Space Weather

RESEARCH ARTICLE
10.1029/2026SW005042

Comprehensive Validation of Novel Deep Learning Architectures to Forecast Geomagnetic Substorms

A. Essop[1], N. Mbatha[2], and J. A. E. Stephenson[1]

[1]Discipline of Physics, School of Science and Agriculture, University of KwaZulu-Natal, Durban, South Africa, [2]Council for Scientific and Industrial Research, Pretoria, South Africa

Key Points:

- Novel machine learning models to forecast the onset of a magnetic substorm over the next 1 min are presented
- The models were developed using a trusted list of onsets spanning 2003 and 2022, achieving 80%–84% for precision and 81%–83% for recall
- The models are cross-validated and achieve 79%–85% in terms of F1 score

Supporting Information:

Supporting Information may be found in the online version of this article.

Correspondence to:

A. Essop,
psychedelicartist@rocketmail.com

Citation:

Essop, A., Mbatha, N., & Stephenson, J. A. E. (2026). Comprehensive validation of novel deep learning architectures to forecast geomagnetic substorms. *Space Weather*, *24*, e2026SW005042. https://doi.org/10.1029/2026SW005042

Received 3 MAR 2026
Accepted 11 JUN 2026

Author Contributions:

Conceptualization: J. A. E. Stephenson
Formal analysis: A. Essop
Methodology: A. Essop
Software: A. Essop
Supervision: N. Mbatha, J. A. E. Stephenson
Writing – original draft: A. Essop

Abstract Magnetic substorms are disturbances in the terrestrial magnetosphere that can have significant space weather impacts, although forecasting their onset accurately remains an open problem. In this study, we develop multiple novel machine learning architectures based on convolutional neural networks, long short-term memory networks, extreme gradient boosting, and their hybrid combinations for the probability prediction of magnetic substorm onsets. The best performing models came from hybrids in the following order: two-dimensional convolution, bidirectional long short-term memory, and finally extreme gradient boosting for binary classification probability prediction with an F1 score of 0.8411 across both classes. Notably, a novel modified stratified K-fold cross-validation benchmark is developed for these deep learning models; the F1 scores fell within 79%–85% across model architectures. The results demonstrate the ability of models to unambiguously forecast substorm onsets from upstream solar wind and IMF conditions. Two-hr historical windows of solar wind velocity, proton number density, and three-dimensional interplanetary magnetic field components, sampled at 1-min cadence, are used to forecast the onset probability at the next minute after the window. The input data set was balanced using the SuperMAG list of substorm onsets between 2003 and 2022, with an equal number of non-substorm intervals, to avoid bias. Principal component analysis showed significant overlap in the solar wind and IMF conditions prior to both substorm and non-substorm intervals, indicating that these parameters alone may not be sufficient to forecast all substorms and that internal magnetotail preconditioning and magnetospheric processes are likely critical areas that should be explored further.

Plain Language Summary Space weather starts with the Sun. When the solar wind-a constant stream of particles from the Sun-hits Earth's magnetic field, it can sometimes trigger explosive energy releases called magnetic substorms, which can disrupt satellites, power grids, and GPS signals. Scientists are good at detecting substorms once they start, but forecasting them ahead of time is very difficult. In this study, we tested several different types of artificial intelligence (AI) models to see if they could learn to predict the probability of a substorm occurring in the next minute, based on measurements of the solar wind taken over the previous 2 hrs. We found that a hybrid AI model, which combined multiple techniques, was the most successful at forecasting substorm onsets. However, we also discovered that the solar wind conditions before a substorm often look very similar to the conditions before a period of calm with no substorm. This suggests that the Sun's immediate output isn't the only factor at play. Our research shows that while AI is a promising tool for space weather forecasting, it has limits. To make better predictions, scientists also need to understand what's happening inside Earth's magnetic field—not just what's coming in from the Sun.

1. Introduction

The movement of energy from the solar wind (SW), through the magnetosphere, and into the Earth's ionosphere is a dynamic process with significant implications for the planet's environment and technological infrastructure. Among the class of events associated with such energy transfers is the *substorm*—a sudden release of stored energy from the magnetosphere into the ionosphere. A substorm is typically categorized into three distinct phases: a growth phase where energy from the solar wind is transferred to and stored in the magnetotail; an expansion phase where stored energy is released from the magnetotail into the ionosphere, and a recovery phase where the system returns to quiet-time behavior (Akasofu, 1964; McPherron et al., 1973). The duration of a substorm is typically 1–2 hr although variations do exist within the phases themselves (Fu et al., 2025; Juusola et al., 2011; Partamies et al., 2013).

Substorms are observed in the ionosphere as intense auroral displays and measured on the ground as magnetic disturbances. Substorms are marked by the abrupt appearance of a bright auroral arc in all-sky film data that occur on the order of a few minutes (Akasofu, 1964), with the successive occurrence of the poleward boundary intensification (PBI) of the auroral oval and a north-south arc moving toward the equator on to the onset latitude (Nishimura et al., 2010, 2025). Such activity is also related to the formation of the Substorm Current Wedge (SCW), a three-dimensional current system that connects the magnetotail to the ionosphere (McPherron et al., 1973). The SCW forms a diversion of the cross-tail current through the ionosphere via FACs (Boström, 1964; McPherron et al., 1973). The ionospheric segment of the SCW, known as the westward auroral electrojet, produces a characteristic magnetic signature detectable by ground-based magnetometers. These metrics are known as the auroral electrojet (AE) indices (Davis & Sugiura, 1966; Newell & Gjerloev, 2011). This study uses the changes in the SuperMAG auroral lower (SML) index to define the substorm (Newell & Gjerloev, 2011).

Substorms have been the subject of intense study from a myriad of perspectives using data from both Earth and space based instruments (Akasofu, 1964, 2023; Lyons & Nishimura, 2020; Maimaiti et al., 2019; McPherron et al., 1973; Newell & Gjerloev, 2011). Of specific interest is the effect on the growth phase of specific solar wind (SW) and interplanetary magnetic field (IMF) variables (Luo et al., 2013; Maimaiti et al., 2019; Newell et al., 2007, 2016). These studies find IMF B_z and SW V_x to be the crucial variables modulating substorm activity. There are two cases that incorporate both of the directional trends that occur in opposite directions with regard to B_z. These are the cases of the southward and northward trends at specific intervals within the B_z timeseries before the substorm onset. In the first case, the southward directed B_z is considered necessary for energy to move from the IMF to the magnetosphere (Kumar et al., 2024; McPherron et al., 1973; Pudovkin, 1991). In the second case, it has been proposed by others that a northward directed B_z may externally trigger the substorm onset (Lyons et al., 1997; Lyons & Nishimura, 2020). This a contentious proposal, since contradictory observations have been found. These imply an internal triggering mechanism for the substorm onset, which is thus not contingent on B_z being northward directed (Johnson & Wing, 2014; Morley & Freeman, 2007). The other variable of importance is SW V_x. It has been reported that the growth phase is associated with an increase in the Earth-bound solar wind (Fu et al., 2021, 2025; Partamies et al., 2013). However, the occurrence of substorms during quiet-time magnetic conditions has also been reported (Kullen & Karlsson, 2004; Peng et al., 2013). This suggests that there is a class of substorms that occur outside of strong geomagnetic periods, departing from this conventional association (Partamies et al., 2013).

The overall mechanism and the order in which the physical processes take place apropos onset triggering remains elusive, and is still a subject of ongoing research. Of particular interest is the order of occurrence for magnetic reconnection and current disruption (Baker et al., 1996; Lui, 1996). It has been observed that magnetotail reconnection happens before current disruption by the Time History of Events and Macroscale Interactions during Substorms (THEMIS) mission (Nishimura et al., 2010). The preconditioning of the magnetotail as a factor that needs addressing is also encountered in this study.

Space-based assets have become vital for nations on the planet due to the increased interconnectedness of these areas within the technology, defense and economic agreements that they partake in Johnson Freese (2007); Kawai and Hanaoka (2025). Substorms pose a direct threat to these infrastructures; they can inject energetic particles that cause satellite surface charging and component damage (Bodeau, 2015; Bodeau & Baker, 2021; Horne et al., 2013), and have induced geomagnetic currents that disrupt power grids (Freeman et al., 2019). Another important effect is that abrupt changes in the ionosphere's electron density can lead to communication and global navigation system errors (Coster & Yizengaw, 2021; Zakharov et al., 2020). The practical need for reliable forecasts has motivated the application of machine learning (ML) to this problem, as its utility has been well established across various fields (Torres et al., 2021). With specific emphasis to the field of space weather, early approaches used neural networks to predict auroral electrojet indices as proxies for substorm activity (Gleisner & Lundstedt, 1997; Hernandez et al., 1993). Although these studies were foundational, these models did not directly predict onsets, and the computational capabilities deployed were nowhere near what can be handled in the contemporary context of big data. Later work by Tanaka et al. (2015) used Support Vector Machines on a small data set of less than 200 intervals, achieved 78% accuracy in onset prediction. All the models mentioned until this point required time-consuming manual feature engineering, an inefficient process that rendered them undesirable when working with large volumes of complex data. The direct precursor to this study is that of Maimaiti

et al. (2019), who applied a deep convolutional network known as *ResNet* to a larger data set and achieved 75% accuracy in predicting onsets within a 60-min window using a 2-hr history of solar wind parameters. The model was developed in order to directly predict onsets, in contrast to the aforementioned cases of using the auroral indices as proxies for substorm activity.

A significant limitation persists in the existing literature: a lack of robust, generalized validation. Many previous studies, including Maimaiti et al. (2019), report performance based on a single, optimal train-test split, which risks overfitting and may not accurately represent model performance on unseen data (King et al., 2021; Seraj et al., 2023). The contrast between this standard, as presented in the study by Maimaiti et al. (2019) and that of the cross validation of one of this study's hybrid models are given in Figure 1, details of which can be found in Section 4.

Figure 1 shows the loss and accuracy curves on the training and validation data sets respectively for the study by Maimaiti et al. (2019). This has been typical of deep learning studies to this point that lack comprehensive cross validation. The curves in the white background with the header "Stratified k-fold Cross Validation" are the accuracy and loss curves from the benchmarked cross validation on the training and validation data sets respectively. They are acquired from five independent partitioned folds of data that preserve the structure of each time series sample in both the training and validation data sets.

In this study, we present a comprehensive machine learning approach to forecast the occurrence probability of a magnetic substorm onset as defined using the SuperMAG SML index (Newell & Gjerloev, 2011). Specifically, we use intervals of 2-hr time history of the Earth-bound solar wind bulk speed (V_x), IMF components (B_x, B_y, and B_z), and proton number density (N_p), within the geocentric solar magnetospheric (GSM) coordinate system as input data in order to forecast the magnetic substorm onset's occurrence probability over the next 1 min. The magnetic substorms used for model development are taken from the SuperMAG database. They comprise about 46,000 magnetic onsets spanning 2003 to 2022. We build upon previous work by: (a) Developing and benchmarking 11 novel model architectures, including hybrids of two-dimensional convolutional networks (Conv2D), long short-term memory (LSTM) networks, and extreme gradient boosting (XGB). (b) Implementing a stringent, temporally aware stratified K-fold cross-validation framework to provide an unambiguous and generalized assessment of model performance. (c) Utilizing an updated and expanded data set from the SuperMAG database to train and test our models. Our goal is not only to achieve high forecasting skill but also to provide a robust, validated benchmark for future work in this domain.

The remainder of this study is structured as follows. Section 2 details the data sources and preprocessing methods, defines technical terms and metrics, outlines the machine learning model architectures and their cross-validation framework. Section 3 presents the results, and Section 4 discusses their implications in relation to the established literature. Finally, Section 5 provides conclusions and perspectives for future work.

2. Data Sources and Models' Deployed

2.1. Data Sets and Data Preprocessing

SW-IMF variable data. The OMNI data set is the main historical high resolution (1-min and 5-min) data set on the solar wind, that is time adjusted to the bow shock (Vokhmyanin et al., 2019). The justification and underlying calculations for the time adjustment of the SW-IMF data are outlined at https://omniweb.gsfc.nasa.gov/html/HROdocum.html. Further requirements (specific to the context of this project) for time adjustments can be found in Maimaiti et al. (2019), where a 10 min delay was chosen for the substorm data, after the authors had experimented with multiple delay times between 0 and 20 min. The reasons for this uniform time delay are threefold viz. the time taken for the solar wind to travel across the magnetosheath, the time taken for the magnetotail to be perturbed by the external disturbance at the magnetopause, and the time taken for the onset to be observed by ground based magnetometers. The SW-IMF input data used in this study are the 1-min combined, definitive, averaged IMF and SW plasma data time shifted to the nose of the Earth's bow shock. They are taken from the ACE, Wind and IMP-8 instruments, as part of the OMNI data base. Furthermore, the data is also time delayed by 10 min following the protocol established by Maimaiti et al. (2019). In particular, the key variables used to develop data points in this project are time series of B_x, B_y, B_z (3-D magnetic field components), V_x (Earth-bound solar wind velocity component) and N_p (density of the charged solar wind particles). An example of a time series incorporating the substorm window is given in Figure 2.

