higher γ values increase focus on hard examples. We tested gamma values [1.0, 1.5, 2.0, 2.5, 3.0] across all SMOTE variants and multiple architectures, reasoning that this approach might succeed where SMOTE failed because it doesn't introduce artificial data patterns. The training curves demonstrate different dynamics as the loss values start much lower, indicating successful downweighting of easy examples.

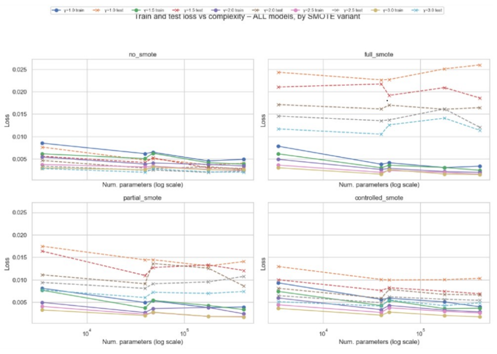




The gamma experiments showed a consistent pattern. When gamma is low (1.0 or 1.5), the model keeps a good balance between precision and recall. When gamma becomes larger (2.5 or 3.0), the model starts to focus too strongly on hard or minority examples. This increases recall for M and X classes, but it also creates many false positives, which lowers precision.

Across all gamma values, the models trained without SMOTE show the cleanest and most stable learning behavior. Training and test losses decrease together as model size increases, and the gap between them remains small. This means the model is learning real structure in the data and is not overfitting. The smooth learning curves across all γ variants confirm that the original dataset is learnable without modification.

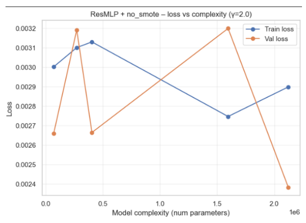
Once synthetic oversampling is introduced, this stability disappears. Full SMOTE produces the largest train–test gap. The model memorizes synthetic samples, leading to low training loss but much higher test loss. Generalization breaks down because the synthetic points do not represent new, meaningful patterns. Partial SMOTE reduces this effect, but the validation loss still stays consistently higher than the no-SMOTE baseline, showing that even limited interpolation harms the data geometry. Controlled SMOTE behaves the best among the SMOTE settings, with smoother curves and a smaller gap, but it still fails to reach the stability and low test loss of the no-SMOTE models.

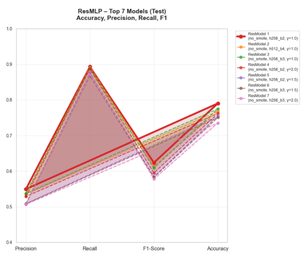


Across all variants of SMOTE with Focal Loss, none of the SMOTE methods outperform the original data, and all introduce different degrees of overfitting, making no smote the strongest and most reliable setting. We can verify that the best model of no smote and γ=1.5 and architecure of (512, 512, 256), which is worse than the (256, 128, 64) model with Relu activation function, has precision, recall, and F1 macro averages of (0.6130, 0.9306, 0.6998).

ResMLP

Given pevious deep network **underperformance** and the mixed results from Focal Loss, we investigated **ResMLP** architectures incorporating **residual skip connections**. We hypothesized that residual connections could enable deeper networks while preserving **gradient flow** and preventing **vanishing gradients**, potentially allowing us to train deeper networks that could capture more complex patterns. ResMLP without focal loss achieved validation F1 of 0.7562 (h512_b4), better than models trained in Deep Networks II but still below our primary model. The learning curves demonstrate smoother convergence, with more stable validation performance throughout training. However, when combined with **focal loss**, ResMLP performance became way worse. The best ResMLP with focal loss achieved only 0.459 test F1 (h256_b2, no_smote, γ=1.0), significantly worse than baseline (0.8408). The complexity ablation graphs demonstrate severe **overfitting** when focal loss is combined with ResMLP. Training loss decreases smoothly to near zero while validation loss increases after initial improvement, reaching values above 1.5-2.0 for larger architectures. The train and val F1 gap widens for complex architectures. This overfitting occurs because the combination of residual connections, batch normalization, dropout, and focal loss creates excessive regularization. Each technique individually helps generalization, but together they create competing regularization mechanisms that prevent effective learning.





Best Deep Network
The analysis across multiple architectures, loss functions, and sampling strategies reveals that simpler models consistently outperform complex ones. The optimal architecture was our baseline network (256_128_64, lr=0.003, wd=0.0) with a macro average of precision, recall and F1-score of (0.749, 0.957, 0.823). Each technique (dropout, batch norm, residual connections, SMOTE, focal loss) individually helps generalization, but combining them creates competing regularization mechanisms that prevent effective learning. The baseline deep network achieved optimal performance because it matched architecture complexity to dataset characteristics without excessive regularization.

Ensemble Model

The ensemble model that combines XGBoost and Baseline Neural Network outperformed all previous individual models because it combined the strengths of both XGBoost and the deep network rather than relying on a single decision process. XGBoost performed very well on the abundant classes, while the deep network was able to capture the patterns of the minority samples and generalize well on unseen datapoints of the M and X class flares. By averaging their predictions, the ensemble reduced the weaknesses of each model—XGBoost's tendency to underdetect rare flares and the neural network's tendency to overfit high-dimensional spaces. This is why the ensemble achieved much higher recall for M- and X-class flares, which were consistently difficult for the individual models.

Table 7: *Ensemble Model*

Class	Precision	Recall	F1-Score
0	0.99	0.94	0.96
C-Class	0.80	0.90	0.84
M-Class	0.72	0.98	0.83
X-Class	0.54	0.97	0.69
Bal. Acc			0.92

Conclusion

With these tested models and their respective ablations, we observed that each of the individual models captured only part of the structure in the data. XGBoost performed well on abundant classes but underdetected rare ones. The neural network learned nonlinear patterns but over-predicted minority classes. The SVM improved minority recall but suffered from precision loss. However, the ensemble model, a combination of XGBoost and Deep Network, emerged as the strongest and most reliable approach in our study. It achieved the highest balanced accuracy and macro-F1, with excellent recall for the critical M- and X-class flares. The ensemble reached 0.96 F1 for the majority class, 0.84 for C-class, 0.83 for M-class, and 0.69 for X-class. These results exceed the performance of every individual model we trained, as the ensemble combined the strengths of the models, which led to improved results across all flare types.

Github Link

Click here to access the Github repository

References

[1] Dataset: Dryad Dataset.

[2] Goel, A., et al. "Solar Flare Prediction using Machine Learning Algorithms on RHESSI Dataset." *ArXiv*, 2022, https://arxiv.org/pdf/2505.03385v1.

[3] Omitoyin, O., et al. "Visualization and Prediction of Solar Flares Class Based on Sunspot and Active Region Data using Machine Learning Techniques." *TechRxiv*, 2024, https://www.techrxiv.org/users/839525/articles/1230137-visualization-and-prediction-of-solar-flares-class-based-on-sunspot-and-active-region-data-using-machine-learning-techniques.