NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCE
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

BSc THESIS

Short-term Wildfire Danger Forecasting Methods in the Mediterranean Using Deep Learning

Eleftheria I. Vrachoriti

Supervisors: **Manolis Koubarakis**, Professor, NKUA
Ioannis Papoutsis, Assistant Professor, NTUA
Principal Investigator, OrionLab NTUA/NOA
Ioannis Prapas, PhD Candidate, OrionLab NTUA/NOA
Spyros Kondylatos, PhD Candidate, OrionLab NTUA/NOA

ATHENS

JULY 2024

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Μέθοδοι Βραχυπρόθεσμης Πρόγνωσης Κινδύνου Πυρκαγιάς στη Μεσόγειο με τη Χρήση Βαθιάς Μάθησης

Ελευθερία Ι. Βραχωρίτη

Επιβλέποντες: **Μανόλης Κουμπαράκης,** Καθηγητής, ΕΚΠΑ
Ιωάννης Παπουτσής, Επίκουρος Καθηγητής, ΕΜΠ
Κύριος Ερευνητής, OrionLab ΕΜΠ/ΕΑΑ
Ιωάννης Πράπας, Υποψήφιος Διδάκτωρ, OrionLab ΕΜΠ/ΕΑΑ
Σπύρος Κονδυλάτος, Υποψήφιος Διδάκτωρ, OrionLab ΕΜΠ/ΕΑΑ

ΑΘΗΝΑ

ΙΟΥΛΙΟΣ 2024

BSc THESIS

Short-term Wildfire Danger Forecasting Methods in the Mediterranean Using Deep Learning

Eleftheria I. Vrachoriti

S.N.: 1115202000026

Supervisors: **Manolis Koubarakis**, Professor, NKUA
Ioannis Papoutsis, Assistant Professor, NTUA
Principal Investigator, OrionLab NTUA/NOA
Ioannis Prapas, PhD Candidate, OrionLab NTUA/NOA
Spyros Kondylatos, PhD Candidate, OrionLab NTUA/NOA

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Μέθοδοι Βραχυπρόθεσμης Πρόγνωσης Κινδύνου Πυρκαγιάς στη Μεσόγειο με τη Χρήση Βαθιάς Μάθησης

Ελευθερία Ι. Βραχωρίτη

Α.Μ.: 1115202000026

Επιβλέποντες: **Μανόλης Κουμπαράκης,** Καθηγητής, ΕΚΠΑ
Ιωάννης Παπουτσής, Επίκουρος Καθηγητής, ΕΜΠ
Κύριος Ερευνητής, OrionLab ΕΜΠ/ΕΑΑ
Ιωάννης Πράπας, Υποψήφιος Διδάκτωρ, OrionLab ΕΜΠ/ΕΑΑ
Σπύρος Κονδυλάτος, Υποψήφιος Διδάκτωρ, OrionLab ΕΜΠ/ΕΑΑ

ABSTRACT

Climate change has lead to extreme weather conditions in the Mediterranean region that are responsible for very frequent wildfire ignition. Machine Learning techniques, such as Deep Learning, have provided us with a series of models that are capable of accurately predicting the daily wildfire danger, based on satellite-derived data or data sourced from meteorological stations. Nevertheless, no efforts have been made on predicting the wildfire risk of a region for time periods longer than a day.

To this end, in this Bachelor's Thesis, we first attempt to combine widely used multi-step forecasting strategies with Deep Learning models to extend the forecast horizon up to 10 days. As an alternative third forecasting method, we employ a Deep Learning model that can exploit not just historical data, but weather forecasts as well.

To assess the performance of each method, we first test all methods at different forecast horizons using the maximum possible forecast window length. Then, we conduct an complimentary comparative study using a common window length in all three methods and steps.

Afterwards, we conduct an extensive experimental performance-based analysis to determine the optimum window length for each method and prediction step combination. The findings suggest that, in all methods, it is possible to reduce the forecast window to 25-40% of its original length, without compromising on the quality of predictions. Also, the results are in favor of the third forecasting technique, showing its superiority in both performance and robustness to changes in forecast window length and number of prediction steps.

Then, we perform an ablation study on groups of features of different sources and types, to evaluate their importance in multi-step forecasts. We found out that satellite-derived data and land cover data play a major role in producing accurate forecasts.

Finally, at an attempt to shed light on the inherent mechanisms of Deep Learning models that are used for wildfire danger forecasting tasks with different forecast lengths or horizons, and to discover the most influential wildfire drivers, we employ various Explainable Artificial Intelligence (xAI) techniques. We conclude that the 2 or 3 days before the wildfire ignition are of greatest importance, regardless of forecast window length or forecast horizon. However, some covariates that are conventionally considered important for ignition are not correctly associated with wildfire risk in the models studied, which implies that a deeper investigation must be done in their architecture.

SUBJECT AREA: Machine Learning

KEYWORDS: Machine Learning, Deep Learning, Wildfire Danger, Forecasting Methods, Explainable Artificial Intelligence (xAI)

ΠΕΡΙΛΗΨΗ

Η κλιματική αλλαγή έχει οδηγήσει στην εμφάνιση ακραίων καιρικών συνθηκών στην περιοχή της Μεσογείου που είναι υπεύθυνες για την συχνή πρόκληση πυρκαγιών. Τεχνικές Μηχανική Μάθησης, όπως η Βαθιά Μάθηση, μας έχουν προσφέρει μια σειρά από μοντέλα που είναι ικανά να προβλέψουν με ακρίβεια τον ημερήσιο κίνδυνο πυρκαγιάς, βασιζόμενα σε δεδομένα που έχουν συλλεχθεί από δορυφόρους ή από μετεωρολογικούς σταθμούς. Εντούτοις, δεν έχουν γίνει προσπάθειες για την πρόβλεψη κινδύνου πυρκαγιάς σε μια περιοχή για χρονικές περιόδους μεγαλύτερες από μια ημέρα.

Για αυτόν τον λόγο, σε αυτή την Πτυχιακή Εργασία, προσπαθούμε πρώτα να συνδυάσουμε ευρέως διαδεδομένες στρατηγικές πρόγνωσης με μοντέλα Βαθιάς Μάθησης για να επεκτείνουμε τον ορίζοντα πρόγνωσης μέχρι και 10 ημέρες. Σαν μια τρίτη εναλλακτική μέθοδο, χρησιμοποιούμε ένα μοντέλο Βαθιάς Μάθησης που μπορεί να εκμεταλλευτεί όχι μόνο ιστορικά δεδομένα αλλά και καιρικές προβλέψεις.

Για να εκτιμήσουμε την απόδοση κάθε μεθόδου, πρώτα εξετάζουμε όλες τις μεθόδους σε διαφορετικούς ορίζοντες πρόγνωσης χρησιμοποιώντας το μέγιστο δυνατό μήκος παραθύρου πρόγνωσης. Στη συνέχεια, διεξάγουμε μια συμπληρωματική συγκριτική μελέτη χρησιμοποιώντας ένα κοινό μήκος παραθύρου σε όλες τις μεθόδους και βήματα.

Έπειτα, διεξάγουμε μια εκτεταμένη πειραματική ανάλυση με βάση την απόδοση με σκοπό τον προσδιορισμό του βέλτιστου μήκους παραθύρου για κάθε συνδυασμό μεθόδου και βήματος πρόγνωσης. Τα ευρήματα δείχνουν ότι, σε όλες τις μεθόδους, είναι δυνατό να μειώσουμε το παράθυρο πρόγνωσης στο 25-40% του αρχικού του μήκους, χωρίς συμβιβασμούς στην ποιότητα των προβλέψεων. Επίσης, τα αποτελέσματα είναι ευνοϊκά προς την τρίτη προγνωστική μέθοδο, δείχνοντας την υπεροχή της σχετικά με την απόδοσή της και στην ευρωστία που έχει απέναντι σε αλλαγές στο μήκος παραθύρου πρόγνωσης και τον αριθμό βημάτων πρόβλεψης.

Κατόπιν, πραγματοποιούμε μια καταλυτική μελέτη πάνω σε ομάδες μεταβλητών διαφορετικών πηγών και τύπων, για να αξιολογήσουμε την σημασία τους στις προγνώσεις πολλών βημάτων. Ανακαλύψαμε ότι τα δορυφορικά δεδομένα και τα δεδομένα κάλυψης εδάφους κατέχουν κυρίαρχο ρόλο στην παραγωγή ακριβών προγνώσεων.

Τελικά, σε μια προσπάθεια να διαλευκάνουμε τους εσωτερικούς μηχανισμούς των μοντέλων Βαθιάς Μάθησης που χρησιμοποιούνται σε εργασίες πρόγνωσης κινδύνου πυρκαγιάς με διαφορετικά μήκη παραθύρων ή ορίζοντες, και για να ανακαλύψουμε τους πιο επιδραστικούς παράγοντες πρόκλησης πυρκαγιάς, αξιοποιούμε διάφορες τεχνικές Επεξηγηματικής Τεχνητής Νοημοσύνης (ΕΤΝ). Συμπεραίνουμε ότι οι 2 ή 3 ημέρες πριν την πρόκληση πυρκαγιάς είναι οι πιο σημαντικές, ανεξαρτήτως του μήκους του παραθύρου πρόγνωσης ή του ορίζοντα πρόγνωσης. Παρά ταύτα, κάποιοι παράγοντες που συμβατικά θεωρούνται σημαντικοί για την πρόκληση πυρκαγιάς δεν είναι σωστά συσχετισμένοι με τον κίνδυνο πυρκαγιάς στα μοντέλα που μελετήθηκαν, το οποίο υποδεικνύει ότι απαιτείται μια ενδελεχής έρευνα σε επίπεδο αρχιτεκτονικής.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Μηχανική Μάθηση

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Μηχανική Μάθηση, Βαθιά Μάθηση, Κίνδυνος Πυρκαγιάς, Στρατηγικές Πρόγνωσης, Επεξηγηματική Τεχνητή Νοημοσύνη (ΕΤΝ)

To my parents.

ACKNOWLEDGMENTS

First of all, I would like to express my sincere gratitude to my supervisor, Prof. Manolis Koubarakis, for his mentorship and expertise. From our very first conversation, he attentively and patiently listened to my thoughts and ideas, which he took into account to provide me the amazing opportunity to collaborate with an outstanding research group, OrionLab, under the supervision of Prof. Ioannis Papoutsis, on a very interesting and interdisciplinary topic.

Secondly, I would like to thank my supervisor, Prof. Ioannis Papoutsis, for making me feel like an equal member of his group. His expertise, guidance and trust in me were essential to the completion of this thesis.