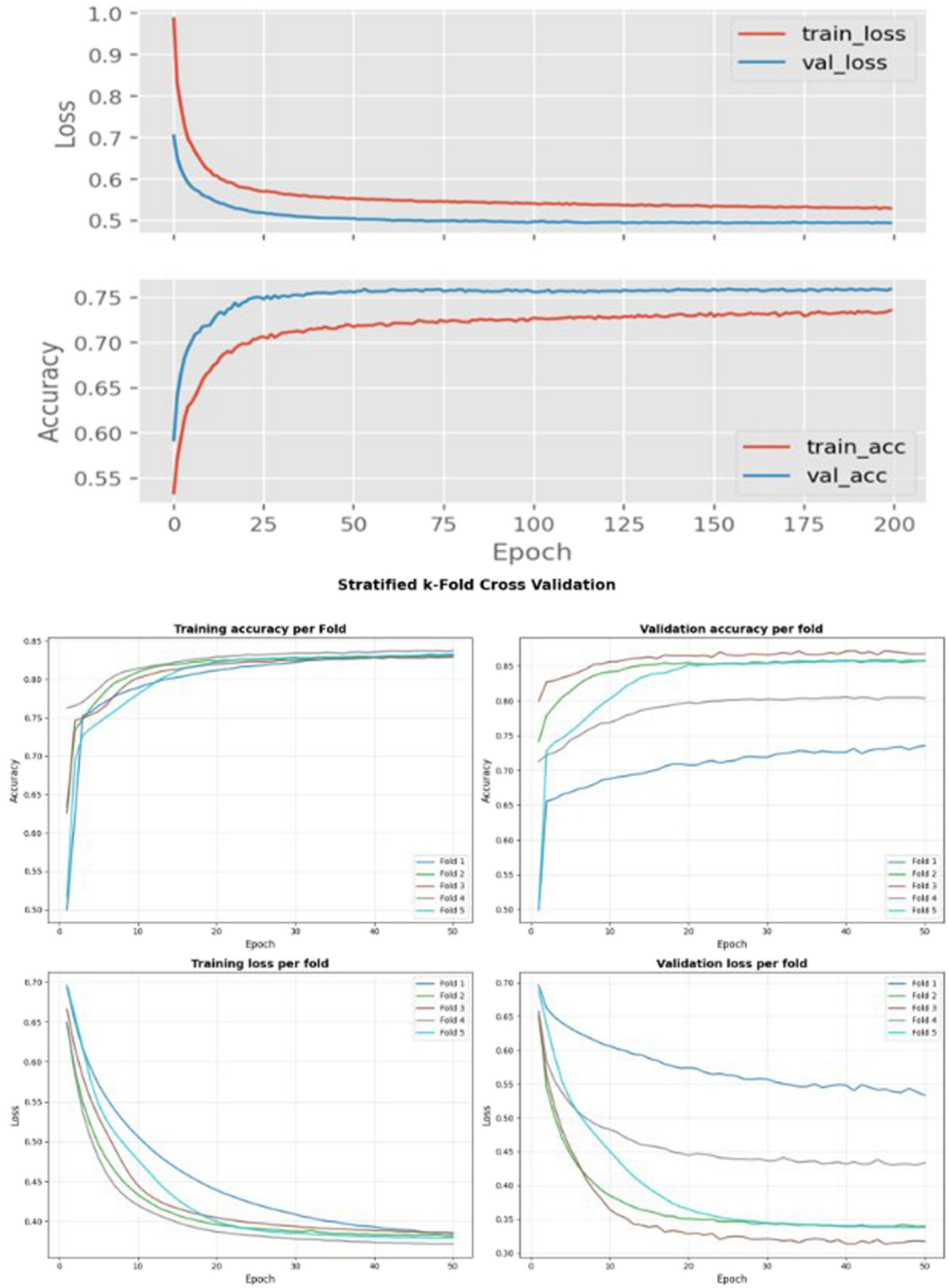

Figure 1. Training and validation loss and accuracy curves from Maimaiti et al. (2019) (gray background) compared with those from this study (white background).

SuperMAG substorm onset list A global network of approximately 300 ground-based magnetometers operated by multiple organizations and national authorities is unified and made available under an initiative known as SuperMAG (see https://supermag.jhuapl.edu/). A comprehensive library consisting of multiple different data sets is available on this website. Of particular interest for this study is the list of onset times for substorms. These are identified by an onset identification algorithm proposed by Newell and Gjerloev (2011). This uses the SuperMAG

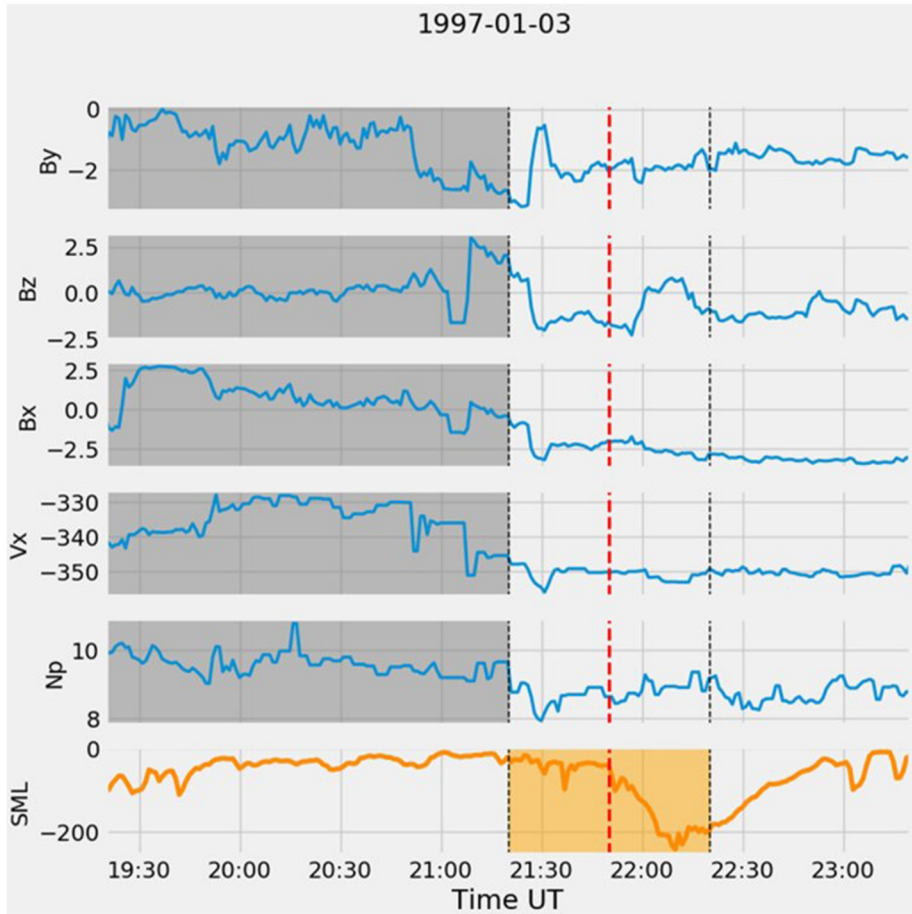

Figure 2. An individual case of the variables used to form a data point incorporating a substorm window showing the onset on time-series plots of B_x, B_y, B_z, V_x and N_p. The shaded gray region indicates the time series data used as input, while the region bounded by the dashed vertical back lines shows the subsequent 60 min prediction window, where the onset is shown by the dashed vertical red line. Adapted from Maimaiti et al. (2019). This study fixes the data point at the 121th min as the onset, and does not confine it within a prediction window. The reader can understand this as the dashed vertical red line always being located after the grayed prediction window in this graphic.

AL index (SML), which is based on the AL (auroral lower envelope) index that gives the lower bound of the magnetic field strength (H-field) when all H curves from multiple stations are superimposed (Davis & Sugiura, 1966). Thus the SML index would be the AL index garnered from the superimposed SuperMAG curves. The SML algorithm used to isolate onset timestamp t_0 requires these four-fold conditions at a cadence of 1 min within a moving 30 min time window (Newell & Gjerloev, 2011):

$$\mathrm{SML}(t_0+1)-\mathrm{SML}(t_0) < -15\,\mathrm{nT} \quad (1)$$

$$\mathrm{SML}(t_0+2)-\mathrm{SML}(t_0) < -30\,\mathrm{nT} \quad (2)$$

$$\mathrm{SML}(t_0+3)-\mathrm{SML}(t_0) < -45\,\mathrm{nT} \quad (3)$$

$$\sum_{i=4}^{i=30} \mathrm{SML}(t_0+i)/26-\mathrm{SML}(t_0) < -100\,\mathrm{nT} \quad (4)$$

According to Equations 1–4, there must be a drop of 45 nT in the SML within 3 min of t_0 and then a subsequent mean drop over the next 27 min of 100 nT in the SML index are both necessary and sufficient conditions for the labeling of t_0 as the onset. The minimum allowed time to search for the next onset is 20 min, and more if the next onset is not found after exactly 20 min (Newell & Gjerloev, 2011). Broad criteria were employed to acquire as

large a number of usable onset sample windows as possible for this study, that were as varied as possible. Thus, the onsets occur between 00:00 and 23:59 magnetic local time, and between 60° and 80° in terms of magnetic latitude. No onset sample filtration took place in terms of outliers based on either SML values, or in terms of nightside auroral zone coordinate range restriction, as done by Maimaiti et al. (2019), because realistic data was desirable for the building of comprehensive models. Outliers and edge cases have demonstrably affected machine learning models' prediction performance (Demir & Sahin, 2023), however, they are a realistic part of dealing with data.

This study harnesses machine learning models with the ultimate objective of binary classification to identify the onset at the 121th min using 120 min sample history windows directly prior to the onset. The input data for the dependent variable is a binary label, with 1 representing the 121th min being an onset and 0 otherwise. The input for the independent variable is a matrix comprising 120 rows (for each of the minutes), and 5 columns (the SW-IMF parameters B_x, B_y, B_z, V_x, and N_p). It was established by Maimaiti et al. (2019) that a time history of this duration adequately captures the B_z trend prior to the prediction time. Figure 2, adapted from Maimaiti et al. (2019), illustrates a single data point as has been described. Shaded in gray is the 120 min by 5 variable input matrix (a single data point). The output binary label is also known since it is associated with the known onset timestamp for the positive cases, and with the absence of the known onset timestamps for the nil cases. The study by Maimaiti et al. (2019) generated data points based on the presence/absence of the known onset timestamps located within 60 min windows. To understand this, consider that if the onset was at 20:17, we could rightfully say that the candidate data point from 17:30 to 19:30 and the next candidate data point when sliding 30 min forward from 18:00 to 20:00 are legitimate as input samples, since the 1 hr window after 19:30 and 20:00 both contain the onset at 20:17. This means that a given data point may overlap with another data point based on the location of the onset within 60 min window, since the window is slid every 30 min forward in time to find whether or not the onset is in a 60 min window. The case explained above shows that the input data would share an overlap from 18:00 to 19:00 for each input sample. Though helpful in the generation of more data points, the overlap does not imply mutually exclusive and independent time history windows. The 60 min window is demarcated by first vertical dotted black lines in Figure 2, while the actual onset is denoted by the red dashed line. The four-fold SML index criteria previously outlined, affirm substorm activity. In the present study, the onset timestamp is always located at the 121th min, and not within a 60 min window. This ensures no overlap of data; each data point is thus unique. Following the protocol of Maimaiti et al. (2019), if there are more than 20 missing values in any of the 5 features for any of the nil or positive datapoint, then that data point is dropped. If there are less than 20 missing values for any of the features, then that data point is resampled with linear interpolation. The temporal structure of each interval is preserved. This means that the sample's elements are kept in place according to how they occurred temporally; they are not shuffled. The samples themselves (not their elements) are then shuffled, to avoid any possible seasonal trends if there were overlaps between the nil and positive cases. Vokhmyanin et al. (2019) mention that the OMNI data are time adjusted to the bow shock. The justifications and calculations for the time adjustment of solar wind and IMF data are outlined by King and Papitashvili (see https://omniweb.gsfc.nasa.gov/html/HROdocum.html). Further requirements for time adjustments can be found in Maimaiti et al. (2019), where a 10 min delay was chosen for the substorm data, after the authors had experimented with multiple delay times between 0 and 20 min. The reasons for this uniform time delay are threefold viz. the time taken for the solar wind to travel across the magnetosheath, the time taken for the magnetotail to be perturbed by the external disturbance at the magnetopause, and the time taken for the onset to be observed by ground based magnetometers. A pertinent deviation from Maimaiti et al. (2019), who used one of the standard practices in machine learning used to improve model performances in specific contexts, is noteworthy here. The authors normalized each of the SW-IMF variables via mean subtraction and subsequent division by the standard deviation. The mean and standard deviation were universal however, in that they were calculated using all the measurements spanning two decades. Given the fact that outliers are used in the data set, this may influence the normalization. Furthermore, normalizing data which is not expected to follow a normal distribution is not desirable. With more informed knowledge for this type of problem, each interval is not normalized. Rather they are scaled using a minimum to maximum scaler for each interval, to speed up data processing without using large numbers. The total number of samples acquired via the aforementioned filtration processes and sample handling methods is 76,292. These are broken down into the training, validation and test set samples as 53,404, 11,444 and 11,444 samples respectively. This corresponds to an approximate train:validation:test ratio of 70:15:15 in terms of percentages.

2.2. Technical Terms

In this section, metrics for model evaluation and performance used later on are defined (Lee, 2019; Murphy, 2022).

True Positive (TP): The model correctly predicts that the 120 min window before the substorm onset is indicative of the substorm onset in the minute after the window. This correlates with the fact that the substorm onset does occur at the 121th min.

False Positive (FP): The model incorrectly predicts that the 120 min window before the substorm onset is indicative of the substorm onset in the minute after the window. This correlates with the fact that the substorm onset does not occur at the 121th min.

True Negative (TN): The model correctly rejects that the 120 min window before the substorm onset is indicative of the absence of a substorm onset in the minute after the window. This correlates with the fact that the substorm onset does not occur at the 121th min.

False Negative (FN): The model incorrectly rejects that the 120 min window before the substorm onset is indicative of a substorm onset in the minute after the window. This correlates with the fact that the substorm onset occurs at the 121th min.

The following metrics are used to evaluate performance:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{5}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{6}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{7}$$

$$\text{F1-Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \tag{8}$$

the F1-score is the harmonic mean of precision and recall. The F1-score reaches the optimum value at 1, meaning perfect precision and recall, while the most undesirable score is achieved at a value of 0. To further understand a metric known as the **ROC AUC** (area under receiver operating characteristics curve), the following metrics used to define it are needed:

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{9}$$

this is the True Positive Rate (TPR), also known as *recall*. It is the proportion of actual positives that are correctly identified by a binary classification model.