Many thanks to Ioannis Prapas and Spyros Kondylatos, PhD Candidates and Researchers in OrionLab, for being my companions in this journey, for their tireless support, their suggestions for the development of this thesis and their weekly invaluable feedback in a both cooperative and friendly environment.

Last, but not least, I would like to express my gratitude to my parents for their unwavering support and encouragement during my undergraduate studies.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

PREFACE

This Bachelor's Thesis was conducted under collaboration with the OrionLab Research Group that specializes in applying Artificial Intelligence methods in Earth Observation. The OrionLab belongs to both the Remote Sensing Lab of the School of Rural, Surveying and Geoinformatics Engineering at the National Technical University of Athens (NTUA) and the National Observatory of Athens (NOA).

1. INTRODUCTION

Almost half (48.7%) of extreme wildfires occurring in the Mediterranean region are heat-induced, meaning that heatwaves and hot drought conditions are more likely to lead to the ignition of larger wildfires [1]. Due to climate change, it is expected that the frequency and intensity of fire weather conditions in the Mediterranean is going to increase [1, 2], potentially resulting in a greater wildfire ignition risk. This suggests the necessity of the development of intelligent systems and methods for making reliable, trustworthy predictions for wildfire danger that will be utilized in enhancing fire management strategies.

The application of Machine Learning methods, such as Deep Learning, has been proved to be very successful in wildfire prediction and management problems so far, leading to the increasing popularity of such techniques during the last decade [3]. However, most works in the field of wildfire forecasting using Machine Learning focus only on predicting next day's wildfire danger. In this work, we go one step beyond, combining multi-step forecasting methods with Deep Learning in order to extend the forecast horizon up to 10 days beyond the last observation. We also employ an alternative model that leverages both hindcasts and forecasts to extract a prediction. Subsequently, an elementary comparison of models with different forecast horizons and maximum possible forecast window length is presented, followed by a comparison of the same models, but this time, with a common predefined forecast window length. Then, an ablation study on the optimal forecast window length for each forecast horizon takes place, as an attempt to minimize the forecast window while compromising little to no performance.

Despite the great performance of Deep Learning methods, they are often hard to interpret due to their black-box nature [4]. Yet, unveiling the relations between fire drivers as they have been modeled at the end of training could enhance the comprehension of mechanisms and conditions behind fire ignition and help us create more efficient models.

To this day, very few efforts have been made in explaining wildfire prediction models in the Mediterranean region. In this context, our work attempts to shed light on the importance of different fire drivers in both next day's wildfire danger prediction models, as well as models of different forecast horizons. For this purpose, both model-agnostic and model-specific methods are employed to provide us with a concrete understanding of the intricacies of the models used.

2. BACKGROUND AND RELATED WORK

Conventionally, the Fire Weather Index (FWI) [5] is used to estimate the next day's fire danger. However, its calculation does not take into account factors such as the local topography, land coverage information of the area or human activity indicators, as it is based solely on meteorological-related data.

On the other hand, Machine Learning methods so far have exhibited excellent performance in modeling Earth Science-related problems, providing predictions for the occurrence and severity of extreme events, such as wildfires, floods and landslides. Advanced Machine Learning techniques, namely Deep Learning, possess the ability to effectively recognize intricate patterns and perform feature extraction on large datasets consisting of multiple different groups of data, as well as on data of different contexts. Regarding wildfire-related tasks, Reichstein et al. [6] have stated the benefits of employing such techniques for exploiting both spatial and temporal contexts of wildfires. In fact, Deep Learning methods have outperformed other Machine Learning techniques in wildfire risk and burned area size prediction tasks [3], proving their potential.

Zhang et al. [7] used Random Forests (RFs), Support Vector Machines (SVMs), Multi-layer Perceptrons (MLPs) and Convolutional Neural Networks (CNNs) to predict spatial wildfire susceptibility in the Yunnan region of China based on topography, climate, vegetation and human activity-related data over almost a decade. Results revealed that the CNN was superior to the rest of models in terms of performance, which is probably due to its sheer capability to utilize the information of neighboring cells, capturing the spatial context of the data. At a global level, Zhang et al. [8] studied the performance of 1D and 2D CNNs and MLPs over four different datasets divided by season of wildfire ignition. Once again, the 2D CNN was the model that achieved the highest accuracy. Additionally, they concluded that the maximum temperature, soil temperature and Normalized Difference Vegetation Index (NDVI) are the most important wildfire drivers, using Explainable Artificial Intelligence (xAI) methods, particularly Feature Permutation (FP).

Huot et al. [9] focused on the temporal context of wildfires and considered the task of daily and weekly aggregated fire danger prediction as a segmentation problem. For this purpose, they trained residual U-Nets, Convolutional Autoencoders and variations of these models on historical remote-sensing wildfire, terrain, vegetation and weather data in the United States. Their experiments indicate the advantage of the Autoencoder model and its variations across all tasks, though the weekly danger prediction task was much more challenging, as expected.

Prapas et al. [10] focused on the next day's wildfire danger prediction problem and identified the challenges that need to be addressed when treating it as a Machine Learning forecasting task. In addition to this, they extracted and published four datasets [11] of historical data of wildfires in Greece with higher spatial resolution, each based on a different context; pixel, spatial, temporal and spatio-temporal context, to be used for training different models. Their findings suggest that models that exploit the spatial or temporal context of wildfires, such as Long Short-term Memory (LSTM) and Convolutional LSTM (ConvLSTM) respectively, demonstrate an improved performance over those that rely solely on the pixel context, while the best results are yielded when both spatial and temporal contexts are used.

Kondylatos et al. [12] extended the area of interest to a larger region, that of Eastern Mediterranean and used datasets utilizing similar contexts, but this time omitting the pure

spatial context, while training the exact same models for the rest of contexts. Results showed improved performance of the Deep Learning methods compared to the FWI.

Moreover, they attempted to understand the inner workings of the LSTM model in order to analyze the interactions between different wildfire covariates and to determine the most important features using xAI methods. In particular, Shapley values (SHAP) were used to investigate the effect of different features on specific severe wildfire events, Partial Dependence Plots (PDPs) to examine the impact of the most conventionally important drivers on the model's prediction on average and Integrated Gradients (IGs) to visualize the temporal evolution of the Integrated Gradients of the model the days preceding the ignition. Overall, their findings indicated that the LSTM tends to base its predictions mostly on fuel-related features.

Building on this, Kondylatos et al. [13, 14] published Mesogeos, the largest available dataset of historical ignition points and burned area sizes of wildfires in the Mediterranean region. This dataset includes a wide range of both dynamic and static fire covariates in a datacube form, allowing for the extraction of specialized sub-datasets that can be applied to several wildfire-related Machine Learning tasks, including wildfire danger prediction and final burned area prediction tasks. They also published some baseline models for each track, including an LSTM, a Transformer, a Gated Transformer Network (GTN) and a U-Net.

Nevertheless, all aforementioned works treat static and dynamic fire drivers the same way, disregarding potential interactions and causality relations between features of different resolution. To this end, Shams Eddin et al. [15] proposed a CNN with two separate branches, while also using a Location-aware Adaptive Normalization Layer (LOAN). More specifically, static data are processed by the first branch with a standard 2D CNN, while dynamic data are fed to a 3D CNN that applies a normalization method based on the embeddings of the static data produced by the 2D convolutional layers in the first branch. This model was trained on the dataset of Prapas et al. [11] and achieved higher scores compared to the models presented in [10].

However, all works mentioned earlier focus only on daily wildfire danger forecasting. While predictions for shorter forecast horizons are more accurate and include less uncertainty in general, acquiring the fire risk for longer horizons would be invaluable for the timely allocation of needed resources. This would improve forestry management and wildfire mitigation strategies in fire-prone areas, such as the Mediterranean region. In this direction, based on previous works for daily wildfire forecasting [13, 14], we attempt to extend the forecast horizon up to 10 days by applying broadly used multi-step forecasting methods with Deep Learning models. We also employ an alternative approach that leverages not only past observations but also forecasts for producing predictions. An evaluation of the performance of all methods in all steps is done using the maximum possible forecast window length, as well as a common predefined forecast window length. Then, an ablation study on the forecast window length takes place, to determine the shortest possible window that does not compromise the quality of the forecasts. In addition to this, we also ablate different features in groups using models of different forecast horizons, to determine their importance for different number of steps. Lastly, we extend the application of xAI methods in [12] to the whole Mediterranean region, using models with different forecast horizons and models trained only using a subset of uncorrelated variables. Given that information, we are able to determine how the focus of the models differentiates as the forecast horizon shrinks or as the number of variables decrease.

3. DATA AND METHODS

3.1 Data

The data used in this thesis is a dataset extracted from the Mesogeos datacube [13, 14], intended to be used for training Deep Learning models for wildfire danger forecasting.

This dataset consists of 8547 positive samples and 17342 negative samples. The forecasting methods that we have employed utilize the temporal context of the data, i.e. each sample is a time series of 30 days preceding a fire event.

This dataset contains a total of 35 variables, out of which 24 are used as input for training. The input variables are either dynamic or static. The dynamic variables are either sourced of meteorological stations or satellite-derived, while the static variables are used to model indicators of human activity and local topography or land cover.

An overview of all input features used, along with their modality and source or type is given in **Table 3.1**.

Table 3.1: Mesogeos input variables used in this thesis

Variable	Dynamic/Static	Source/Type
t2m (maximum 2 meters temperature during the day)	Dynamic	Meteorological
d2m (maximum 2 meters dewpoint temperature during the day)	Dynamic	Meteorological
rh (minimum relative humidity)	Dynamic	Meteorological
tp (total precipitation)	Dynamic	Meteorological
sp (maximum surface pressure)	Dynamic	Meteorological
ssrd (average Surface Solar Radiation Downwards)	Dynamic	Meteorological
wind_speed (maximum wind speed)	Dynamic	Meteorological
lst_day (Land Surface Temperature during the day)	Dynamic	Satellite-derived
lst_night (Land Surface Temperature during the night)	Dynamic	Satellite-derived
ndvi (Normalized Difference Vegetation Index)	Dynamic	Satellite-derived
smi (Soil Moisture Index)	Dynamic	Satellite-derived
lai (Leaf Area Index)	Dynamic	Satellite-derived
roads_distance (minimum distance from roads)	Static	Human Indicators
population (population each year)	Static	Human Indicators
dem (elevation)	Static	Topography
slope	Static	Topography
lc_agriculture (fraction of agriculture in the pixel)	Static	Land Cover

lc_forest (fraction of forest in the pixel)	Static	Land Cover
lc_grassland (fraction of grassland in the pixel)	Static	Land Cover
lc_settlement (fraction of settlement in the pixel)	Static	Land Cover
lc_shrubland (fraction of shrubland in the pixel)	Static	Land Cover
lc_sparse_vegetation (fraction of sparse vegetation in the pixel)	Static	Land Cover
lc_water_bodies (fraction of water bodies in the pixel)	Static	Land Cover
lc_wetland (fraction of wetland in the pixel)	Static	Land Cover

In order to examine the distribution of input feature values across samples of both classes, we plot the Probability Density Function (PDF) of each feature based on the values observed in the last day of each sample, as shown in **Figure 3.1**.

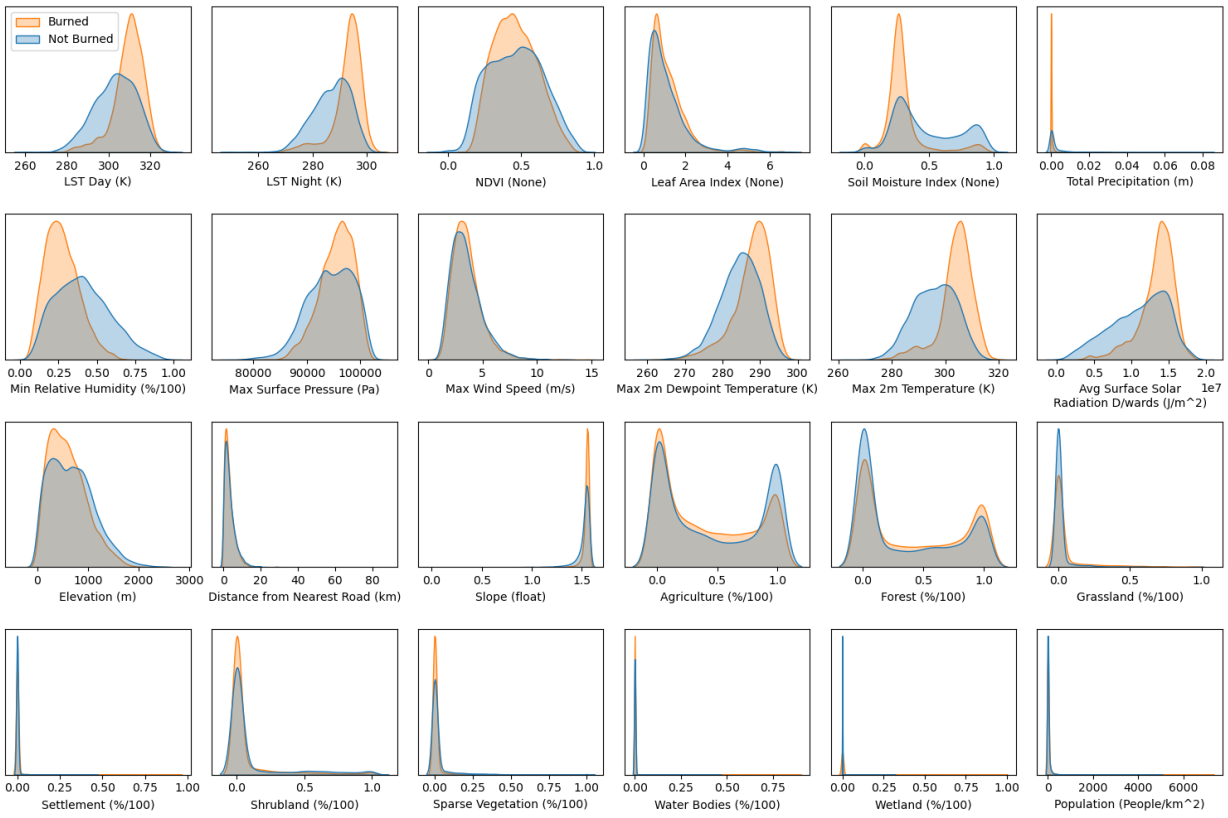


Figure 3.1: Probability Distribution Function (PDF) plots of input features in both classes based on the values observed in the last day

3.2 Models

The models used in this thesis are the same models as those mentioned in the Mesogeos paper [13], namely a Long Short-Term Memory (LSTM), a Transformer and a Gated Transformer Network (GTN) that uses the attention mechanism both in time and feature dimensions.

3.3 Loss Functions

In Next Day Forecasting and Direct Forecasting, the end goal is to classify each sample into the correct class, so the Cross Entropy loss function is used [13]. On the other hand, in Iterative Forecasting, even though the objective at the last step remains unchanged, the input feature values are forecasted in the rest of steps, which suggests that the forecasting loss should also be monitored. In this case, apart from the Cross Entropy loss function the Mean Squared Error loss function is also used.

3.3.1 Cross Entropy (CE) Loss

For a multi-class classification problem of N samples and M classes, the Cross Entropy loss function is defined using the following formula:

$$CE = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M}(y_{ij}\log\hat{y}_{ij}) \tag{3.1}$$

where $y_{ij} = 1$ if the actual class of sample i is j otherwise zero, and $\hat{y}_{ij}$ is the predicted probability that sample i belongs to class j.

In the special case of binary classification ($m = 2$), formula (3.1) can be rewritten as follows:

$$CE = -\frac{1}{N}\sum_{i=1}^{N}[y_i\log\hat{y}_i + (1-y_i)\log(1-\hat{y}_i)] \tag{3.2}$$

where $y_i = 1$ if the actual class of sample i is the positive class otherwise zero, and $\hat{y}_i$ is the predicted probability that sample i belongs to the positive class.

3.3.2 Mean Squared Error (MSE) Loss

For a regression problem of N samples, the Mean Squared Error loss function is defined using the following formula:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{3.3}$$

where y_i is the actual value of sample$i$and $\hat{y}_i$ the predicted value of sample i.

3.4 Metrics

The same metrics for binary classification as in [13] are used in order to be able to compare all models under exact the same circumstances.

3.4.1 Accuracy

The Accuracy metric is defined as the number of correct predictions for samples of both classes over the total number of samples.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{3.4}$$

where TP are the True Positive samples, FP the False Positive samples, TN the True Negative samples and FN the False Negative samples.

3.4.2 Precision

The Precision or Positive Predicted Value (PPV) metric is defined as the number of samples that were correctly classified as Positive over the total number of predicted Positive samples.

$$Precision = \frac{TP}{TP + FP} \tag{3.5}$$

A high Precision indicates a low False Positive Rate (FPR), meaning that the model makes pretty accurate predictions.

3.4.3 Recall

The Recall or True Positive Rate (TPR) or Sensitivity metric is defined as the number of correctly classified samples in Positive class over the total true Positive samples.

$$Recall = \frac{TP}{TP + FN} \tag{3.6}$$

A high Recall indicates a low False Negative Rate (FNR), meaning that the model predicts the majority of Positive samples correctly.

3.4.4 F1-score

The f1-score is the harmonic mean of the Precision and Recall metrics, defined by the following formula:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision - Recall} \tag{3.7}$$

The f1-score is a robust metric for an imbalanced dataset, like ours, where the Negative class is observed twice as frequent as the Positive class.

3.4.5 Area Under Precision-Recall Curve (AUPRC)

The Precision-Recall (PR) Curve demonstrates the relationship between the Precision and Recall metrics for different thresholds. It is used in binary classification problems and mostly for imbalanced datasets. An example is given in **Figure 3.2**.

The Area Under the Precision-Recall Curve (AUPRC) is a value that summarizes the predictor's performance and is not dependent on any thresholds or operating points [16].

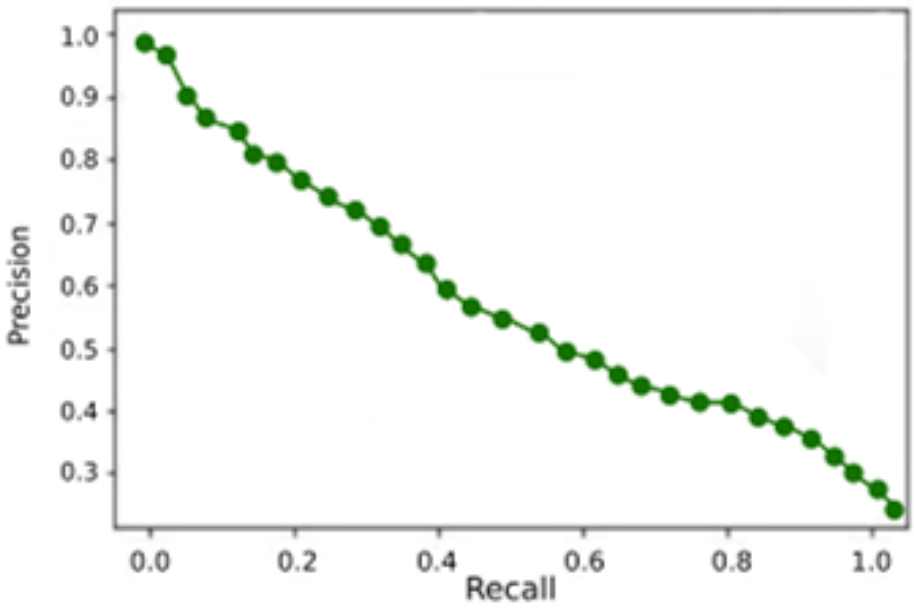


Figure 3.2: Example of a Precision-Recall Curve

3.5 Time Series Forecasting

3.5.1 Introduction to Forecasting

3.5.1.1 Forecasting

Forecasting refers to the practice of predicting the values of a variable y in the future based on past observations.

3.5.1.2 Forecast Window

The set of past observations Y used for forecasting is called forecast window or historical data or hindcasts. The number of observations L used is called forecast window length and is also often referred to as lag.

3.5.1.3 Forecast Origin

The last available observation y_t is called forecast origin.

3.5.1.4 Forecast Horizon

The number of steps predicted into the future H is called forecast horizon [17].

3.5.2 Forecasting Methods and Strategies

Forecasting methods are divided into subcategories based on the number of variables that are forecasted as well as on the length of the forecast horizon.

A short overview of forecasting methods classification can be found in **Figure 3.3**.