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{10}$$

this is the False Positive Rate (FPR), which is the proportion of actual negatives incorrectly identified as positives by a model.

$$\text{TNR} = \frac{\text{TN}}{\text{TN} + \text{FP}} = 1 - \text{FPR} \tag{11}$$

this is the True Negative Rate (TNR), also known as *specificity*.

$$\text{FNR} = \frac{\text{FN}}{\text{FN} + \text{TP}} = 1 - \text{TPR} \tag{12}$$

Table 1
Models Deployed in This Study

Model abbreviation	Description
Conv2D	Two-dimensional convolutional neural network
LSTM	Long Short-Term Memory network
XGB	Extreme Gradient Boosting
BiLSTM	Bidirectional Long Short-Term Memory network
Conv2D-LSTM	Convolutional LSTM hybrid
Conv2D-BiLSTM	Convolutional bidirectional LSTM hybrid
Conv2D-XGB	Convolutional XGB hybrid
LSTM-XGB	LSTM XGB hybrid
BiLSTM32-XGB	Bidirectional LSTM (32 units) with XGB
BiLSTM16-XGB	Bidirectional LSTM (16 units) with XGB
Conv2D-LSTM-FCN16-XGB	Conv2D-LSTM with FCN (16 units) and XGB
Conv2D-BiLSTM32-FCN32-XGB	Conv2D-BiLSTM (32) with FCN (32 units) and XGB
Conv2D-BiLSTM32-XGB	Conv2D-BiLSTM (32 units) with XGB

this is the False Negative Rate (FNR), or the proportion of actual positives classified as negative by the model. The graph that plots the TPR against FPR, is known as the receiver operating characteristic (ROC). Furthermore, the area under curve (AUC or ROC AUC), is a single number that ranges from 0.5 to 1. A value close to 0.5 represents a classification probability that in no better or worse than random (tossing a coin e.g.), and value close to 1 indicates a perfect classifier. The AUC can further be interpreted as the probability that a randomly chosen sample from the substorm class is ranked higher than a randomly chosen sample from the non-substorm class.

$$\mathrm{HSS} = \frac{2[(\mathrm{TP} \times \mathrm{TN}) - (\mathrm{FP} \times \mathrm{FN})]}{(\mathrm{TP} + \mathrm{FN})(\mathrm{FN} + \mathrm{TN}) + (\mathrm{TP} + \mathrm{FP})(\mathrm{FP} + \mathrm{TN})} \tag{13}$$

this is the Heidke Skill Score (HSS), a metric that evaluates how much better a model is compared to random guessing. It is used for space weather applications (Haiducek et al., 2020). The HSS is bounded from −1 to 1, with 1 representing a perfect forecast, 0 a random forecast without any skill, and −1 representing the worst forecast out of all possibilities.

2.3. Models' Development

In this study, multiple machine learning models are trained to forecast the substorm onset prediction probability, at the 121th min, using the 120-min SW-IMF parameter history as input. Each of the models formulated in this study can be found in Table 1, which summarizes the models implemented in this study. All models besides the one labeled "Conv2D" are novel apropos substorm onset prediction in the literature. Note that each model outputs binary classification prediction probability values, using fully connected network layers (FCN), except where extreme gradient boosting (XGB) is stated, after being processed with other models in previous layers, in the case of stacked hybrid models. A detailed workflow diagram of the best performing model viz. Conv2D-BiSTM-XGB is given in Figure 3.

Straightforward models CNNs were first proposed by le Cun et al. (1989), and can have multiple layers to find patterns within the complexity of data that is either pixelated or vectorized, thus rendering them ideal for deciphering the spatio-temporal dynamics of SW-IMF data as presented in Figure 2. Further information pertaining to how the Conv2D models work, can be found in Dumoulin and Visin (2016). A specific application of a type of stacked (layered) Conv2D model known as ResNet can be found in Maimaiti et al. (2019).

As part of the recurrent neural network family, the goal of an LSTM network is to convert a sequence of inputs over time into a single, compact hidden state vector. This is accomplished through connections within the network that allow it to learn and remember patterns and relationships across different time steps in the sequence

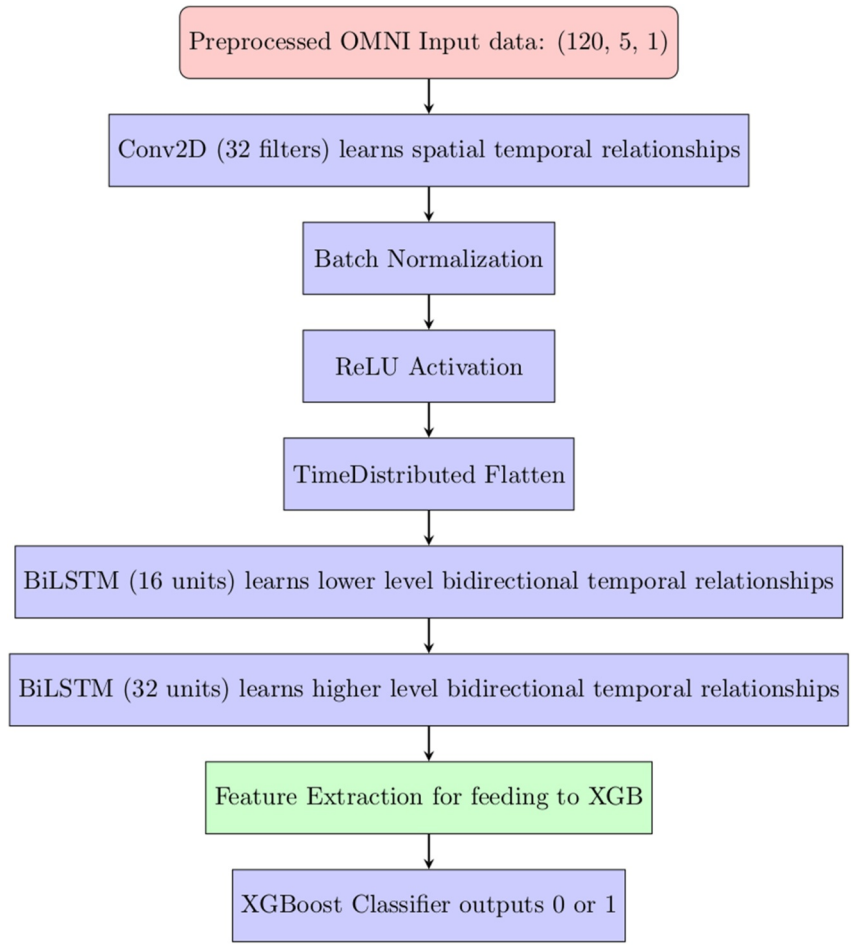

Figure 3. Workflow schematic of the best-performing Conv2D–BiLSTM32–XGB pipeline.

(Hochreiter & Schmidhuber, 1997). After the input data has been passed through Conv2D and LSTM layers, the FCN layer handles the binary classification probability prediction.

Extreme gradient boosting (XGB) builds an ensemble of decision trees for supervised learning tasks, and is often used for classification and regression. It combines multiple weak classifiers, yielding a strong binary classifier that can solve binary classification problems efficiently. It is an ensemble method that builds sequential decision trees to minimize a predefined loss function, such as binary logistic loss for classification tasks, as pursued in this study. This seminal modification of gradient boosting can be found in the work of Chen (2016).

Hybrid models. The BiLSTM network model applies LSTM bidirectionally for a given input sequence. In terms of time series pattern recognition, it can thus learn patterns "after the fact" viz. when the sequence has already happened, in both the forward and reverse directions, which is useful for looking at historical data.

The Conv2D-LSTM and Conv2D-BiLSTM models first process the input sequence via Conv2D and then pass that refined data onto LSTM to learn possible deeper patterns, before performing the final binary classification via the FCN.

All the models which contain "XGB" use gradient boosting instead of the FCN to perform the final binary classification probability prediction of the data. Where numbers are used in the model abbreviations, they can be understood as transferring the partially processed output between specific portions of the hybrid model. The number may represent the layer of the model with that specific amount of filters or neurons, depending on the layer from which data is acquired (the architectures of each model will be explained in more detail in the next subsection). To understand this, take as an example the model with the designation Conv2D-BiLSTM32-FCN32-

XGB. After going through some other layers, this model processes data with and subsequently from, 32 convolutional filters that have a two-dimensional setup. The extracted feature maps are then reshaped without into the correct input format for the BiLSTM32 component, which ultimately consists of 32 LSTM memory cells in both the forward and backward directions. This layer generates a concatenated hidden state vector of 64-dimensions. This output from the BiLSTM32 is then passed on to FCN32, which ultimately comprises a dense layer with 32 neurons. A dense layer applies a specific mathematical consisting of neurons, which apply a unique combination of a function and weighting to achieve the final classification prediction probabilities, or to reduce data dimensions (le Cun et al., 1989). Finally, these dimensionally reduced feature vectors are input to an XGB classifier to acquire the desired binary classification probability predictions.

2.3.1. Models' Architectures

Thirteen models were developed, with 11 of them being novel when compared to the literature at the time of writing, in so far as machine learning models that were developed for binary classification probability prediction. Maimaiti et al. (2019) employed a stacked Conv2D architecture based on ResNet. The Conv2D model used in this study cannot then be said to be an entirely novel approach. Furthermore, the XGB model performed poorly when implemented by itself but worked well when used only as the last stage binary classifier in hybrid models. Each model requires independent architectural details. Note that the models described here are not detailed explanations of the inner workings of generalized approaches, rather they are labeled as such and explained in context of how they are formulated. Generalized abstract concepts will be necessary to explain certain facets/layers of the models ad hoc.

Conv2D. The modified incorporation of a CNN-FCN model known as a residual neural network (ResNet) was carried out by Maimaiti et al. (2019). ResNet uses layered blocks of convolutional neural networks to find patterns in the data, and was conceived for image recognition by He et al. (2016), since the authors found that is was able to find detailed large and small scale patterns in a stable manner without undesirable gradient explosion or diminishment. Using CNN, albeit not in the exact same ResNet formulation, served as a useful starting point before moving on to other types of models, that would later be combined into hybrid models for this study.

The process begins with the SW-IMF data, which is first passed to an initial two-dimensional convolutional layer (Conv2D) employing 32 filters with a 3 by 3 kernel size and ReLU activation function. This function coverts negative values to zero and preserves positive vaues; it is chosen for computational efficiency and does not normalize the data. This layer extracts low-level spatial features by parsing the input with small, learnable windows. This is a technique in CNN that allows the model to identify basic patterns and local structures within the data. The output then flows into a second Conv2D layer with 64 filters, also using 3 by 3 kernels and ReLU activation. This builds upon the earlier features to capture more complex, hierarchical representations by combining lower level patterns detected in the previous layer.

Following feature extraction, a MaxPooling2D layer with a 2 by 2 pool size reduces data dimensions by selecting the maximum activation within each window. This downsamples the data without losing vital information within it. A regularization step is then applied via the first Dropout Layer, which randomly sets 25% of the activations to zero during model training. This assists to prevent overfitting.

The resulting three-dimensional feature maps are converted into a one-dimensional vector by the Flatten Layer. This reshapes data into the format appropriate for the Dense Layer (fully connected layer) containing 128 units with ReLU activation. A second, more aggressive Dropout Layer with a 50% rate is then applied to further regularize these dense representations before the final Output Layer condenses the 128-dimensional signal into a single logit value. A sigmoid activation function is used to produce a probability score between 0 and 1 with the logit value as input. The sigmoid function is a normalized exponential function that maps vectors onto a normal distribution, thus producing probabilities as output.

LSTM. LSTM is a specific type of recurrent neural network (RNN), which identifies patterns in time series data in the either forward or reverse (or both) directions (Salem, 2022). The LSTM architecture overcomes the unique vanishing gradient problem via gated mechanisms that regulate information flow in an optimized manner (Hochreiter & Schmidhuber, 1997). This is ideal for time-series classification tasks where the information from the past influences current predictions. The LSTM model process begins with an Input Sequence representing the time-series data. This input into a LSTM layer containing 32 memory units. This recurrent layer processes the

sequential input step-by-step temporally. However, it maintains an internal cell state that captures long-range temporal dependencies. The configuration set-up such that it returns only the final hidden state, which is a vector that encodes the entire sequence's contextual information.

Following the LSTM layer, a Batch Normalization operation is applied to stabilize and accelerate training by normalizing the activations across each mini-batch (Ioffe, 2015). It is important to note that batch normalization does not force normalization on data the way that scaling the data beforehand might have done. Rather, it used independently on data batches to speed up calculations and minimize errors. The normalized data passes into a fully connected layer with 32 units and ReLU activation. A final Output Layer with a single unit and sigmoid activation function then reduces the 32-dimensional representation into a single logit value. This is finally converted through the logistic function into a probability score between 0 and 1 for binary classification.

XGB The XGBoost model is trained to optimize computational resources and prevent the model from becoming too biased and overfitting. Thus, an early stopping mechanism is employed to monitor the model's performance on a mutually exclusive validation set with fewer samples. This system terminates the training process when the validation error (measured by logarithmic loss) fails to improve for a previously configured number of consecutive training cycles (boosting rounds). This ensures that the model maintains generalization capability without unnecessary computation.

The model incorporates two types of regularization parameters. These control its complexity and prevent overfitting. L2 regularization adds a penalty proportional to the squared magnitude of the model's weights. Thus, it encourages smaller weight values across all parameters, allowing for smoother and more stable solutions. Concurrently, L1 regularization, also imposes a penalty based on the absolute values of the weights. This tends to drive less important feature weights to exactly zero, thereby creating sparser models with built-in feature selection. These regularization techniques work in tandem to give desirable solutions.

The prediction process begins with an input feature vector which passes through a series of decision trees that have been sequentially trained. Each subsequent tree focuses on correcting the errors made by the previous ensemble members. Trees independently process the input features through a hierarchy of decision rules. This produces a numerical score that represents its contribution to the final prediction. All the individual tree scores are summed to create a combined score, which passes through a logistic transformation function that converts the unbounded numerical value into a probability value.

BiLSTM. The bidirectional LSTM (BiLSTM) model architecture process begins with an Input Sequence containing temporal data points that are fed into the first BiLSTM layer with 16 memory units in each of the forward and backward directions. This layer produces a sequence of hidden states which encode bidirectional temporal patterns. These are input to a second BiLSTM layer containing 32 units per direction. The model is thus enabled to learn more complex, higher-level temporal patterns built upon the features extracted by the preceding layer. A Batch Normalization layer is then applied and these features are then fed forward to a Dense Layer with 32 units and ReLU activation, concluding with an Output Layer containing a single unit with sigmoid activation, condensing the 32-dimensional representation into a probability score.

Conv2D-LSTM. The Conv2D-LSTM hybrid model process begins with an input tensor of dimensions 120 by 5 by 1, representing 120 time steps across 5 features with a single channel. This input first passes through a Conv2D layer with 16 filters of size 3 by 3 and ReLU activation, which extracts initial spatial patterns by applying convolutional operations across the temporal and feature dimensions. The results are then fed into a second Conv2D layer with 32 filters, maintaining the kernel size and activation function. This enables the model to learn more complex spatial patterns by building upon the features detected in the preceding layer. Following the CNN layers, a Batch Normalization layer is applied, followed by a ReLU activation layer. A Reshape operation takes place that transforms the three-dimensional convolutional output into a two-dimensional matrix with dimensions of 120 by 160, suitable for the LSTM. This reshaped sequence feeds into an LSTM layer with 16 memory units that processes the data step-by-step temporally, capturing sequential dependencies and long-term patterns across the 120 time steps. The LSTM's output then passes through a Flatten layer that converts the temporal patterns into a one-dimensional vector. This subsequently enters a Dense layer with 16 units and ReLU activation for nonlinear feature combination and abstraction. The architecture concludes with an Output layer containing a single unit with sigmoid activation, for binary classification.

Conv2D-BiLSTM. The Conv2D-BiLSTM hybrid model process begins with an Input tensor of dimensions 120 by 5 by 1, representing 120 temporal observations across 5 features with a single channel. This input first passes through a Conv2D layer with 32 filters of size 3 by 3 and ReLU activation, which extracts spatial patterns by applying convolutional operations across the temporal and feature dimensions to capture local patterns within the data. The resulting feature maps then proceed through a Batch Normalization layer, followed by an ReLU activation layer that reintroduces non-linearity to the normalized features. The architecture then employs a TimeDistributed Flatten operation that independently flattens the convolutional features at each time step while preserving the temporal sequence structure, transforming the three-dimensional convolutional output into a two-dimensional sequence suitable for BiLSTM. This flattened temporal sequence feeds into the first BiLSTM layer with 16 memory units in each direction configured to return the full sequence of hidden states, which then passes to a second BiLSTM layer with 32 units per direction configured to return only the final hidden state. This bidirectional temporal data subsequently enters a Dense layer with 32 units and ReLU activation, and concludes with an Output layer containing a single unit with sigmoid activation, which condenses the 32-dimensional representation into a probability score for binary classification.

XGB classification models. The Conv2D-XGBoost hybrid model process begins with an Input tensor of dimensions 120 by 5 by 1, representing 120 temporal observations across 5 features with a single channel. This input first passes through a Conv2D layer with 32 filters of size 3 by 3 and ReLU activation to capture spatial patterns. The results are then restructured for the XGBoost Classifier, which processes them through an ensemble of decision trees trained using gradient boosting. This ultimately produces a binary classification prediction probability.

The LSTM-XGBoost hybrid model process begins with Input Sequence Data representing temporal observations across multiple time steps. This input first passes through an LSTM layer with 32 memory units, to capture temporal dependencies and long-term patterns across the time series. The outputs then pass through a Batch Normalization layer. These normalized temporal features are transformed into a structured feature representation suitable for the XGBoost Classifier.

BiLSTM-XGB. The BiLSTM-XGBoost hybrid model process begins with the Input Sequence representing trends across multiple time steps. This first passes through a BiLSTM layer with 16 memory units in each direction, to capture bidirectional temporal dependencies and contextual information from both past and future time steps simultaneously. The outputs then proceed to a second BiLSTM layer with 32 units per direction, enabling the model to learn more complex patterns, without losing what was learned in the previous layer. The features then pass through a Batch Normalization layer, and these normalized features are subsequently transformed for the XGBoost Classifier.

The BiLSTM16-XGB model process involves the exact same implementation as above, however, it excludes both the second BiLSTM layer with 32 units per direction, and the Batch Normalization Layer.

The Conv2D-LSTM-FCN16-XGB (CNN-LSTM-XGB) hybrid model process begins with an Input Sequence representing temporal observations structured in a format suitable for spatial processing. This input first passes through a Conv2D layer with 16 filters, which captures spatial patterns. The results are then fed into a second Conv2D layer with 32 filters, so that the model can learn more complex patterns. Outputs then proceed through a Batch Normalization layer. A Reshape operation then transforms the three-dimensional convolutional output into a two-dimensional matrix format suitable for input to an LSTM layer with 16 memory units. The LSTM's output then passes through a Flatten layer that converts the data into a one-dimensional vector, which subsequently enters a Dense layer with 16 units. The features are then transferred to a Feature Extraction stage, where the outputs are transformed into a structure suitable for an XGBoost Classifier, which ultimately produces the binary classification prediction probabilities.

The Conv2D-BiLSTM-FCN32-XGB (CNN-BiLSTM-XGB) hybrid model process begins with an Input tensor of dimensions 120 by 5 by 1 representing temporal observations. This input first passes through a Conv2D layer with 32 filters, which captures spatial patterns. The results then proceed through a Batch Normalization layer and a ReLU Activation layer. A TimeDistributed Flatten operation then transforms the three-dimensional convolutional output into a two-dimensional sequence format suitable for input to a BiLSTM layer with 16 memory units, so that the model can learn bidirectional temporal dependencies. The outputs then feed into a second BiLSTM layer with 32 memory units, enabling the model to learn more complex bidirectional patterns. These features then enter a

Dense layer with 32 units. The combined spatiotemporal representations are then transferred to a Feature Extraction stage, where the neural network outputs are transformed into a structure suitable for an XGBoost Classifier, which ultimately produces the binary classification prediction probabilities.

The Conv2D-BiLSTM32-XGB model process involves a similar implementation to the previous Conv2D-BiLSTM-FCN32-XGB architecture. It excludes the Dense layer with 32 units and extracts features directly from the second BiLSTM layer's outputs to the XGBoost Classifier, to get the binary classification prediction probabilities.

Output probability values more than 0.5 are converted to 1 and 0 if otherwise for all models. It is important to note that probability calibration is necessary for any outputs of a model containing XGB as the final layer. This process is necessary for the comparison of the probability predictions with the actual binary labels, and is performed with a Platt Scaling equation that uses learned parameters from the validation data set (Ojeda et al., 2023).

2.4. Training Each Model

The deployment of the machine learning models took place with Google Colaboratory, which is compatible with the Python programming language. The A100 GPU (graphical processing unit) was used, which accelerated the modeling; this has been established as one of the competitive cloud computing resources available for the nature of problems such as the one in this project, where big data is used (Arifin et al., 2025). The Keras library that is integrated with TensorFlow was used to implement the models (Géron, 2022). Most of the models use a back-propagation algorithm (Rumelhart et al., 1986), which seeks to minimize the training loss using regularization to appropriately adjust weights according to gradient descent (iteratively updating weights). The exception is XGB, which uses gradient boosting, which is the addition of trees that approximate gradients (Chen, 2016). The objective of model training is to reduce the binary cross-entropy loss for each output (probability)-(actual) target pair. The actual values are represented by 1 for a substorm onset and 0 for a non-substorm onset. The binary cross-entropy loss is given in Equation 14:

$$\mathcal{L} = -[y \log \hat{y} + (1 - y)\log(1 - \hat{y})] \tag{14}$$

where $\mathcal{L}$ represents the scalar loss value quantifying the prediction error (lower is more desirable), y is the true binary label (1 for a substorm onset and 0 for a non-substorm onset), and $\hat{y} \in (0, 1)$ is the predicted probability. The gradient descent optimizer used within the models that incorporate CNN and LSTM is called "Adam" (Kingma, 2014). The default parameters maintained for Adam are the exponential decay rates $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a learning decay rate set at 0.0. The other key modified hyperparameters (tunable inputs) of the models are given in Table 2.

The hyperparameters are understood as follows: "Epochs/Rounds" refers to the number of complete training iterations or boosting trees used to fit the model, "Batch Size" is the number of training samples processed before updating the model's internal weights, "Max Depth" (always fixed at 5) is the maximum allowable depth of individual decision trees in boosting ensembles, λ is the L2 regularization penalty that discourages large weight magnitudes, and α is the L1 regularization penalty that encourages sparsity (both are fixed at 100) (Hastie et al., 2009).

2.5. Model Validation

Stratified K-Fold Cross-Validation is a technique used to evaluate the performance of a model. This sections draws largely on the book by Murphy (2022). It breaks up the data into mutually exclusive folds, while ensuring that each fold retains the original class distribution of the data set. A nuanced approach is needed for temporally structured data, where each data point is a time series matrix. It is essential to maintain the temporal structure within each matrix, and not to shuffle the data at each time step, as is done with other data. The 5-fold cross-validation setup uses a train/validation/test split of 80%/10%/10% as opposed to 70%/15%/15% (Russell & Norvig, 2016). This is because we need more data to be input for each fold in terms of training when we have 5 folds compared to the data required for training in a case run. The 70%/15%/15% results are useful for direct comparison with Maimaiti et al. (2019). We define the data partitions for each fold k as:

Table 2
Training Hyperparameters for Pure and Hybrid Models[a]

Model	Learning Rate(s)	Epochs/Rounds	Batch size
Conv2D	0.000001	150	128
LSTM	0.00001	45	128
XGB	0.0001	5,000	–
BiLSTM	0.000001	70	128
Conv2D-LSTM	0.000001	50	128
Conv2D-BiLSTM	0.000001	100	128
Conv2D-XGB	0.00001 (C) & 0.005 (X)	100 (C) & 340 (X)	128
LSTM-XGB	0.000001 (L) & 0.001 (X)	110 (L) & 1,500 (X)	100
BiLSTM32-XGB	0.000001 (B) & 0.005 (X)	70 (B) & 600 (X)	128
BiLSTM32-XGB	0.000001 (B) & 0.005 (X)	70 (B) & 600 (X)	128
Conv2D-LSTM-FCN16-XGB	0.000001 (CBF) & 0.001 (X)	60 (CBF) & 3,200 (X)	128
Conv2D-BiLSTM32-FCN32-XGB	0.000001 (CBF) & 0.001 (X)	100 (CBF) & 3,500 (X)	128
Conv2D-BiLSTM32-XGB	0.000001 (C) & 0.001 (X)	100 (C) & 3,500 (X)	128

[a]The alphabets or their combination refer to the individual model's components in hybrid models.

$$\mathcal{D}_k^{(\text{train})} = \bigcup_{j \in T_k} \mathcal{D}_j, \quad \mathcal{D}_k^{(\text{val})} = \bigcup_{j \in V_k} \mathcal{D}_j, \quad \mathcal{D}_k^{(\text{test})} = \bigcup_{j \in S_k} \mathcal{D}_j$$

Where:

- $T_k, V_k, S_k \subset \{1, \ldots, K\}$ are disjoint index sets for the training, validation, and test sets respectively,
- $T_k \cup V_k \cup S_k = \{1, \ldots, K\}$,
- $T_k \cap V_k = V_k \cap S_k = T_k \cap S_k = \varnothing$,
- The sets satisfy:

$$\frac{|T_k|}{K} \approx 0.80, \quad \frac{|V_k|}{K} \approx 0.10, \quad \frac{|S_k|}{K} \approx 0.10$$

The model is trained on $\mathcal{D}_k^{(train)}$ and evaluated it on $\mathcal{D}_k^{(val)}$. The same loss function is used in each fold.

In terms of implementation, the temporal distribution of the training, validation, and test data sets is handled as follows. The full data set consists of 76,292 independent samples, each being a 120-min by 5 SW-IMF parameter matrix. Although the samples are derived from a longer time series, they are treated as independent examples because the input windows are constructed as non-overlapping. Consequently, standard cross-validation can be applied without requiring strict chronological separation across folds. For the 5-fold stratified K-fold cross-validation, each fold initially assigns approximately 80% of the samples to the training set and 20% to the validation set, preserving the original 120 min temporal order of samples within each set. The validation set is then split internally into two equal halves: the first half serves as the validation set for monitoring training and early stopping, while the second half serves as the test set. This two-step process yields the final 80%/10%/10% split per fold. The temporal ordering ensures that no leakage occurs between either of the training, validation or test sets within each fold. The use of five folds also allowed the best results for the cross validation when explored in terms of the largest number of folds that could be used.

3. Results

In the previous sections, the focus was given to the data, its processing and the architecture, training and validation of the models. The problem was developed within the framework of binary classification prediction task. Multiple neural network models were developed to forecast the substorm onset at the next minute after 2 hrs input windows of SW-IMF data. The current section presents the models' convergence results, their forecasts, assesses their

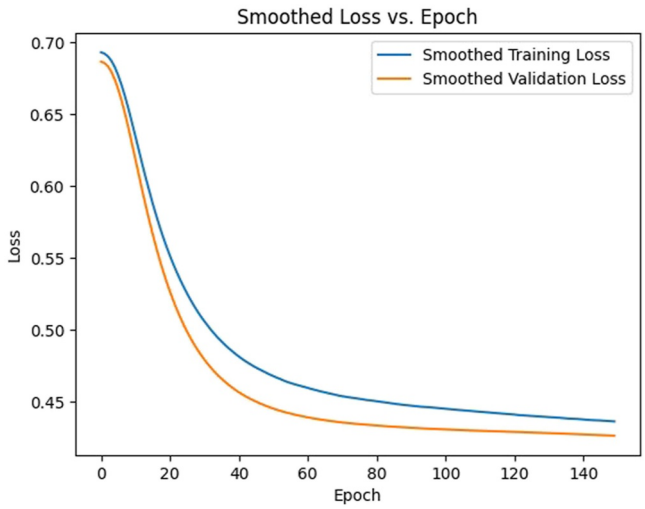

Figure 4. Training and validation loss curves for the two-dimensional convolutional model.