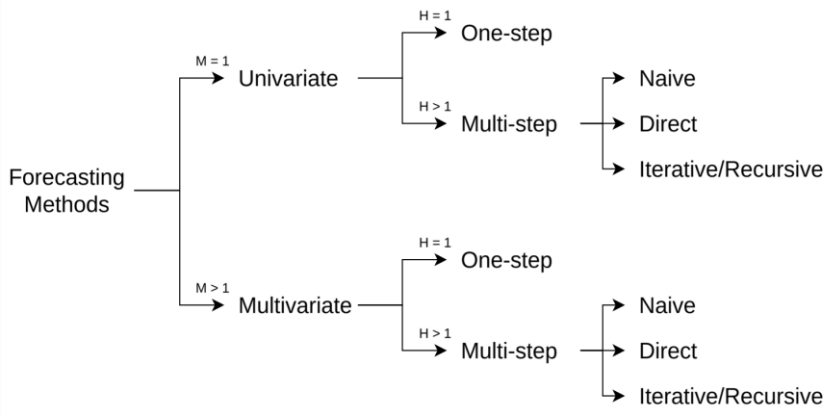


Figure 3.3: Short overview of forecasting methods classification. Diagram sourced from [19] and modified for this thesis' purposes.

3.5.2.1 Univariate Forecasting

Univariate forecasting methods are used to forecast the future values of a single variable y. In this case, the forecasting window$Y$is a vector consisting of L past observations of this particular variable $Y = [y_{t-L+1}, \ldots, y_{t-1}, y_t]$.

3.5.2.2 Multivariate Forecasting

Multivariate forecasting methods are used to forecast the future values of a set of M variables $y_1, y_2, \ldots, y_M$ at the same time. In this case, the forecasting window Y is a matrix of size $M \times L$ consisting of row vectors Y_i that contain L past observations for each variable i, $i = 1, \ldots, M$ [19]:

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \\ \ldots \\ Y_M \end{bmatrix} = \begin{bmatrix} y_{1,t-L+1} & \cdots & y_{1,t-1} & y_{1,t} \\ y_{2,t-L+1} & \cdots & y_{2,t-1} & y_{2,t} \\ \cdots & \cdots & \cdots & \cdots \\ y_{M,t-L+1} & \cdots & y_{M,t-1} & y_{M,t} \end{bmatrix} \tag{3.8}$$

3.5.2.3 One-step Forecasting

For univariate forecasting, one-step methods are used to forecast the future of a single variable y one step into the future, i.e. given the forecasting window $Y = [y_{t-L+1}, \ldots, y_{t-1}, y_t]$ to predict y_{t+H} where $H = 1$.

Similarly, for multivariate forecasting, the above definition is extended to forecasting the future of a set of variables $\{y_i\}$, $i = 1, \ldots, M$ one step into the future, i.e. given the forecasting window $Y = [Y_i]$, $i = 1, \ldots, M$ to predict $y_{i,t+H}$, where $H = 1$.

3.5.2.4 Multi-step Forecasting

The multi-step forecasting methods are a generalization of one-step forecasting methods for a larger number of steps. In this case, the forecast horizon is extended and all next $H > 1$ values are calculated.

For univariate forecasting, given the forecasting window $Y = [y_{t-L+1}, \ldots, y_{t-1}, y_t]$ the values $y_{t+1}, y_{t+2}, \ldots, y_{t+H}$, are predicted where $H > 1$ [18].

For multivariate forecasting, given the forecasting window $Y = \{Y_i\}$, $i = 1, \ldots, M$ the values $y_{i,t+H}$, are predicted where $H > 1$ [19].

3.5.2.4.1 Naive/Persistent Forecasting

The naive or persistent forecasting strategy [20] is the simplest forecasting strategy. It assumes that all forecasts have the same values as the forecast origin.

$$\hat{y}_{t+H} = y_t, \forall H \geq 1 \tag{3.9}$$

The above definition can be extended for multivariate forecasting as follows [19]:

$$\hat{y}_{i,t+H} = y_{i,t}, i = 1, \ldots, M, \forall H \geq 1 \tag{3.10}$$

resulting to the following matrix of predictions $\hat{Y}_{t+H}$:

$$\hat{Y}_{t+H} = \begin{bmatrix} \hat{y}_{1,t+H} \\ \hat{y}_{2,t+H} \\ \cdots \\ \hat{y}_{M,t+H} \end{bmatrix} = \begin{bmatrix} y_{1,t} \\ y_{2,t} \\ \cdots \\ y_{M,t} \end{bmatrix} =, \forall H \geq 1 \tag{3.11}$$

This forecaster is only intended to be used as a quality measure for the results of more advanced forecasting methods; a model exhibiting worse performance than that of the naive forecaster is considered a poor forecaster, while one that outperforms it is considered satisfactory.

3.5.2.4.2 Iterative/Recursive Forecasting

The Iterative or Recursive forecasting strategy [18] is based on a single one-step forecaster f that is used iteratively for a total of H times in order to produce all predictions until the forecast horizon is reached. At the end of each step, the most recent predictions are appended to the forecast window, while the oldest observations are removed, to ensure the lag remains unchanged.

For univariate forecasting, the iterative forecasting method can be formulated as follows:

$$\begin{aligned} &\hat{y}_{t+1} = f(y_{t-L+1}, \ldots, y_{t-1}, y_t) \\ &\hat{y}_{t+2} = f(y_{t-L+2}, \ldots, y_{t-1}, y_t, \hat{y}_{t+1}) \\ &\ldots \\ &\hat{y}_{t+h} = f(y_{t-L+h}, \ldots, y_{t-1}, y_t, \hat{y}_{t+1}, \ldots, \hat{y}_{t+h-1}), 1 \leq h \leq H \end{aligned} \tag{3.12}$$

In the case of multivariate forecasting, the above iterative scheme can be extended by using a one-step Multi-Input Multi-Output forecaster f in a similar way as in Direct multivariate forecasting:

$$\begin{aligned}
&\hat{Y}_{t+1} = f(Y_{t-L+1}, \dots, Y_{t-1}, Y_t) \\
&\hat{Y}_{t+2} = f(Y_{t-L+2}, \dots, Y_{t-1}, Y_t, \hat{Y}_{t+1}) \\
&\dots \\
&\hat{Y}_{t+h} = f(Y_{t-L+h}, \dots, Y_{t-1}, Y_t, \hat{Y}_{t+1}, \dots, \hat{Y}_{t+h-1}), h = 1, \dots, H
\end{aligned} \tag{3.13}$$

The iterative scheme can also be described with the following pseudocode:

Algorithm 1 Iterative Forecasting

Input: one-step forecaster $f: \mathbb{R}^{M \times L} \to \mathbb{R}$, forecast window $Y = \{Y_{t-L+1}, \dots, Y_t\} \in \mathbb{R}^{M \times L}$ (in univariate forecasting, row vector Y_i degrades to y_i) and forecast horizon $H > 1$
Output: forecasts $\hat{Y} = [\hat{Y}_{t+1}, \dots, \hat{Y}_{t+H}] \in \mathbb{R}^{M \times H}$

1: $\hat{Y} = \emptyset$
2: **for** $h = 1$ to H **do**:
3: $\quad \hat{Y}_{t+h} = f(Y)$
4: $\quad \hat{Y} = \hat{Y} \cup \hat{Y}_{t+h}$
5: $\quad Y = [Y_{t-L+1}, \dots, Y_t, \hat{Y}_{t+1}, \dots, \hat{Y}_{t+h-1}]$
6: **end for**
7: **return** $\hat{Y}$ as an $M \times H$ matrix

The Iterative forecasting strategy, while being intuitive and not requiring a lot of computational resources, it has a major downside; as the number of steps increases, more observations are being replaced by forecasts leading to the accumulation of prediction error. For this reason, the Iterative forecasting strategy is mostly preferred when the available data are noise-free [19].

3.5.2.4.3 Direct Forecasting

The Direct forecasting strategy [18] is based on using H independent forecasters f_h, each producing predictions at a particular step h, $h = 1, \dots, H$. In this case, the forecast window is common for all models and remains unchanged during the whole forecasting process.

For univariate forecasting, the direct forecasting method can be defined as follows:

$$\begin{aligned}
&\hat{y}_{t+1} = f_1(y_{t-L+1}, \dots, y_{t-1}, y_t) \\
&\hat{y}_{t+2} = f_2(y_{t-L+1}, \dots, y_{t-1}, y_t) \\
&\dots \\
&\hat{y}_{t+H} = f_H(y_{t-L+1}, \dots, y_{t-1}, y_t)
\end{aligned}$$

which can be summarized in the following formula:

$$\hat{y}_{t+h} = f_h(y_{t-L+1}, \dots, y_{t-1}, y_t), h = 1, \dots, H \quad (3.14)$$

In the case of multivariate forecasting, the above scheme can be extended by using H one-step Multi-Input Multi-Output forecasters f_h [19] that receive the forecasting window Y as input in the form of an $M \times L$ matrix and output a column vector $\hat{Y}_{t+h}$ containing the predictions for each variable i at step h:

$$\begin{aligned}
&\hat{Y}_{t+1} = f_1(Y_{t-L+1}, \dots, Y_{t-1}, Y_t) \\
&\hat{Y}_{t+2} = f_2(Y_{t-L+1}, \dots, Y_{t-1}, Y_t) \\
&\dots \\
&\hat{Y}_{t+H} = f_H(Y_{t-L+1}, \dots, Y_{t-1}, Y_t)
\end{aligned}$$

which can be summarized in the following formula:

$$\hat{Y}_{t+h} = f_h(Y_{t-L+1}, \dots, Y_{t-1}, Y_t), h = 1, \dots, H \quad (3.15)$$

This scheme can also be described with the following pseudocode:

Algorithm 2 Direct Forecasting

Input: H independent one-step forecasters $f_i: \mathbb{R}^{M \times L} \to \mathbb{R}$, $h = 1, \dots, H$,
forecast window $Y = [Y_{t-L+1}, \dots, Y_t] \in \mathbb{R}^{M \times L}$ (in univariate forecasting Y_i degrades to y_i)
and forecast horizon $H > 1$
Output: forecasts $\hat{Y} = [\hat{Y}_{t+1}, \dots, \hat{Y}_{t+H}] \in \mathbb{R}^{M \times H}$

1: $\hat{Y} = \emptyset$
2: **for** $h = 1$ to H **do**:
3: $\quad \hat{Y}_{t+h} = f_h(Y)$
4: $\quad \hat{Y} = \hat{Y} \cup \hat{Y}_{t+h}$
5: **end for**
6: **return** $\hat{Y}$ as an $M \times H$ matrix

Since neither any forecaster makes use of any of their predictions, nor does any forecaster relies on the predictions of forecasters used for a previous step, the prediction error at a particular step$h$does not influence the performance of the forecasters used for the rest of steps. This suggests that the Direct forecasting strategy is more robust to model misspecification [21]. However, this strategy is very computationally expensive, since it requires the training of H different forecasters.