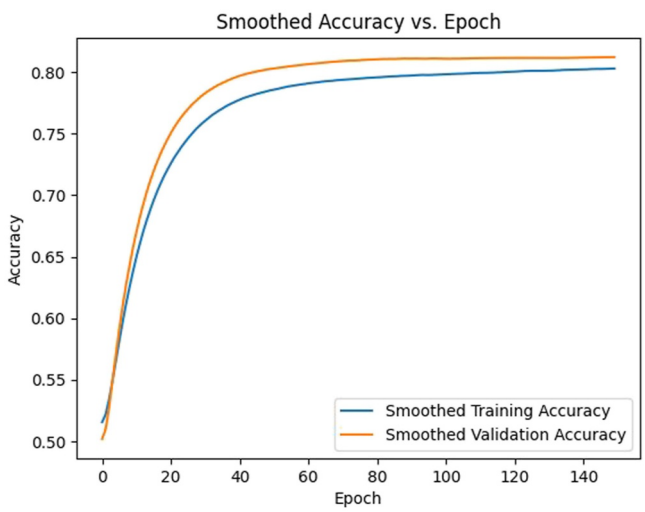

Figure 5. Training and validation accuracy curves for the two-dimensional convolutional model.

performance statistically, and discusses the importance of input parameters. The results in this section apply only to the test set, and on neither the training nor the validation sets of the data.

3.1. Training Results

Figures 4 and 5, respectively, show the loss and accuracy against the number of epochs for the two-dimensional convolutional model with blue for the training data and orange for the validation data. Both the training and validation losses decrease after the first 20 epochs, thereafter making a gradual approach to asymptotic behavior; with 150 epochs being sufficient for convergent behavior. The loss curves exhibit no signs of overfitting. This can be seen indicated by the loss and accuracy curves for the validation data persisting, respectively, below and above their correlates for the training data set.

The loss and accuracy curves for the rest of the models are provided in the Supporting Information S1. They exhibited similar behavior to those included here in terms of loss and accuracy respectively. There is no accuracy curve for hybrid models involving extreme gradient boosting for the final prediction step, as accuracy is a metric that cannot be directly optimized by XGB models, although the logistic loss can be optimized for. In general, all models showed similar convergent behavior without any apparent overfitting. The corresponding loss and accuracy curves for the remaining models are provided in Supporting Information S1.

3.2. Models' Performance and Validation

The models developed were used to predict the onset (or lack thereof) between 2003 and 2022. The evaluation metric ranges across all the models can be seen in Table 3, and in detail in the Table S1 of Supporting Information S1.

The class label of 0 refers to a 120 min interval before which no substorm (nonsubstorm) takes place at the 121th min, while the class label of 1 refers to the same window of time, however it is before a substorm onset at the 121th min. Note that these results stay stable across each instance that a given model is run. The total number of samples used is 76,292. These are broken down into the training, validation and test set samples as 53,404, 11,444 and 11,444 respectively. This corresponds to an approximate train to validation to test ratio of 70% to 15% to 15%, respectively. The evaluation metrics listed in these tables are based on an onset probability threshold such that probability values more than 0.5 are converted to 1 and 0 if otherwise for all models. Within each of the training, test and validation sample sets, there are even numbers of non-substorm and substorm cases, classed as 0 and 1 respectively. The order in which the models have been tabulated are in exact correspondence with their algorithmic descriptions, as outlined in Section 2.3. The models achieve F1-scores within a range of 0.80–0.83 for the nonsubstorm class and 0.78–0.83 for the substorm class, respectively. In terms of the best performance across both the classes, the Conv2D-XGB and Conv2D-BiLSTM32-FCN32-XGB model, performed equally well, achieving equal scores of 0.83 across each class. The result of each model can be found in the Table S1 of Supporting Information S1.

The models' performance can be affected varying the probability threshold between the minimum of 0 and maximum of 1, respectively, in a structured manner. This can be shown using their Receiver Operating Characteristic (ROC) curves. This plots the recall against the false positive rate at different thresholds. Hence, it is an effective means for the evaluation of models aimed at binary classification. The ROC curve corresponding to the two-dimensional convolutional model in this study is shown in Figure 6; the

Table 3
Models' Performance Ranges for Training, Validation, and Test Data Sets

Class label	Precision	Recall	F1-score	# Intervals
Prediction for training data				
0 (Nonsubstorm)	0.77–0.83	0.84–0.90	0.81–0.85	26,702
1 (Substorm)	0.83–0.88	0.75–0.83	0.80–0.84	26,702
Prediction for validation data				
0 (Nonsubstorm)	0.80–0.85	0.84–0.89	0.82–0.85	5,722
1 (Substorm)	0.83–0.88	0.78–0.84	0.81–0.85	5,722
Prediction for test data				
0 (Nonsubstorm)	0.76–0.82	0.84–0.89	0.80–0.83	5,722
1 (Substorm)	0.82–0.87	0.72–0.81	0.78–0.83	5,722

ROC curves for the rest of the models can be found in the Supporting Information S1.

As the threshold is increased, there is a corresponding increase in the recall; with an accompanied increase in the false positive rate. There can be advantages and disadvantages in the choice of a certain threshold, and it is ultimately left to the end user. For comparison, the ROC curve of any model that is a random guess predictor, is given by the diagonal dashed black line in Figure 6. This means that the further away from the diagonal line the blue curve is, the better the discrimination ability of the model is between onsets and non-onsets. This model achieves a ROC area under the curve (ROC-AUC) of 0.87. Across all models, the ROC-AUC values (see the ROC graphs in the Supporting Information S1) fall within the range of 0.87–0.93. The last of the metrics explored is known as the accuracy. This is calculated with the results of the confusion matrix elements, according to Equation 5. These elements appear in two by two matrix form (row and column) are the true negatives, false positives, false negatives and true positives. The full list of these values for each model can be found in the Table S2 of Supporting Information S1. Taking the two-dimensional convolutional model, the accuracy calculated from the TN, FP, FN and TP viz. 4,783, 939, 1,443 and 4,279 respectively, was 0.79. The accuracy range across the models is 0.79–0.83, with the lowest performing one being the standalone two-dimensional convolutional model and the highest, tied in rank, are the hybrid models (except the LSTM-XGB hybrid). This shows the models overall, misclassify about 20% of events. It is important to note that the confusion matrices' values from all models, which can be used to calculate skill scores are given in Table S2 of Supporting Information S1. The HSS values are discussed in Section 4.1.

The results of the stratified K-fold validation for each model are given in Table 4. The models are ranked by F1 score, which is the best metric for balanced data in binary classification problems (Sokolova & Lapalme, 2009). The italicized model indicates the only model (Conv2D) that has been used, albeit in a unique way, in the published literature (Maimaiti et al., 2019).

3.3. Variation in the Input Parameters

The dimensionality of the data, where each of the data points is a 120 row by 5 column matrix across substorm and nonsubstorm classes (1 and 0), is reduced via principal components analysis. The variation dependence and cumulative variance for each of the input parameters, along with the results of the PCA, reduced to three components, are shown in the graphs given in the multipanel Figure 7.

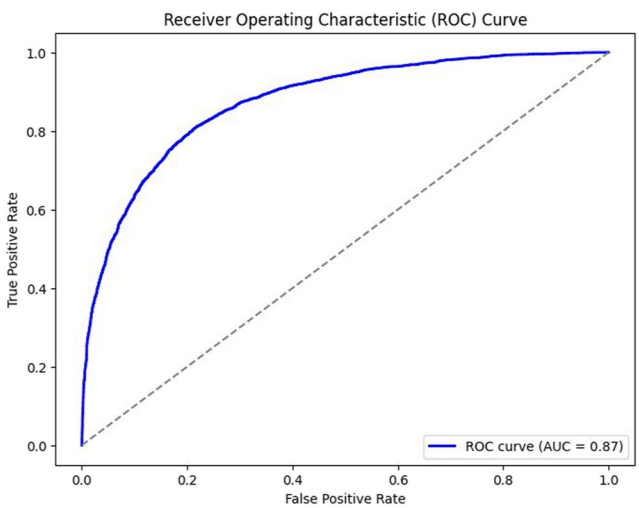

Figure 6. Receiver operating characteristic curve for the two-dimensional convolutional model. The dashed line represents a classifier that is no better than random guessing. The area under the curve was 0.87.

4. Discussion

In this study, multiple deep learning models to forecast the occurrence probability of magnetic substorm onsets over the next minute were developed. The rounded performance metric range intervals of results, given in the Table S1 of Supporting Information S1 and the graphs in the Supporting Information S1 respectively, fell within the following ranges for precision $= 82 \pm 2\,\%$, recall $= 82 \pm 1\,\%$, and ROC AUC $= 90 \pm 3\,\%$, respectively. Furthermore, the 5-fold cross validation results, in terms of the appropriate balanced classifier metric, was F1 $= 82 \pm 2\,\%$4. The decrease in the performance of the models' prediction results could be attributed to a strong overlap across both the onset and non-onset classes as shown by the PCA, the results of which are given Figure 7. In this section, further analysis of the results, along with a comparison of the results from previous studies, will take place.

4.1. Comparison With Existing Approaches

Early studies made use of neural networks to predict auroral indices using IMF and solar wind data (Gavrishchaka & Ganguli, 2001; Gleisner &

Table 4
Stratified K-Fold Cross Validation Results for Each of the Models[a]

Model	Average 5-fold F1-score	Rank
Conv2D-BiLSTM32-XGB	0.8411	1st
BiLSTM32-XGB	0.8410	2nd
Conv2D-XGB	0.8407	3rd
BiLSTM16-XGB	0.8403	4th
Conv2D-BiLSTM16-FCN32-XGB	0.8393	5th
Conv2D-LSTM-FCN16-XGB	0.8380	6th
Conv2D-LSTM	0.8351	7th
Conv2D-BiLSTM	0.8348	8th
BiLSTM	0.8307	9th
LSTM	0.8288	10th
XGB	0.8277	11th
LSTM-XGB	0.8264	12th
Conv2D	0.7920	13th

[a]The italicized model is a modified from a similar one extant in the literature (Maimaiti et al., 2019).

Lundstedt, 1997; Hernandez et al., 1993). Though foundational, these approaches faced limitations, since auroral indices are an uncertain proxy for substorm activity (Kullen et al., 2009). Furthermore, it has also been discovered that predicting short-term auroral variations is much more complicated when compared to average patterns (Luo et al., 2013). Subsequent work improved forecasts by developing data-derived coupling functions (Weigel et al., 2003), or by incorporating magnetospheric loading history into models (Barkhatov et al., 2017). This established context motivated a shift toward direct onset prediction. Studies around 2015 began this transition, with Tanaka et al. (2015) achieving ∼78% accuracy using support vector machines with a small, yet balanced data set (Tanaka et al., 2015). Later on, Maimaiti et al. (2019) later used a two-dimensional convolution model to achieve ∼75% accuracy. Recent work has increasingly utilized image data, with Yang et al. (2022) achieving 87.5% accuracy for detection (Yang et al., 2022) and Han et al. (2025) reporting metrics above 90% (Han et al., 2025). Building on these approaches, this study developed models that directly forecast substorm onset probability. In agreement with prior research, the incorporation of solar wind history has been crucial. Furthermore, it has also been demonstrated that the deep learning approaches outlined in this study effectively capture complex solar wind-magnetosphere interactions. This model thus makes a strong case as an alternative to those involving

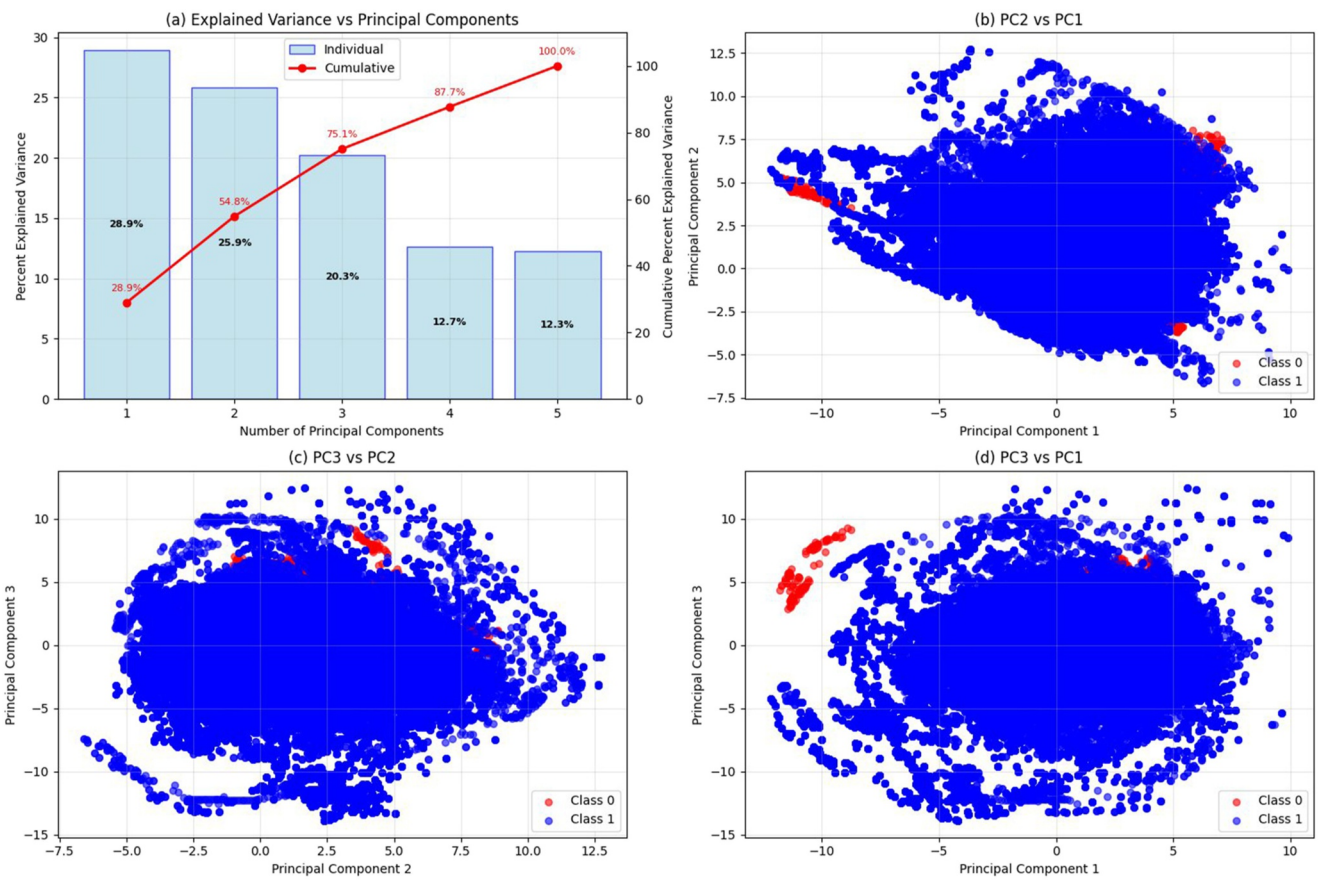

Figure 7. Principal Component Analysis (PCA) of magnetic field and plasma parameters. (a) Plot showing the percent explained variance versus number of principal components, where the individual variance contributions are displayed within each bar and cumulative variance is shown with red circles. The first three principal components explain approximately 75% of the total variance in the data set. (b) Projection of the two classes (0 and 1) onto the first two principal components (PC1 vs. PC2). (c) Projection onto the second and third principal components (PC2 vs. PC3). (d) Projection onto the first and third principal components (PC3 vs. PC1).