3.5.3 Time Series Forecasting Using Machine Learning

Time series forecasting techniques can either be model-driven or data-driven. While model-driven methods are solely based on statistical approaches that are mathematically well-defined [18, 19], they lack the ability to learn the intrinsic relationships that lie within the given data. Conversely, data-driven methods are based on Machine Learning models that adjust to the data used. In this context, these models are trained under the supervised learning scheme, aiming at matching the model's output to the forecasts of the training data which are used as labels. This concept has so far been successfully applied yielding pretty accurate forecasts, especially in problems with a large number of available samples N, though the complexity of this models is very high, which hinders their explainability.

For univariate forecasting, one-step data-driven models produce a single output that corresponds to the prediction of the variable that is being forecasted. Artificial Neural Networks (ANNs) with a single node at the output layer are a perfect choice for this purpose. For multi-step forecasting, there are two choices; either use this model as a one-step forecaster in the Direct or Iterative forecasting strategy, or add H nodes at the output layer to obtain all H forecasts at once [19].

For multivariate forecasting, both one-step and multi-step data-driven methods ought to have a MIMO structure in order to be able to produce forecasts for all variables at once in the form of an M sized vector or an $M \times H$ matrix. Just like with univariate forecasting, multivariate one-step data-driven models can be used as one-step forecasters in the Direct or Iterative forecasting strategy. Possible Machine Learning models for multivariate data-driven forecasting are ANNs with multiple nodes at the output layer can be used, or more sophisticated architectures that exploit the temporal context of the data, such as RNNs, LSTMs or Gated Recurrent Units (GRUs).

3.6 Alternative Forecasting Approach Using Hindcasts and Forecasts

All aforementioned approaches are solely based on past observations. Usually, we can obtain forecasts for variables of meteorological interest. Given that the quality of these forecasts is good enough, we argue that we can achieve better performance by taking advantage of hindcasts and forecasts at the same time compared to employing the Direct or Iterative forecasting methods that were discussed earlier.

The problem is now formulated as follows: the goal is to feed hindcasts of M_h variables for T days, $[Y_1, Y_2, \ldots, Y_T]$ and forecasts of $M_f \leq M_h$ variables for the next H days, $[Y_{T+1}, Y_{T+2}, \ldots, Y_{T+H}]$ to a single model f in order to predict the wildfire danger of day $T + H$.

The model that we are going to use for this purpose is based on the Handoff Forecast LSTM, the model used in Google Flood Hub for flood danger forecasting [23] with the addition of extra components that perform dimensionality reduction on the input data [24].

Let hindcasts be a time series of T time steps, and forecasts another time series of H time steps. The Handoff Forecast LSTM presented in [24] consists of two separate LSTMs; a hindcast and a forecast LSTM that will process each of the two time series. In both LSTMs, the static variables are first fed to an encoder, a fully connected layer for each LSTM, to extract an embedding with less dimensions. The encoded static data are then concatenated with the dynamic variables, constructing a time series with less variables that will be processed by the LSTMs. After the hindcast LSTM processes the hindcasts

sequentially, a “handoff” happens to transfer the last cell state and hidden state $h_{Hind}(T)$ of the hindcast LSTM, to the forecast LSTM. However, because it is not always possible to obtain forecasts for all hindcast variables, the forecast variables might be fewer, meaning that the “handoff” between the two LSTMs cannot be straightforward. For this reason, a transfer network has to be used [22] to produce a new pair of cell and hidden states that will be the first cell $c_{Fore}(0)$ and hidden $h_{Fore}(0)$ state of the forecast LSTM. Then, the forecast LSTM processes the forecasts sequentially until the H-th step is reached. The output of the hindcast LSTM at time step T, as well as the output of the forecast LSTM at all time steps from $T+1$ to $T+H$ are directed to other components that can be used for regression or classification purposes.

A diagram of the Handoff Forecast LSTM’s architecture is shown in **Figure 3.4**.

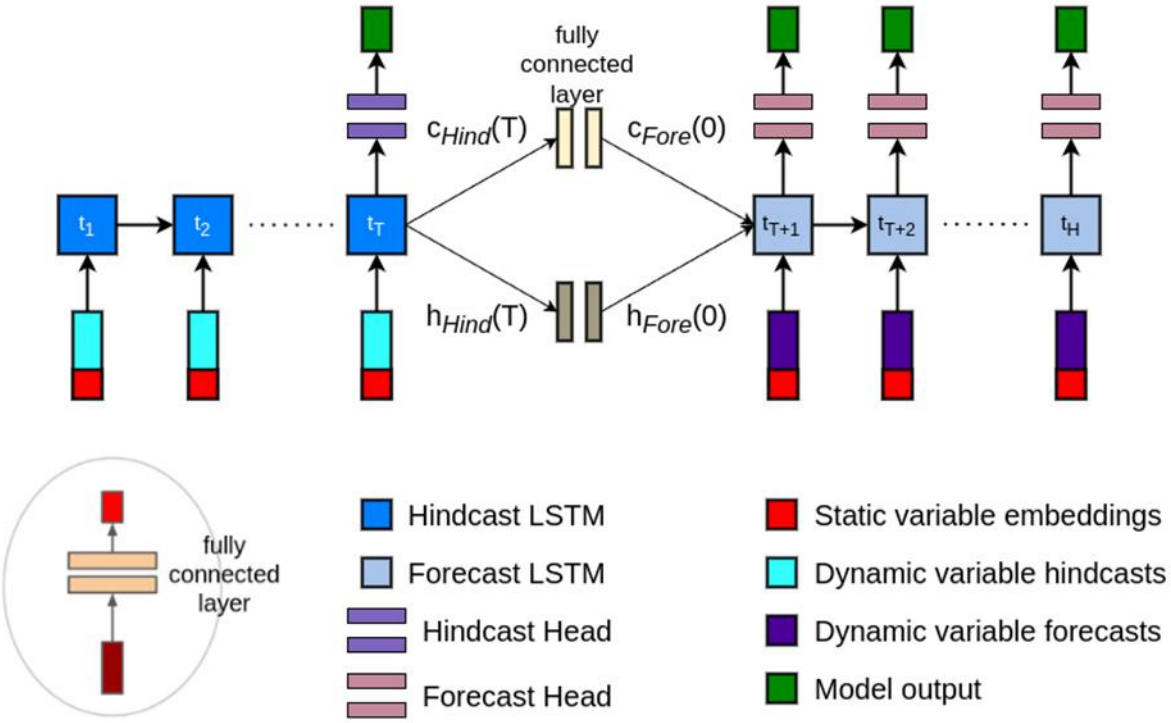


Figure 3.4: Handoff Forecast LSTM model architecture.
Diagram sourced from [24] and modified for clarity.

3.7 Explainable Artificial Intelligence (xAI)

3.7.1 Introduction to xAI

Machine Learning models range from simplistic models that are based on statistical principles, such as Logistic Regression, to very complicated models, such as cutting-edge Deep Learning techniques. In the former category of models, we can easily investigate their inner workings to comprehend and evaluate their decision-making abilities on specific input. Conversely, models with intricate inner representations of training data are much harder to interpret, which is why they are mostly used as black boxes. Hence, we resort to Explainable Artificial Intelligence (xAI) methods that try to peek through trained models to provide substantial explanations for the way they respond to the test data. Such knowledge is valuable when it comes to determining whether the models in question are suitable for the task they have been trained for, so that we can improve their performance or develop alternative and potentially more successful methods.

3.7.2 xAI Methods

xAI methods are divided into subcategories depending on their transparency, scope and generality.

A short, yet informative overview based on [25, 26, 27] can be found in **Figure 3.5**.

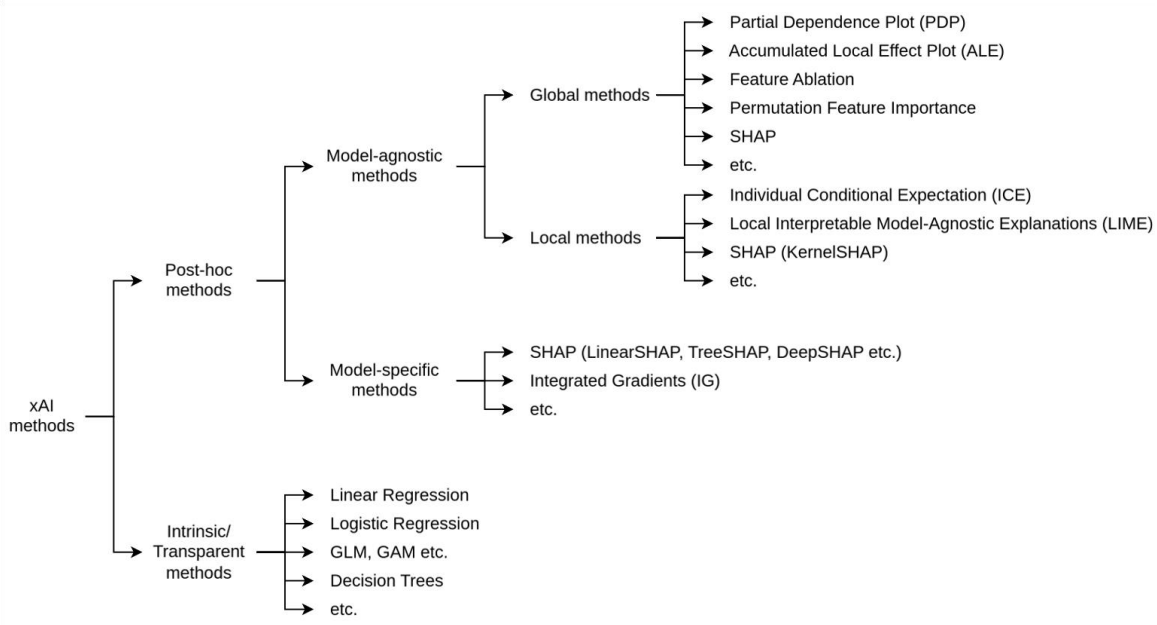


Figure 3.5: Short overview of the most used xAI methods

3.7.2.1 Intrinsic/Transparent vs Post-hoc xAI Methods

xAI methods can be divided into Intrinsic or Transparent methods and post-hoc methods depending on their transparency. The former group of methods focuses on reducing the complexity of Machine Learning models by training simpler, interpretable models for the same task [25], e.g. Linear Regression, Decision Trees, but might lead to compromising performance. On the contrary, post-hoc methods analyze already trained models.

3.7.2.2 Model-specific vs Model-agnostic xAI Methods

Post-hoc xAI methods can be divided into Model-specific and Model-agnostic methods depending on their specificity. Model-specific methods can only be applied to models of architecture that is assumed by the particular method used, while Model-agnostic methods can be applied to any model, which is very useful when it comes to comparing various models for the same task [25].

3.7.2.3 Global vs Local xAI Methods

Model-agnostic xAI methods can be divided into Global and Local methods depending on their scope. Global methods take all predictions into account to explain how different features influence the model's prediction on average. Conversely, Local methods focus on interpreting how individual predictions are made.

3.7.3 Feature Ablation

Feature ablation is a Global xAI method that is based on replacing each input feature with a baseline (e.g. zeros or a predefined value) and calculating the difference in the model's output or performance based on an evaluation metric (e.g. F1-score), which is then used to compute the attribution of each feature [28]. Positive attributions of a feature indicate that there is a positive correlation between values of this feature and the difference in the model's output or score.

3.7.4 Partial Dependence Plots (PDPs)

Partial Dependence Plots (PDPs) is a Global xAI method that visualizes the marginal effect of a feature on the model's prediction [25, 29].

Let f be the output function of the model and X be the set of all variables $X = \{x_1, x_2, \dots, x_M\}$. Let $Z \subset X$ be a set of variables of interest, $Z = \{z_1, z_2, \dots, z_P\}$, $z_i \in X$, $i = 1, \dots, P$ and C be the complimentary set of variables, i.e. $Z \cup C = X$, then the partial dependence function of f on Z is given by the following formula [29]:

$$f_Z = E_C[f(Z, C)] = \int f(Z, C)p(C)dC \tag{3.16}$$

where each subset Z has its own dependence function f_Z, which gives the average value of f when Z is fixed and C varies over its marginal distribution $p(C)$.

f_Z can then be estimated using the following sum:

$$\hat{f}_Z = \frac{1}{N}\sum_{i=1}^{N} f(Z, C_i) \tag{3.17}$$

where C_i represents the different permutations of values of variables in C.

In most applications, Z includes a single variable x_Z, in order to plot the effect of each individual feature on the model's output separately. In this case, x_Z values are displayed on the x-axis and the mean model outputs for each value are displayed on the y-axis.

An algorithm for producing the Partial Dependence Plot for a single feature x_Z as a scatter plot is described with the following pseudocode which is based on the Scikit-learn package's [30] implementation:

Algorithm 3 Partial Dependence Plot (PDP) for a single variable of interestx_Z

Input: model f, test set X, variable of interest x_Z
Output: partial dependence plot for x_Z

1: Let X' be a copy of X
2: **for** x_i **in** values of x_S **do**:
3: Substitute all values of x_S in X' with x_i
4: $\hat{Y} = f(X')$
5: $\mu = mean(\hat{Y})$
6: Plot (x_i, μ)
7: **end for**

The Partial Dependence Plots provide a visual, clear, causal interpretation of the effect of the features in question on the model's output [25]. However, each plot requires the usage of all samples, which increases the computational time of the method. A possible solution to this could be to apply this method to a subset of samples instead [29]. Besides that, this method provides the best results when the input data are uncorrelated. Otherwise, the calculation of the partial dependence of a feature might lead to the generation of input samples for the algorithm that are impossible to exist in the context of our problem, leading to biased estimations [25].

3.7.5 Integrated Gradients (IGs)

Integrated Gradients (IGs) is a model-specific xAI method that is applied to Deep Learning models to compute feature attributions. Suppose we have function $f: \mathbb{R}^n \rightarrow [0,1]$ representing a Deep Neural Network. Let $x \in R^n$ be the input at hand and $x' \in R^n$ be the baseline input, then we consider the straight-line path (in R^n) from the baseline x' to the input x and compute the gradients at all points along the path [31].

Integrated gradients are defined as the path integral of the gradients along the straight-line path from the baseline x' to the input x:

$$IG_i(x) = (x_i - x_i') \times \int_{a=0}^{1} \frac{\partial f(x' + a \times (x - x'))}{\partial x_i} da \tag{3.18}$$

For most Deep Neural Networks, it is possible to choose a baseline such that the prediction at the baseline is near zero, i.e. $f(x') \approx 0$.

The integral in (3.18) can be approximated via the following summation:

$$IG_i^{approximate}(x) = (x_i - x_i') \times \sum_{k=1}^{m} \frac{\partial f(x' + \frac{k}{m} \times (x - x'))}{\partial x_i} \times \frac{1}{m} \tag{3.19}$$

where m is the number of steps in the Riemman approximation of the integral.

Positive attributions indicate that the feature positively contributes to the model's prediction, while the magnitude of the attribution acts as a measure of the feature's contribution [28].

4. EXPERIMENTS

All experiments were run in a system with 2 NVIDIA GeForce RTX3080 GPUs [13].

4.1 Forecasting

4.1.1 Iterative/Recursive Forecasting

To apply the H-step Iterative forecasting strategy to our task, we first train a forecast LSTM layer with 128 neurons, that, given a time series with $30 - H$ days lag, predicts the feature values the $(30 - H + 1)$ -th day. After the training has completed, this model is used as a one-step forecaster in **Algorithm 1** to produce the forecasts of the variables in remaining days $T + 1,\ T + 2,$ ..., until the 30th day, $T + H$, is reached. Then, we can concatenate the initial time series with the forecasts of each intermediate step to construct a time-series with 30 days lag, which can be fed into any of the three baselines to reach our end goal, which is wildfire danger forecasting.

4.1.2 Direct Forecasting

To apply the H-step Direct forecasting strategy to our task, we just train a model with the same architecture as the three baseline models but with $30 - H$ days lag, as described in **Algorithm 2**. In this case, we do not perform any predictions on the features, but on the wildfire danger of the 30th day.

At this point, we should note that, even though the definitions given in **Section 3.5.2.3** imply that one-step Direct forecasting and next day forecasting are interchangeable, meaning that we should have used $30 - H + 1$ days lag instead of $30 - H$, we resort to the latter to ease the comparison of the Direct forecasting strategy with rest of methods.

4.1.3 Alternative Forecasting Approach Using Hindcasts and Forecasts

As an alternative forecasting approach, in this thesis we use the Handoff Forecast LSTM described in **Section 3.6**, with some adjustments in the input data used and the part of the architecture that handles the outputs of the LSTMs.

Data-wise, to apply H-step forecasting using the Handoff forecast LSTM, we need to ensure the availability of data that will be used as hindcasts and forecasts. However, since our dataset contains no forecasts, but only actual observations, we consider the first T days as hindcasts and the remaining H as forecasts of high quality.

In addition to this, in the hindcast LSTM all data are used as input, regardless of their modality or type. Conversely, as it is not possible to generate forecasts for satellite-derived data, only meteorological data and static data are fed into the forecast LSTM.

Architecture-wise, because only labels for the 31st day are available in our dataset, only the output of the forecast LSTM at time step $T + H$ is used. This output is forwarded to a classification head that is the exact same as the one used in the LSTM baseline in [13].

4.1.4 Comparison of Different Forecasting Methods

Initially, we train models for all forecasting methods outlined for H-steps, $H = 1,2,3$ for Iterative forecasting and $H = 1,2,\ldots,10$ for Direct forecasting and Handoff Forecast LSTM, using the maximum possible lag $30 - H$ for each step.

Nevertheless, on this basis, the number of steps and lag are changing at the same time, making comparisons of performance at different steps harder, because the effect of each of the two factors on the performance is not clear. Therefore, we repeat the training and testing process using a constant lag of 20 days for all methods and steps. A visualization of this method can be found in **Figure 4.1**.

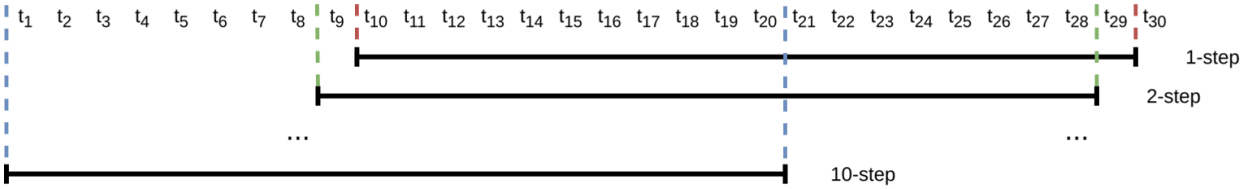


Figure 4.1: Visualization of experiments with 20 days lag

4.1.5 Ablation Study Based on Lag

As a sequel to the experiments described in the previous section, we perform an ablation study to determine the optimum lag for each method and step. For H-step forecasting, we apply the respective method with all possible lags L, $L = 1,2,\ldots,30 - H$ to see how the performance is affected by the lag used. A visualization of this method can be found in **Figure 4.2**.

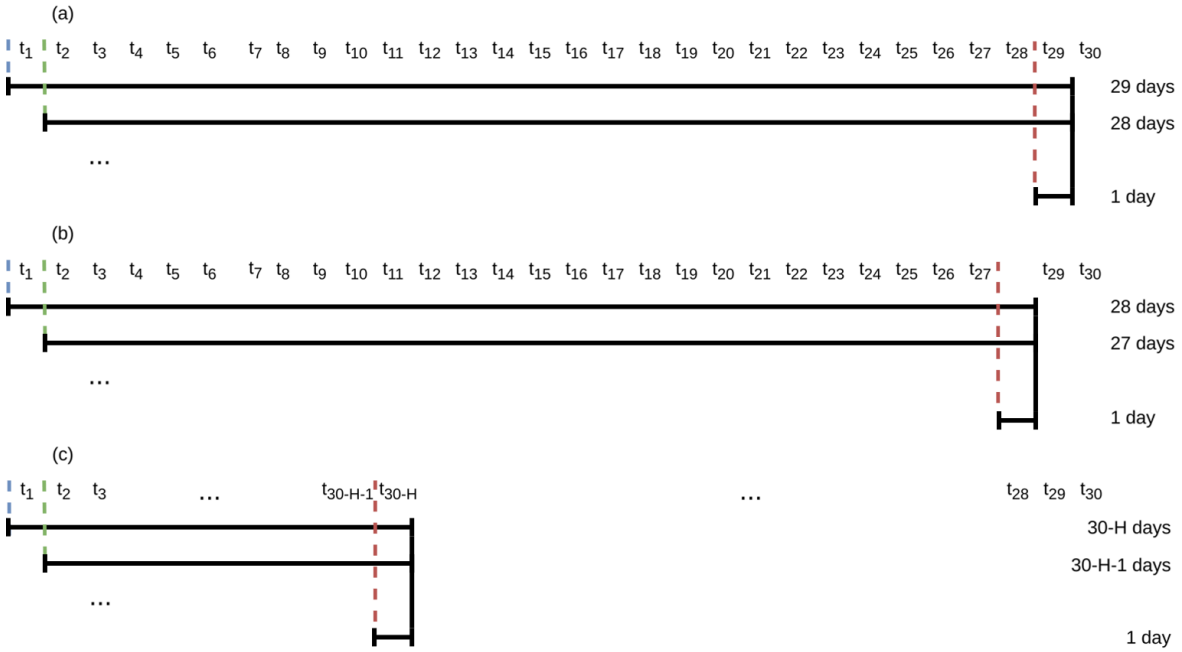


Figure 4.2: Visualization of experiments for the ablation study based on lag for (a) 1-step, (b) 2-step and (c) H-step forecasting

4.1.6 Feature Ablation in Groups

As mentioned earlier, we have already removed the satellite-derived data from the forecast LSTM of the Handoff Forecast LSTM. At this point, it is useful to experiment with ablating more features from both LSTMs to observe how the next day with 20 days lag and 1, 5, 10-step forecasting performance with 20 days lag is affected.