coupling functions. With the results acquired by the two-dimensional convolutional model in this study, a direct comparison can now be made with the established literature viz. the work of Maimaiti et al. (2019). Using a completely different model known as ResNet, and with the same five solar wind and IMF data variables for substorms spanning 1997–2017 based on the SuperMAG list of onsets, they acquired a balanced data set of about 61,000 samples. However, their approach was more conservative, since they used a 60 min onset containing window after the 120 min pre-onset window. This is in contrast to the onset always being located at the 121th min in this study. The metric results that they achieved were 72 ± 2% precision and 77 ± 4% recall rates and an ROC area of 0.83. The two-dimensional convolutional model in this study achieved 80 ± 3% precision and 79.5 ± 4.5% recall rates and an ROC area of 0.87, using about 75,000 balanced samples spanning 2003–2022. Thus there are minimum improvements of 7% in precision and 5% in recall when comparing the models directly. The rest of the models differ in their performance. All models achieved F1-scores in the 0.82–0.85 range on training data. These minor differences are indicative of consistent optimization across the different architectures. None of the models showed extreme overfitting on the training data alone, such as unrealistically high F1 scores.

Based on Table S1 of Supporting Information S1, some models can be grouped in terms of their performance results. These are the LSTM, BiLSTM and Conv2D-LSTM models, where the F1 score was approximately 0.82–0.83. There was also slightly lower recall for class 1, indicative of a slight bias toward class 0. The pure XGB model had F1 scores of 0.83–0.84, as well as very balanced precision and recall, which would generally indicate a good general baseline. However, in general, the XGB model by itself did not perform well over multiple boosting rounds, and performance was generally slow. The performance improved and was more stable for hybrid models that incorporated XGB and neural nets. Here the F1 scores improved slightly to 0.84–0.85 which was better than XGB itself. This could suggest that the hybridization of models improved discrimination between the classes without the disadvantage of overfitting. The best training performance came from Conv2D-XGB, BiLSTM32-XGB, Conv2D-BiLSTM16-FCN32-XGB, and Conv2D-BiLSTM32-XGB models, all of which reached mean F1 scores of about 0.84–0.845.

The best F1 score came from the Conv2D-BiLSTM32-XGB model. From the perspective of the structure of the data, the Conv2D layers extract spatial features from the OMNI input data, while the BiLSTM layers with capture long-range temporal dependencies like the growth-phase accumulation of magnetotail energy. XGBoost, applied to the concatenated deep features, then handles non-linear decision boundaries and feature interactions that would be suboptimal for a standard fully connected layer, making this hybrid architecture better suited for multivariate time series data with mixed spatiotemporal patterns. The model's superior performance reflects the known internal physical properties of substorms. Substorm onset is not a global simultaneous process but often begins in a localized sector and expands, and the Conv2D layers mimic the ability to detect this expansion by learning joint spatial patterns. Furthermore, the substorm growth phase can last 30–180 min (Milan et al., 2017), which means that current conditions do have a dependence on past solar wind driving. The BiLSTM's gating mechanism directly encodes this memory effect, which simpler models tend to miss. Thus, the model's best F1-score is not just a statistical value but aligns with the multi-scale, spatiotemporal, and threshold-dependent nature of substorms. The model thus indicates that deep learning models can be interpreted physically and not just in terms of feature structure importance.

The HSS scores are given in Table S2 of Supporting Information S1. Across all 13 models evaluated, these scores range from 0.573 to 0.665, indicating that every model performs significantly better than a no-skill random forecast. The top-performing group, achieving scores in the 0.65–0.67 range, includes eight ensemble-based architectures. The best overall model is Conv2D-BiLSTM16-FCN32-XGB with an HSS of 0.665, closely followed by Conv2D-XGB (0.660), Conv2D-BiLSTM (0.659), and Conv2D-BiLSTM32-XGB (0.658). Slightly behind but still within the tier are Conv2D-LSTM-FCN16-XGB (0.653), along with BiLSTM32-XGB, BiLSTM16-XGB, and Conv2D-LSTM (all tied at 0.651). These results demonstrate that hybrid architectures combining convolutional or recurrent layers with XGBoost consistently outperform standalone models. In the tier below this (0.60–0.64), standalone XGB achieves 0.640, while LSTM-XGB drops to 0.603. The weakest performances fall in the lowest tier (0.57–0.59), occupied entirely by non-ensemble models: BiLSTM alone (0.585), Conv2D alone (0.578), and the lowest overall, LSTM alone (0.573). Importantly, no model achieves a score above 0.70, and none fall close to 0 or to −1. The ensemble models, particularly those merging deep learning feature extractors with gradient boosting, provide a substantial improvement over single-architecture models.

A type of Conv2D-XGB was found in a nuclear physics study to perform better than the individual Conv2D and XGB models when they were deployed independently (Thongsuwan et al., 2021), as well as with success for image classification (Zou et al., 2025). With respect to BiLSTM-XGB, the approach has recently been used with success for load forecasting at electric vehicles charging stations (Mansour et al., 2026). Two types of Conv2D-BiLSTM-XGB models have been cited with success in recent separate conference papers, where they were used for electric vehicle load prediction (Liu & Zhou, 2024), and for the detection of malicious software attacks (Tripathi et al., 2025).

The aforementioned hybrid models that had the best training performance balanced both precision and recall for both classes effectively. The advantage of BiLSTM models is that they apply LSTM for both the forward and backward directions apropos timeseries data, which means that they are ideal for learning patterns "after the fact." The XGB classification at the final step in hybrid models could have improved due to meaningful patterns in the data that could have been learned by the different Conv2D, LSTM, BiLSTM and fully connected layers. This suggests that hybrid models improve performance because each component contributes somewhat independently to filtering the data for the identification of unambiguous patterns in the 120 min sample windows.

The cross validation results with respect to the 5-fold average F1 score range from 0.79 to 0.84. This suggests that the models are able to find consistent patterns using five mutually exclusive, yet equally sized, folds of data, with each fold maintaining the ratio of training, validation and test samples that are balanced across both the classes. The F1 score also suggests that the models, at worst, do not discriminate accurately in about 20% of the cases. This means that credence can possibly be given to the results of Maimaiti et al. (2019), where the authors' two-dimensional convolution model, which was accurate about 75% of the time, yet did not undergo cross validation. It also means that there is a gap in the models' abilities to discriminate between the classes, although this gap can be narrowed by the introduction and hybridization of other deep learning algorithms such as the LSTM and XGB architectures. The gap was narrowed by about 5% in terms of the F1 score as the models became more complex (see Table S1 of Supporting Information S1). When comparing the F1 scores on the training data to that of Maimaiti et al. (2019), the Conv2D model was 0.79 versus 0.74 and the best hybrid model Conv2D-BiLSTM32-XGB comparison was 0.84 versus 0.74. Similar results are achieved for the other models. The F1 score gap between the worst and best performing models was about 5%, while the gap between the second worst performing model and the best model was about 1.5%. This shows the advantage of using other architectures for this problem in improving performance. The Conv2D model, when combined with LSTM and BiLSTM, had about a 4% improvement over the Conv2D model when it was implemented alone in this study.

4.2. Factors Affecting Model Performance

The models search for patterns in the available data. Trends that are known to be prevalent in about 15 years (2003–2017) of the almost 20 years of this study' data (2003–2022) have been found by Maimaiti et al. (2019) with a basis in the literature. The authors found, via an epoch analysis that in pre-substorm windows, there was a drop toward the negative direction in the IMF B_z values closer to the prediction time, implying that their model was indeed capturing the growth phase patterns. Though not exclusively accepted in the literature, there is broad support for the necessary condition that triggers the substorm (see Section 1) being the accumulation of solar wind energy within the magnetotail (McPherron et al., 1973; Lyons & Nishimura, 2020). This could be a possible explanation for the models in this study being able to recognize patterns within the data so well.

The probabilistic nature of the models' predictions is useful and in the cases of the best performing models that use XGB for the final classification, is evidenced by the calibration curve results (see the Supporting Information S1), which show that the predicted onset probabilities closely align with the observed frequency of onset events across probability bins (Dormann, 2020). This calibration indicates that the models produce well-behaved probability estimates that can be interpreted directly as likelihoods of substorm occurrence, rather than requiring post-hoc rescaling. The calibration curves demonstrate that the hybrid models incorporating XGB exhibit particularly reliable probability estimates across the full range from 0 to 1, with minimal deviation from the ideal diagonal. The fact that the models are well calibrated also implies the absence of significant frequency bias, which refers to the tendency of a model to systematically over-predict or under-predict the occurrence of an event relative to its true base rate (Mavrogiorgos et al., 2024). A model with low frequency bias will have its mean predicted probability approximately equal to the observed fraction of positive events in the data set. This is particularly

valuable for operational forecasting where sustained over-prediction would lead to alert fatigue and under-prediction would result in missed events.

Acknowledging the uncertainties in the OMNI propagation delay is crucial for interpreting the model's performance, particularly when predicting onset probability at the 121th min from a 120-min input history. The OMNI data set employs a time-shifting algorithm to propagate solar wind and IMF measurements from the L1 Lagrange point to Earth's bow shock nose, but this propagation delay is subject to uncertainty that is especially pertinent for models that aim to predict the onset in the next 1-min (Blüthner et al., 2026; Papitashvili et al., 2014). Case and Wild (2012) conducted a large-scale statistical study comparing cross-correlation derived propagation delays with OMNIweb estimates, finding that while OMNI delays broadly agree with other methods for typical conditions, variations of up to approximately 30 min are possible, and there are intervals where different propagation methods yield significantly disparate estimates. More recently, a comprehensive analysis of about 50,000 Near Earth Solar Wind events demonstrated that approximately 50% of OMNI propagation delays have uncertainties under 5 min, while more than 5% exhibit differences exceeding 20 min from statistically estimated delays. These timing uncertainties directly impact the alignment of solar wind drivers with substorm onset signatures, meaning that an apparent onset prediction at the 121th min could, in reality, correspond to a driver that arrived several minutes earlier or later. Consequently, the models' reported predictive accuracies should be interpreted with these propagation uncertainties in mind, and future work could benefit from probabilistic approaches that marginalize over plausible time delay distributions or from incorporating ensemble propagation models.

Model performance assumes that historical solar wind and IMF data can clearly discern the underlying dynamics of substorms. It is vital to note that there issues with this assumption. There exist fine differences between the solar wind and IMF conditions that prevail before minor substorms, nonsubstorm intervals and pseudo-onsets (Kullen & Karlsson, 2004; Waters et al., 2025). The gap in the models' ability to identify clear patterns from the data during these types of events are in concordance with the finding that imaging data may have unclear resolution during low energy phases and events, so as to be considered unreliable for data processing (Kullen & Karlsson, 2004).

The models miss the distinctions between the two classes about 20% of the time, according to their cross validated F1 score metrics. The PCA analysis, given in Figure 7, was useful in highlighting this limitation. It revealed an overlap in the between the substorm and nonsubstorm classes. This is known to have a significant impact on model performance (Rezvani & Wang, 2023). The inherent problem in defining the two classes is based on the physics established apropos the timeseries phenomena. Since substorm phases and quiet time phases are not entirely understood, the distinction between them as was defined before preprocessing, from a pure machine learning perspective, becomes a heuristic, instead of clear cut criteria for class distinction. Heuristics are known to influence the results of neural network models since they rely on domain knowledge (Walczak & Cerpa, 1999).

The performance of the models in this study rely on the best available domain knowledge to provide the initial criteria required for class balancing. This is the SML criteria given in 2.1. Other criteria that use auroral electroject indices exist, and there is no consensus on the type of criteria to use based on the problem that is being investigated (Borovsky & Yakymenko, 2017). Yet another concern for model performance could be inherent in these criteria. This is whether the list itself is comprehensive and concise enough (error-free), on the basis of the criteria used themselves being verified. This means that errors in the models could arise from the onset list itself containing errors. The SuperMAG data base is updated to address issues by the repository personnel, and its structure is most conducive for the pursuit of machine learning approaches.

The present study, like that of Maimaiti et al. (2019), adopts the SuperMAG SML-based onset list originally developed by Newell and Gjerloev (2011). The authors' list is well-suited for machine learning because it provides a large, homogeneously derived catalog of approximately 35,000 onsets from 1997 to 2017. This overlaps with the data set used in this study, which spans 2003–2022. What is critical to note is that for forecasting, models trained on one list may not generalize perfectly to another because the label noise or bias differs. For example, lists based on auroral electrojet indices data often capture more localized intensifications, whereas SuperMAG SML indices data provide a globally averaged measure of the westward electrojet. The present study's reliance on the SuperMAG list ensures direct comparability with Maimaiti et al. (2019), but for operational forecasting, users should be aware that absolute onset timing and probability thresholds might shift if a different onset definition is adopted. Future work could benefit from multi-list ensemble training or from probabilistic labels that account for

inter-list disagreement, thereby making the model more robust to the inherent uncertainty in substorm onset definition.