We begin by removing the satellite-derived data from the forecast LSTM as well, so that the model solely relies on meteorological and static data for its predictions.

Then, it is useful to gradually ablate the static features to figure out how essential they are for achieving a high prediction score. Since the number of static features is relatively large, we avoid ablating one by one at first. Instead, we sort them into groups, as discussed in **Section 3.1** and perform ablation to all combinations of groups; first, ablate each group separately, then ablate pairs of groups and lastly, ablate all groups, completely removing the static data. In each ablation, the model is trained from scratch.

The results of these experiments are indicative of the importance of each feature group. If ablating a group leads to worse performance than that of the baseline, then that group is important, otherwise it is not valuable for our task and/or model. Additionally, we can also sort different groups or combinations of groups by their importance. For example, if ablating group A leads to worse performance than when ablating group B, then A is of greater importance than B.

Moreover, we also conduct the same study to the LSTM baseline model.

4.2 xAI

4.2.1 Feature Ablation

To extend the feature ablation study described previously, we apply feature ablation to all features, using the Captum package [28] on all test samples. Particularly, the attribution of each feature in each day and sample are returned in the form of a 3D tensor of size $N \times L \times M$. Therefore, we average over the sample and time dimension sequentially, to yield a 1d vector containing the average attribution of each feature.

Regarding the experiments, first, we apply feature ablation to the LSTM baseline to obtain an initial ranking of features according to their importance. All samples are used at first and then positive and negative samples are used separately to try to get a clearer picture. Then, the same method is used to an LSTM with the same architecture that has been trained for next day forecasting using 5 and 10 days lag, to determine how the relationship between various fire drivers and wildfire danger is affected by lag.

Afterwards, we repeat the experiments using an LSTM for 1, 5 and 10-step Direct forecasting with 20 days lag, to find out if the most important features change when the forecast horizon increases.

Lastly, we apply the same method to an LSTM that has been trained for next day forecasting using only five uncorrelated variables: ndvi, rh, wind_speed, smi, tp, as an attempt to determine if the importance of these variables would change when no correlated features are used.

Two baselines are used; the zero and the average of the negative test samples baseline, exactly like in [12].

4.2.2 Partial Dependence Plots (PDPs)

We first need to adjust **Algorithm 3** to be used with time series data instead of regular multivariate samples. In this case, the iteration and substitution take place over time series X_i of the feature of interest x_Z. In particular, inside the loop, all time series of x_Z in X' are replaced by X_i. Additionally, the values on the x-axis are the mean value of x_Z in X_i.

After these modifications, we obtain a Partial Dependence Plot of the mean values of x_Z in each sample with respect to the wildfire danger. At this point, if we try connecting all (x_i, μ) points to produce a line plot, it might turn out to be very noisy, which is why smoothing is applied in most plots by grouping multiple values of x_Z into bins and plotting the average model output for each bin instead.

Apart from plotting the mean output of the model, we also provide an upper and lower bound with 95% confidence interval.

To reduce the time needed for computation, we use only 300 test samples that have been selected randomly.

The exact same experiments as the ones described in the previous section are conducted to investigate how changes in lag, forecast horizon and correlation in variables affect the relationship of different fire drivers with wildfire danger.

4.2.3 Integrated Gradients (IGs)

We apply the Integrated Gradients method implementation provided by Captum [28] on positive samples only. Similar to the feature ablation implementation, the attribution of each feature in each day and sample are returned in the form of a 3D tensor of size $N \times L \times M$. Hence, we average over the sample dimension, to yield a 2D matrix of size $L \times M$ containing the temporal evolution of each feature, according to [12].

The mean of negative samples baseline is also applied in this algorithm.

Similar to the Partial Dependence lots, we provide the mean and an upper and lower bound with 95% confidence interval.

At first, we produce the Integrated Gradients plots for the LSTM baseline and then, the exact same experiments as the ones described in the previous section are conducted to investigate how changes in lag, forecast horizon and correlation in variables affect the temporal evolution of different fire drivers.

5. RESULTS AND DISCUSSION

5.1 Forecasting

5.1.1 Iterative/Recursive Forecasting

The feature predictions of the LSTM forecaster are far off the actual values, as seen in **Figure 5.1**. As a result, the forecasting loss is high even at the very first forecasting step. This error accumulates as the number of steps increases (**Table 5.1**, **Figure 5.2**) as expected, which has a great impact on the classification results of all three models, leading to performances akin to that of a random classifier in terms of f1-score.

In comparison to the Naive forecaster (**Table 5.2**), we can observe that the LSTM forecaster is slightly more accurate in the first predictions of 1, 2 and 3-step forecasting, and much better in the rest of predictions, but still not good enough in general.

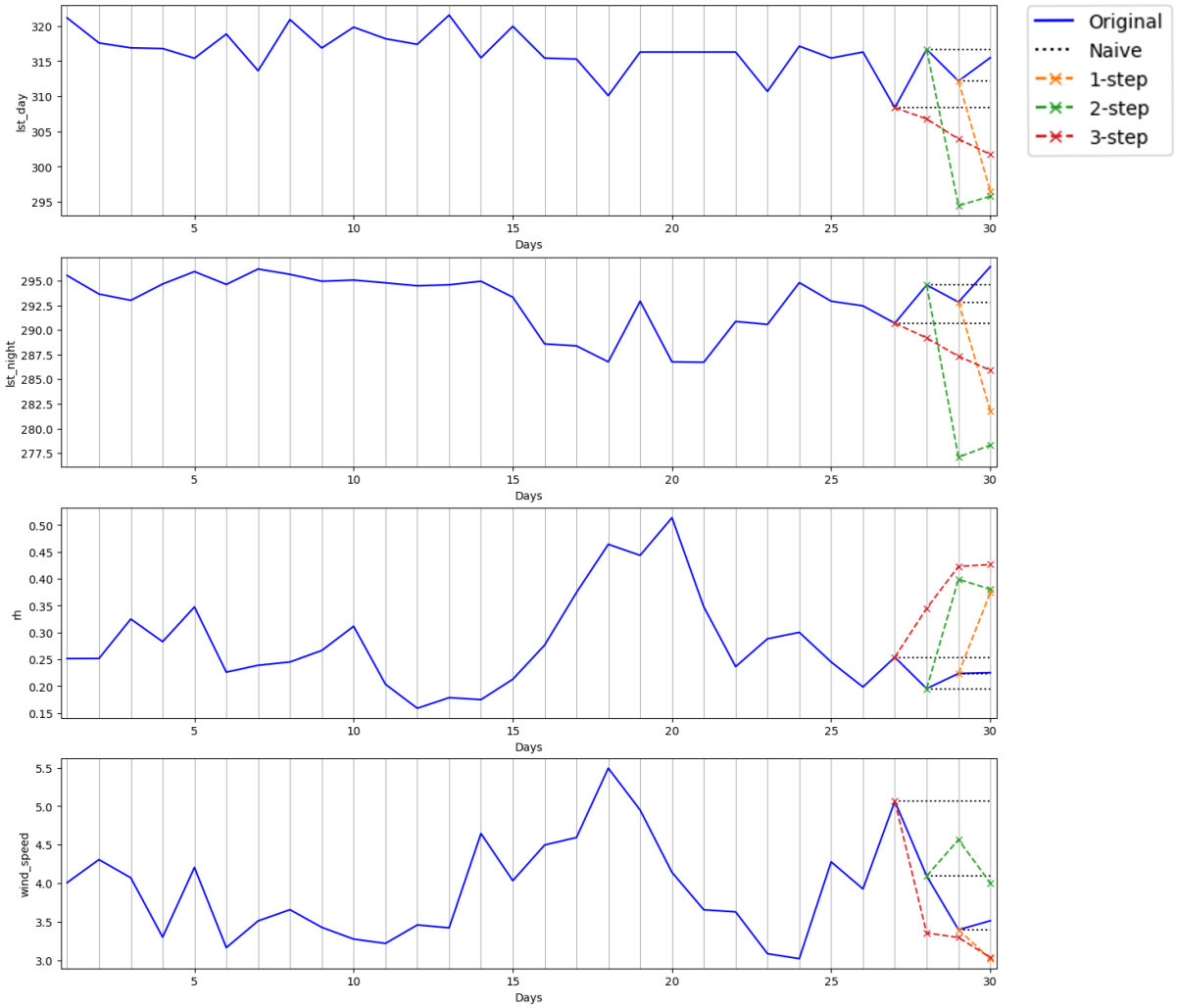
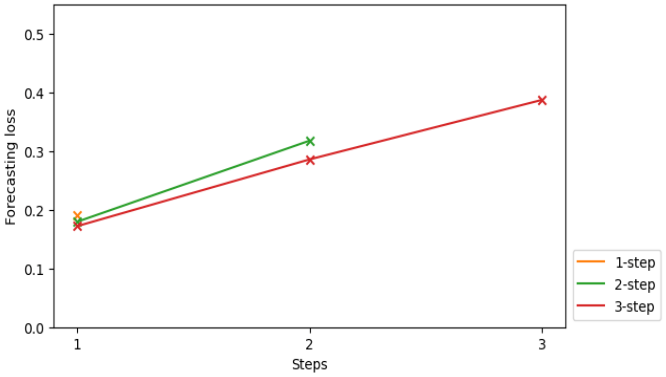