The models also assume that the conditions the data points capture fully the solar wind and IMF conditions for the entire magnetopause. This is problematic when it comes to dynamic phenomena, since the inherent fixed data measurement and subsequent collection rate of the instruments used, may not be appropriate enough (Fuselier et al., 2024). The ability to capture the entire state of the magnetosphere's dynamics is another problematic assumption apropos the machine learning models' performance in this study. In theory this is certainly true since neural network models are supposed to be able to universally approximate results from fully connected (modified sigmoidal) data (Cybenko, 1989). It is clear that this will not be possible, since computational capabilities have their own inherent limitations. Equally important is the lack of data available on the internal magnetosphere, where it has now been established that the predominant primary reaction to magnetospheric changes is no longer the substorm (McPherron et al., 2008).

The patterns recognised by the models used for class discrimination may also not be able to be receptive toward a key feature in the observed trends of B_z. Even though most substorms are observed with a concurrent negative (southward) decrease in the IMF B_z (Kullen & Karlsson, 2004), observations of (sometimes successive) substorms with a concurrent positive (northward) increase, have been reported (Miyashita et al., 2011). These types of intervals, though fewer in number, decrease the performance of the models. A modification on the basis of (as yet not understood) a physics-based filter that sorts these types of intervals into the pre-substorm onset class is a gap that needs addressing.

Finally, another factor unrelated to those already mentioned, can also be responsible for the decrease in model performance. The 120 min interval may need to be enlarged or contracted, as it has been discovered that the interval of time needed to capture the releases of excess magnetospheric energy storage is indeed a variable period of time (Kullen & Karlsson, 2004; Miyashita et al., 2011). In terms of future work to incorporate a physics based filter with machine learning to this problem, Freeman and Morley (2004) proposed a minimal loading-unloading model where substorms occur periodically with an approximate intrinsic period range of 2.6–2.9 hr under constant solar wind driving. The model follows a recursive relationship that calculates the total solar wind energy accumulated over the time interval between two successive substorms. To embed this behavior into deep learning, future work could: (a) augment input features with a running estimate of the magnetotail's loading state computed by integrating the solar wind power input from the last known onset; (b) impose a refractory period regularization that suppresses predictions immediately following a substorm; or (c) use the authors' model as a physics-based baseline for comparison. However, as the authors note, strong non-linearity and uncertainties in solar wind measurements limit deterministic prediction, suggesting that hybrid physics-informed neural networks, rather than a purely physics-based approach, could represent future research potential.

Further research for operational space weather forecasting can also make use of ensemble models, which use a set of probability predictions to yield a final probability prediction within a reasonable confidence interval (Murray, 2018). This is an area of active research and it does not come without limitations in terms of the methods to use not being standardized for all problems as well as computational costs (Camporeale et al., 2018). It should be noted that the OMNI data cannot be used in real-time since the data set is by definition published after the fact. Using all models to get a weighted sum of the probabilities and then averaging them could be a crude ensemble approach with higher weights assigned to better performing models. A better improvement could be the use of Monte Carlo dropout, where multiple forward passes with dropout could yield a distribution of onset probabilities, bootstrap aggregating, where models are trained on resampled data subsets produce ensemble predictions. Following the ideas put forth by both Murray (2018) and Camporeale et al. (2018), an ensemble approach using the best performing model in terms of F1 score viz. Conv2D-BiLSTM32-XGB with dropout during inference, reporting both the mean onset probability and the 5th–95th percentile confidence interval, and issuing alerts only when the lower confidence bound exceeds a threshold could be explored. The closest these models can get to live operational utility would be to use appropriate and available live satellite data that is subjected to cumulative data window sampling and treatment. These rolling windows would encounter a delay, as they are subjected to carefully developed decision criteria on how to handle fixed rolling data windows due to say, gaps or missing data.

5. Conclusions

In this study, novel models to forecast the magnetic substorm onset, were developed. The list of substorm timestamps was acquired from the SuperMAG database. The problem to be addressed was the occurrence probability prediction of the substorm onset over the next minute, using the 120 min history of the solar wind and IMF parameters (V_x, N_p, B_x, B_y, and B_z) prior to the substorm onsets as inputs. Basic two-dimensional convolution, long short-term memory, and extreme gradient boosting architectures were developed. These served as the basis for the development of more complex hybrid architectures that incorporated two or more of them. The models were trained on a data set comprising pre-substorm onset intervals found in the SuperMAG database from 2003 to 2022. In addition, an equal number of intervals during this period that are not from the pre-substorm class were used as input to remove bias from the model. The data was partitioned into three disjoint sets, each of which are fractions of the overall data viz. the training set (70%), validation set (15%) and test set (15%) for individual case runs. The Stratified K-fold validation was performed by splitting the amount of samples within each fold to training set (80%), validation set (10%) and test set (10%) to allow more data for training per given fold. Overall, the models which could be cross-validated demonstrated stable predictive behavior on the test data, corresponding to approximately 82 $\pm$ 2 % in terms of precision, 82 $\pm$ 1 % in terms of recall, and 82 $\pm$ 2 % in terms of the F1 score. This suggests consistent performance across all architectures. The best performing models above an F1 score of 0.84, were the Conv2D-BiLSTM32-XGB, BiLSTM32-XGB, Conv2D-XGB, and BiLSTM16-XGB models, with F1 scores of 0.8411, 0.8410, 0.8407, and 0.8401 respectively. The metrics show that the models can forecast the occurrence of a majority of substorm onsets using the 2 hr data window beforehand. This observation aligns with the idea that most substorms are not activated by a directional reversal of B_z. A modified form of stratified K-fold cross validation, which preserved the temporal sequence of the data, was also implemented across 5 folds for each of the models. The F1 scores, which represent the best metric for the assessment of a binary classifier, fell within an approximate range of 82 $\pm$ 3 %. This provides unambiguous evidence that the models worked across the data when it was partitioned, even while maintaining the training, test and validation ratios within each fold that the data was partitioned into. Finally, principal component analysis reveals that the inputs provided to the models may not always be adequate enough to forecast all substorms. This may be due to other influences, such as potential internal magnetospheric instabilities.

Conflict of Interest

The authors declare no conflicts of interest relevant to this study.

Availability Statement

The substorm list used in this study can be acquired from https://supermag.jhuapl.edu/, while the OMNI data can be acquired from https://cdaweb.gsfc.nasa.gov/. The majority of analysis and visualization were completed with open-source software tools such as matplotlib graphing library (Hunter, 2007), IPython (Pérez & Granger, 2007), the pandas library (McKinney, 2010), and Google Colaboratory https://colab.research.google.com/. The Jupyter notebooks for the models in the paper are hosted at Zenodo and is preserved at https://doi.org/10.5281/zenodo.20069090 (Essop, 2026).

Acknowledgments
We thank the University of KwaZulu Natal for their funding. We acknowledge the substorm timing list identified by the Newell and Gjerloev technique (Newell & Gjerloev, 2011), the SMU and SML indices (Newell & Gjerloev, 2011); and the SuperMAG collaboration (https://supermag.jhuapl.edu/info/?page=acknowledgement). Likewise, we acknowledge the SuperMAG collaborators and the use of the NASA/GFSC Space Physics Data Facility's OMNIWeb and CDAWeb service.

References

Akasofu, S.-I. (1964). The development of the auroral substorm. *Planetary and Space Science*, *12*(4), 273–282. https://doi.org/10.1016/0032-0633(64)90151-5

Akasofu, S.-I. (2023). A new understanding of why the Aurora has explosive characteristics. *Monthly Notices of the Royal Astronomical Society*, *518*(3), 3286–3300. https://doi.org/10.1093/mnras/stac3187

Arifin, O., Azim, F., Hartati, Y., Widyawati, D. K., & Ariffin, A. L. A. K. (2025). Comparative study on the efficiency of deep learning model training in cloud environments: Google Colab vs AWS. *Decode: Jurnal Pendidikan Teknologi Informasi*, *5*(2), 324–332. https://doi.org/10.51454/decode.v5i2.1197

Baker, D. N., Pulkkinen, T., Angelopoulos, V., Baumjohann, W., & McPherron, R. (1996). Neutral line model of substorms: Past results and present view. *Journal of Geophysical Research*, *101*(A6), 12975–13010. https://doi.org/10.1029/95ja03753

Barkhatov, N., Vorob'ev, V., Revunov, S., & Yagodkina, O. (2017). Effect of solar dynamics parameters on the formation of substorm activity. *Geomagnetism and Aeronomy*, *57*(3), 251–256. https://doi.org/10.1134/S0016793217030021

Blüthner, G., Volwerk, M., Schmid, D., Nakamura, R., Temmer, M., Roberts, O., et al. (2026). Evaluating the OMNI database: Statistical analysis of time-shifted L1 data versus direct near-Earth solar wind observations. *Journal of Geophysical Research: Space Physics*, *131*(3), e2025JA034781. https://doi.org/10.1029/2025JA034781

Bodeau, M. (2015). Review of better space weather proxies for spacecraft surface charging. *IEEE Transactions on Plasma Science*, *43*(9), 3075–3085. https://doi.org/10.1109/TPS.2015.2441038

Bodeau, M., & Baker, D. N. (2021). *Effects of space radiation on contemporary space-based systems II: Spacecraft internal and external charging and discharging effects* (pp. 13–61). Space weather effects and applications. https://doi.org/10.1002/9781119815570.ch2

Borovsky, J. E., & Yakymenko, K. (2017). Substorm occurrence rates, substorm recurrence times, and solar wind structure. *Journal of Geophysical Research: Space Physics*, *122*(3), 2973–2998. https://doi.org/10.1002/2016JA023625

Boström, R. (1964). A model of the auroral electrojets. *Journal of Geophysical Research*, *69*(23), 4983–4999. https://doi.org/10.1029/JZ069i023p04983

Camporeale, E., Wing, S., & Johnson, J. (2018). *Machine learning techniques for space weather*. Elsevier. https://doi.org/10.1016/c2016-0-01976-5

Case, N., & Wild, J. (2012). A statistical comparison of solar wind propagation delays derived from multispacecraft techniques. *Journal of Geophysical Research*, *117*(A2). https://doi.org/10.1029/2011JA016946

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *ACM*, 785–794. https://doi.org/10.1145/2939672.2939785

Coster, A. J., & Yizengaw, E. (2021). *GNSS/GPS degradation from space weather* (pp. 165–181). Space Weather Effects and Applications. https://doi.org/10.1002/9781119815570.ch8

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, *2*(4), 303–314. https://doi.org/10.1007/BF02551274

Davis, T. N., & Sugiura, M. (1966). Auroral electrojet activity index AE and its universal time variations. *Journal of Geophysical Research*, *71*(3), 785–801. https://doi.org/10.1029/JZ071i003p00785

Demir, S., & Sahin, E. K. (2023). Application of state-of-the-art machine learning algorithms for slope stability prediction by handling outliers of the dataset. *Earth Science Informatics*, *16*(3), 2497–2509. https://doi.org/10.1007/s12145-023-01059-8

Dormann, C. F. (2020). Calibration of probability predictions from machine-learning and statistical models. *Global Ecology and Biogeography*, *29*(4), 760–765. https://doi.org/10.1111/geb.13070

Dumoulin, V., & Visin, F. (2016). A guide to convolution arithmetic for deep learning. *arXiv*. https://doi.org/10.48550/arXiv.1603.07285

Essop, A. (2026). Jupyter notebook models for the manuscript. *Zenodo*. https://doi.org/10.5281/zenodo.20069090

Freeman, M., & Morley, S. K. (2004). A minimal substorm model that explains the observed statistical distribution of times between substorms. *Geophysical Research Letters*, *31*(12). https://doi.org/10.1029/2004GL019989

Freeman, M. P., Forsyth, C., & Rae, I. J. (2019). The influence of substorms on extreme rates of change of the surface horizontal magnetic field in the United Kingdom. *Space Weather*, *17*(6), 827–844. https://doi.org/10.1029/2018SW002148

Fu, H., Yue, C., Zong, Q.-G., Zhou, X., Wang, S., Liu, J., et al. (2025). Statistical analysis of substorms with different time durations. *Earth and Planetary Physics*, *9*(6), 1177–1186. https://doi.org/10.26464/epp2025068

Fu, H., Yue, C., Zong, Q.-G., Zhou, X.-Z., & Fu, S. (2021). Statistical characteristics of substorms with different intensity. *Journal of Geophysical Research: Space Physics*, *126*(8), e2021JA029318. https://doi.org/10.1029/2021JA029318

Fuselier, S., Petrinec, S., Reiff, P., Birn, J., Baker, D., Cohen, I., et al. (2024). Global-scale processes and effects of magnetic reconnections on the geospace environment. *Space Science Reviews*, *220*(4), 34. https://doi.org/10.1007/s11214-024-01067-0

Gavrishchaka, V. V., & Ganguli, S. B. (2001). Optimization of the neural-network geomagnetic model for forecasting large-amplitude substorm events. *Journal of Geophysical Research*, *106*(A4), 6247–6257. https://doi.org/10.1029/2000JA900137

Géron, A. (2022). *Hands-on machine learning with Scikit-learn, Keras, and tensor flow*. O'Reilly Media, Inc.

Gleisner, H., & Lundstedt, H. (1997). Response of the auroral electrojets to the solar wind modeled with neural networks. *Journal of Geophysical Research*, *102*(A7), 14269–14278. https://doi.org/10.1029/96JA03068

Haiducek, J. D., Welling, D. T., Morley, S. K., Ganushkina, N. Y., & Chu, X. (2020). Using multiple signatures to improve accuracy of substorm identification. *Journal of Geophysical Research: Space Physics*, *125*(4), e2019JA027559. https://doi.org/10.1029/2019JA027559

Han, Y., Han, B., & Hu, Z. (2025). A tailored deep learning network with embedded space physical knowledge for auroral substorm recognition: Validation through special case studies. *Universe*, *11*(8), 265. https://doi.org/10.3390/universe11080265

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2nd ed.). Springer. https://doi.org/10.1007/978-0-387-84858-7

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on computer vision and pattern recognition* (pp. 770–778). https://doi.org/10.48550/arXiv.1512.03385

Hernandez, J., Tajima, T., & Horton, W. (1993). Neural net forecasting for geomagnetic activity. *Geophysical Research Letters*, *20*(23), 2707–2710. https://doi.org/10.1029/93GL02848

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Horne, R., Glauert, S., Meredith, N., Boscher, D., Maget, V., Heynderickx, D., & Pitchford, D. (2013). Space weather impacts on satellites and forecasting the Earth's electron radiation belts with SPACECAST. *Space Weather*, *11*(4), 169–186. https://doi.org/10.1002/swe.20023

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Ioffe, S. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv*. https://doi.org/10.48550/arXiv.1502.03167

Johnson, J. R., & Wing, S. (2014). External versus internal triggering of substorms: An information-theoretical approach. *Geophysical Research Letters*, *41*(16), 5748–5754. https://doi.org/10.1002/2014GL060928

Johnson Freese, J. (2007). *Space as a strategic asset*. Columbia University Press. https://doi.org/10.7312/john13654

Juusola, L., Østgaard, N., Tanskanen, E., Partamies, N., & Snekvik, K. (2011). Earthward plasma sheet flows during substorm phases. *Journal of Geophysical Research*, *116*(A10). https://doi.org/10.1029/2011JA016852

Kawai, M., & Hanaoka, S. (2025). Strategic national objectives in space: Strategic alignment framework. *Astropolitics*, *23*(2), 1–26. https://doi.org/10.1080/14777622.2025.2570428

King, R. D., Orhobor, O. I., & Taylor, C. C. (2021). Cross-validation is safe to use. *Nature Machine Intelligence*, *3*(4), 276. https://doi.org/10.1038/s42256-021-00332-z

Kingma, D. P. (2014). Adam: A method for stochastic optimization. *arXiv Preprint arXiv:1412.6980*. https://doi.org/10.48550/arXiv.1412.6980

Kullen, A., & Karlsson, T. (2004). On the relation between solar wind, pseudobreakups, and substorms. *Journal of Geophysical Research*, *109*(A12). https://doi.org/10.1029/2004JA010488

Kullen, A., Ohtani, S., & Karlsson, T. (2009). Geomagnetic signatures of auroral substorms preceded by pseudobreakups. *Journal of Geophysical Research*, *114*(A4). https://doi.org/10.1029/2008JA013712

Kumar, S., Pulkkinen, T., & Gjerloev, J. (2024). Magnetotail variability during magnetospheric substorms. *Journal of Geophysical Research: Space Physics*, *129*(3), 17–22. https://doi.org/10.1029/2023JA031722

le Cun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., & Jackel, L. (1989). Handwritten digit recognition with a back-propagation network. *Advances in Neural Information Processing Systems*, *2*. https://proceedings.neurips.cc/paper/1989/file/53c3bce66e43be4f209556518c2fcb54-Paper.pdf

Lee, W.-M. (2019). Supervised learning-classification using logistic regression. *Python® machine learning*, 151–175.

Liu, G., & Zhou, X. (2024). Electric vehicle load prediction based on CNN-BILSTM and XGBoost combined model. In *2024 China automation congress* (pp. 2601–2606). https://doi.org/10.1109/CAC63892.2024.10865472

Lui, A. (1996). Current disruption in the Earth's magnetosphere: Observations and models. *Journal of Geophysical Research*, *101*(A6), 13067–13088. https://doi.org/10.1029/96JA00079

Luo, B., Li, X., Temerin, M., & Liu, S. (2013). Prediction of the AU, AL, and AE indices using solar wind parameters. *Journal of Geophysical Research: Space Physics*, *118*(12), 7683–7694. https://doi.org/10.1002/2013JA019188

Lyons, L., Blanchard, G., Samson, J., Lepping, R., Yamamoto, T., & Moretto, T. (1997). Coordinated observations demonstrating external substorm triggering. *Journal of Geophysical Research*, *102*(A12), 27039–27051. https://doi.org/10.1029/97JA02639

Lyons, L. R., & Nishimura, Y. (2020). Substorm onset and development: The crucial role of flow channels. *Journal of Atmospheric and Solar-Terrestrial Physics*, *211*, 105474. https://doi.org/10.1016/j.jastp.2020.105474

Maimaiti, M., Kunduri, B., Ruohoniemi, J., Baker, J., & House, L. L. (2019). A deep learning-based approach to forecast the onset of magnetic substorms. *Space Weather*, *17*(11), 1534–1552. https://doi.org/10.1029/2019SW002251

Mansour, H. S., Mohamed, A. S., & Abdel Aziz, M. (2026). Electric vehicles charging stations load forecasting based on hybrid xgboost-BiLSTM model. *Scientific Reports*, *16*(1), 374. https://doi.org/10.1038/s41598-025-29739-z

Mavrogiorgos, K., Kiourtis, A., Mavrogiorgou, A., Menychtas, A., & Kyriazis, D. (2024). Bias in machine learning: A literature review. *Applied Sciences*, *14*(19), 8860. https://doi.org/10.3390/app14198860

McKinney, W. (2010). Data structures for statistical computing in python. *Scipy*, *445*(1), 51–56. https://doi.org/10.25080/Majora-92bf1922-00a

McPherron, R., Russell, C., Kivelson, M., & Coleman, P., Jr. (1973). Substorms in Space: The correlation between ground and satellite observations of the magnetic field. In *Proceedings of the chapman memorial symposium on magnetospheric motions* (pp. 190–195). https://doi.org/10.1029/RS008i011p01059

McPherron, R. L., Weygand, J. M., & Hsu, T.-S. (2008). Response of the Earth's magnetosphere to changes in the solar wind. *Journal of Atmospheric and Solar-Terrestrial Physics*, *70*(2–4), 303–315. https://doi.org/10.1016/j.jastp.2007.08.040

Milan, S. E., Clausen, L. B. N., Coxon, J. C., Carter, J. A., Walach, M.-T., Laundal, K., et al. (2017). Overview of solar wind–magnetosphere–ionosphere–atmosphere coupling and the generation of magnetospheric currents. *Space Science Reviews*, *206*(1), 547–573. https://doi.org/10.1007/s11214-017-0333-0

Miyashita, Y., Kamide, Y., Liou, K., Wu, C.-C., Ieda, A., Nishitani, N., et al. (2011). Successive substorm expansions during a period of prolonged northward interplanetary magnetic field. *Journal of Geophysical Research*, *116*(A9). https://doi.org/10.1029/2011JA016719

Morley, S. K., & Freeman, M. (2007). On the association between northward turnings of the interplanetary magnetic field and substorm onsets. *Geophysical Research Letters*, *34*(8). https://doi.org/10.1029/2006GL028891

Murphy, K. P. (2022). *Probabilistic machine learning: An introduction*. MIT press.

Murray, S. A. (2018). The importance of ensemble techniques for operational space weather forecasting. *Space Weather*, *16*(7), 777–783. https://doi.org/10.1029/2018SW001861

Newell, P., & Gjerloev, J. (2011). Evaluation of SuperMAG auroral electrojet indices as indicators of substorms and auroral power. *Journal of Geophysical Research*, *116*(A12). https://doi.org/10.1029/2011JA016779

Newell, P., Liou, K., Gjerloev, J., Sotirelis, T., Wing, S., & Mitchell, E. (2016). Substorm probabilities are best predicted from solar wind speed. *Journal of Atmospheric and Solar-Terrestrial Physics*, *146*, 28–37. https://doi.org/10.1016/j.jastp.2016.04.019

Newell, P., Sotirelis, T., Liou, K., Meng, C.-I., & Rich, F. (2007). A nearly universal solar wind-magnetosphere coupling function inferred from 10 magnetospheric state variables. *Journal of Geophysical Research*, *112*(A1). https://doi.org/10.1029/2006JA012015

Nishimura, Y., Lyons, L., Gabrielse, C., Donovan, E., & Angelopoulos, V. (2025). Transition between substorm auroral onset and expansion phase activity. *Journal of Geophysical Research: Space Physics*, *130*(4), e2025JA033770. https://doi.org/10.1029/2025JA033770

Nishimura, Y., Lyons, L., Zou, S., Angelopoulos, V., & Mende, S. (2010). Substorm triggering by new plasma intrusion: THEMIS all-sky imager observations. *Journal of Geophysical Research*, *115*(A7). https://doi.org/10.1029/2009JA015166

Ojeda, F. M., Jansen, M. L., Thiéry, A., Blankenberg, S., Weimar, C., Schmid, M., & Ziegler, A. (2023). Calibrating machine learning approaches for probability estimation: A comprehensive comparison. *Statistics in Medicine*, *42*(29), 5451–5478. https://doi.org/10.1002/sim.9921

Papitashvili, N., Bilitza, D., & King, J. (2014). OMNI: A description of near-earth solar wind environment. *40th COSPAR scientific assembly*, (Vol. 40, pp. C0–C1).

Partamies, N., Juusola, L., Tanskanen, E., & Kauristie, K. (2013). Statistical properties of substorms during different storm and solar cycle phases. *Annales Geophysicae*, *31*(2), 349–358. https://doi.org/10.5194/angeo-31-349-2013

Peng, Z., Wang, C., Yang, Y., Li, H., Hu, Y., & Du, J. (2013). Substorms under northward interplanetary magnetic field: Statistical study. *Journal of Geophysical Research: Space Physics*, *118*(1), 364–374. https://doi.org/10.1029/2012JA018065

Pérez, F., & Granger, B. E. (2007). IPython: A system for interactive scientific computing. *Computing in Science & Engineering*, *9*(3), 21–29. https://doi.org/10.1109/MCSE.2007.53

Pudovkin, M. (1991). Physics of magnetospheric substorms: A review. *Magnetospheric Substorms*, *64*, 17–27. https://doi.org/10.1029/GM064p0017

Rezvani, S., & Wang, X. (2023). A broad review on class imbalance learning techniques. *Applied Soft Computing*, *143*, 110415. https://doi.org/10.1016/j.asoc.2023.110415

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536. https://doi.org/10.1038/323533a0

Russell, S., & Norvig, P. (2016). *Artificial intelligence: A modern approach* (3rd ed.). Pearson Education.

Salem, F. M. (2022). *Recurrent neural networks*. Springer. https://doi.org/10.1007/978-3-030-89929-5

Seraj, A., Mohammadi Khanaposhtani, M., Daneshfar, R., Naseri, M., Esmaeili, M., Baghban, A., et al. (2023). Cross-validation. In *Handbook of hydroinformatics* (pp. 89–105). Elsevier. https://doi.org/10.1016/B978-0-12-821285-1.00021-X

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, *45*(4), 427–437. https://doi.org/10.1016/j.ipm.2009.03.002

Tanaka, T., Kitao, D., Sato, Y., Tanaka, Y., & Ikeda, D. (2015). Forecasting auroral substorms from observed data with a supervised learning algorithm. In *2015 IEEE 11th International Conference on E-Science* (pp. 284–287). https://doi.org/10.1109/eScience.2015.48

Thongsuwan, S., Jaiyen, S., Padcharoen, A., & Agarwal, P. (2021). ConvXGB: A new deep learning model for classification problems based on CNN and XGBoost. *Nuclear Engineering and Technology*, *53*(2), 522–531. https://doi.org/10.1016/j.net.2020.04.008

Torres, J. F., Hadjout, D., Sebaa, A., Martínez Alvarez, F., & Troncoso, A. (2021). Deep learning for time series forecasting: A survey. *Big Data*, *9*(1), 3–21. https://doi.org/10.1089/big.2020.0159

Tripathi, A., Dwivedi, A., Aleem, S. S., Zaman, M. S., Pandey, R., & Priyanka, N. (2025). Hybrid CNN-BiLSTM with XGBoost for enhanced DDoS detection in software-defined networks. In *2025 World Skills Conference on universal data analytics and sciences (WorldSUAS)* (pp. 1–9). https://doi.org/10.1109/WorldSUAS66815.2025.11199135

Vokhmyanin, M., Stepanov, N., & Sergeev, V. (2019). On the evaluation of data quality in the OMNI interplanetary magnetic field database. *Space Weather*, *17*(3), 476–486. https://doi.org/10.1029/2018SW002113

Walczak, S., & Cerpa, N. (1999). Heuristic principles for the design of artificial neural networks. *Information and Software Technology*, *41*(2), 107–117. https://doi.org/10.1016/S0950-5849(98)00116-5

Waters, J. E., Lamy, L., Milan, S., Walach, M.-T., & Chané, E. (2025). Auroral acceleration at the northern magnetic pole during sub-Alfvénic solar wind flow at Earth. *Journal of Geophysical Research: Space Physics*, *130*(1), e2024JA033056. https://doi.org/10.1029/2024JA033056

Weigel, R., Klimas, A., & Vassiliadis, D. (2003). Solar wind coupling to and predictability of ground magnetic fields and their time derivatives. *Journal of Geophysical Research*, *108*(A7). https://doi.org/10.1029/2002JA009627

Yang, Q. J., Ren, J., & Xiang, H. (2022). Spatio-temporal detection of auroral substorm based on deep learning. *Chinese Journal of Geophysics*, *65*(3), 898–907. https://doi.org/10.6038/cjg2022P0047

Zakharov, V., Chernyshov, A., Miloch, W., & Jin, Y. (2020). Influence of the ionosphere on the parameters of the GPS navigation signals during a geomagnetic substorm. *Geomagnetism and Aeronomy*, *60*(6), 754–767. https://doi.org/10.1134/S0016793220060158

Zou, X., Wang, Q., Chen, Y., Wang, J., Xu, S., Zhu, Z., et al. (2025). Fusion of convolutional neural network with XGBoost feature extraction for predicting multi-constituents in corn using near infrared spectroscopy. *Food Chemistry*, *463*, 141053. https://doi.org/10.1016/j.foodchem.2024.141053