Figure 5.1: Example of predictions of the LSTM forecaster and the Naive forecaster for 1, 2 and 3-step Iterative forecasting

Table 5.1: 1, 2 and 3-step Iterative forecasting using the LSTM forecaster and each of the three baselines as classifiers

	Forecasting test loss (MSE)			Classification loss (CE), f1-score and AUPRC		
Lag\Steps	1	2	3	LSTM	GTN	Transformer
29	0.19134	-	-	2.76671 0.59730 0.65256	3.66812 0.55635 0.60699	3.38922 0.56879 0.62759
28	0.17946	0.31756	-	3.61100 0.53791 0.57165	3.99249 0.50619 0.55107	3.87985 0.52697 0.59698
27	0.17196	0.28568	0.38705	3.95919 0.51587 0.55504	4.05165 0.50312 0.54756	3.91774 0.51690 0.60430

(a)



(b)

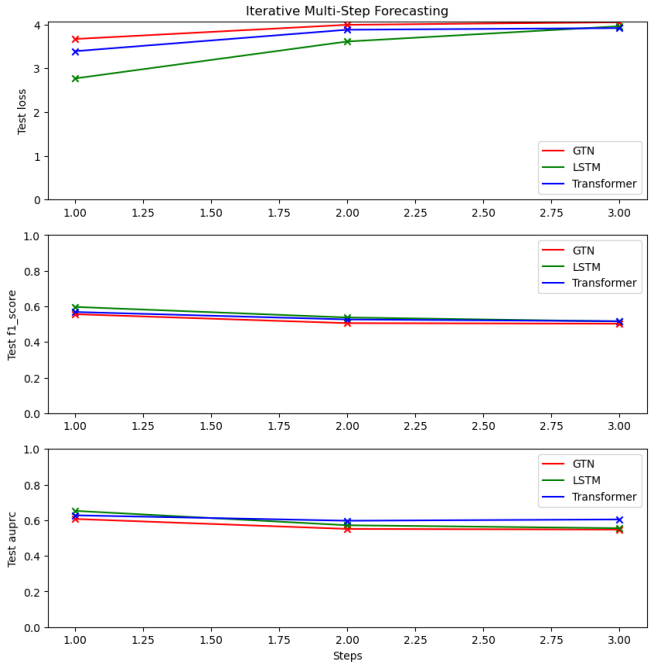


Figure 5.2: Comparison of 1, 2 and 3-step Iterative forecasting using the LSTM forecaster and each of the three baselines as classifiers with the maximum possible lag, based on (a) the forecasting loss and (b) classification metrics

Table 5.2: 1, 2 and 3-step Iterative forecasting using the Naive forecaster and the LSTM baseline as classifier

	Forecasting test loss (MSE)			Classification loss (CE), F1-score and AUPRC
Lag\Step	1	2	3	LSTM
29	0.19536	-	-	3.66812 0.55635 0.60699
28	0.19829	0.31381	-	3.99249 0.50619 0.55107
27	0.18800	0.28568	0.38705	4.05165 0.50312 0.54756

5.1.2 Direct Forecasting

As shown in **Table 5.3**, the loss increases as the forecast horizon increases, as expected, with the largest increase being recorded during the first 5 steps. Regarding the rest of classification metrics, these are not affected as much. In fact, even at 10-step forecasting, it is possible to yield acceptable results that are satisfactory considering the extended forecast horizon.

Even though all three models are pretty much indistinguishable in terms of performance (**Figure 5.3**), the most stable model for smaller forecast horizons seems to be the LSTM, while for larger horizons the GTN is probably a better choice.

Table 5.3: 1, 2, …, 10-step Direct forecasting using the LSTM, GTN and Transformer baselines with the maximum possible lag

	LSTM			GTN			Transformer		
Step	Test loss	Test f1-score	Test AUPRC	Test loss	Test f1-score	Test AUPRC	Test loss	Test f1-score	Test AUPRC
1	1.53381	0.76147	0.82293	1.51461	0.76016	0.82817	1.50012	0.76132	0.83468
2	1.61215	0.75102	0.79941	1.61747	0.73940	0.80495	1.59522	0.75075	0.81201
3	1.68500	0.73795	0.78705	1.69482	0.72578	0.79197	1.67126	0.73608	0.80467
4	1.71836	0.73349	0.78091	1.74523	0.72708	0.77944	1.75712	0.72204	0.78136
5	1.76562	0.72585	0.76910	1.80636	0.71733	0.77273	1.81141	0.70885	0.75981
6	1.81698	0.72047	0.76037	1.81284	0.70729	0.76357	1.79617	0.71237	0.77630
7	1.82679	0.71617	0.76279	1.84677	0.70309	0.75654	1.82806	0.71010	0.76216
8	1.85290	0.70916	0.75661	1.83405	0.71089	0.76434	1.88004	0.70051	0.76370
9	1.89847	0.69794	0.74854	1.85650	0.69868	0.76403	1.89635	0.69817	0.76410
10	1.86845	0.70443	0.75590	1.83823	0.70551	0.76464	1.87183	0.69025	0.75138

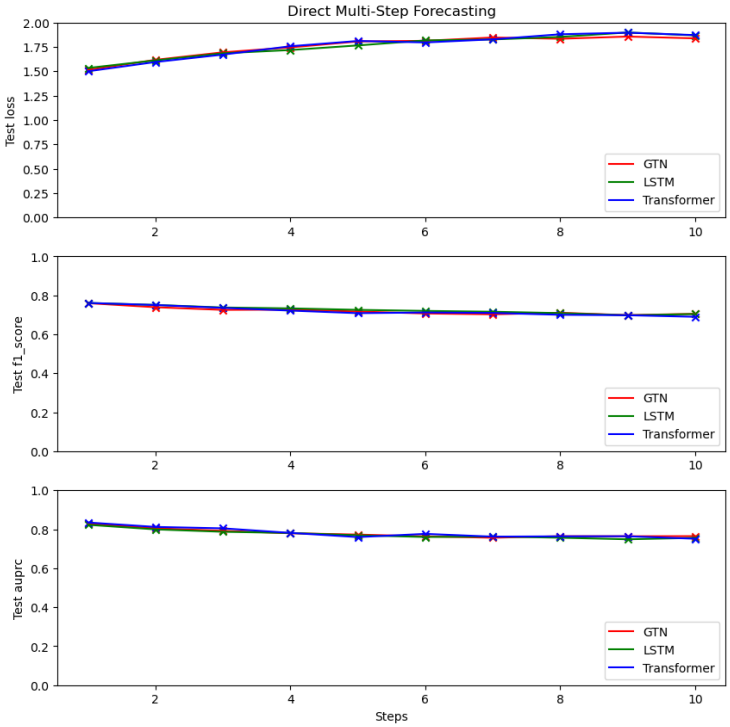


Figure 5.3: Comparison of 1, 2, …, 10-step Direct forecasting using the LSTM, GTN and Transformer baselines with the maximum possible lag based on classification metrics

5.1.3 Alternative Forecasting Approach Using Hindcasts and Forecasts

In **Table 5.4**, we can observe that the loss of the Handoff Forecast LSTM increases at a very small rate, while the rest of metrics are slightly affected by the expansion of the forecast horizon. In addition to this, the difference in performance between 1 and 10-step forecasting is almost negligible.

Compared to the Direct forecasting strategy, the Handoff Forecast LSTM outperforms all three models, as seen in **Figure 5.4**.

Table 5.4: 1, 2, …, 10-step forecasting using the Handoff Forecast LSTM

Step	Test loss	Test f1-score	Test AUPRC
1	1.36042	0.78129	0.85505
2	1.39057	0.78100	0.84976
3	1.40826	0.78043	0.84679
4	1.39917	0.77793	0.84881
5	1.39765	0.77762	0.84727
6	1.39796	0.77522	0.84902
7	1.41492	0.77900	0.84774
8	1.41194	0.77430	0.85014
9	1.42387	0.77314	0.84756
10	1.42801	0.77014	0.84897

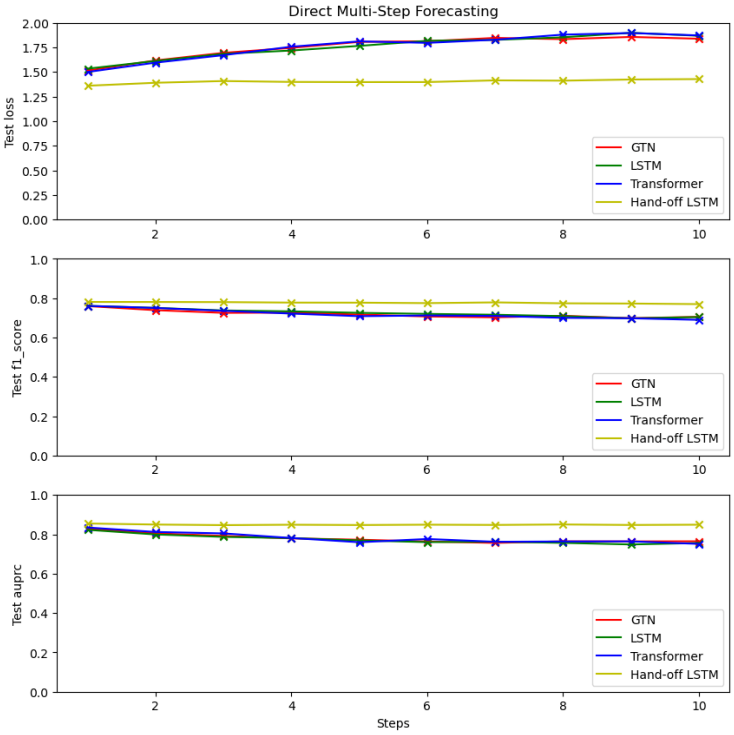


Figure 5.4: Comparison of 1, 2, …, 10-step Direct forecasting using the three baselines and 1, 2, …, 10-step forecasting using the Handoff Forecast LSTM with the maximum possible lag based on classification metrics

5.1.4 Comparison of Different Forecasting Methods

In the Iterative forecasting strategy using the LSTM forecaster and the LSTM baseline as classifier with 20 days lag, it is observed that in each error sequence (row in **Table 5.5**) the forecasting loss is increasing, as expected.

Moreover, the largest increase in the classification loss is reported between 1 and 2-step forecasting, which is also shown in **Figure 5.5**.

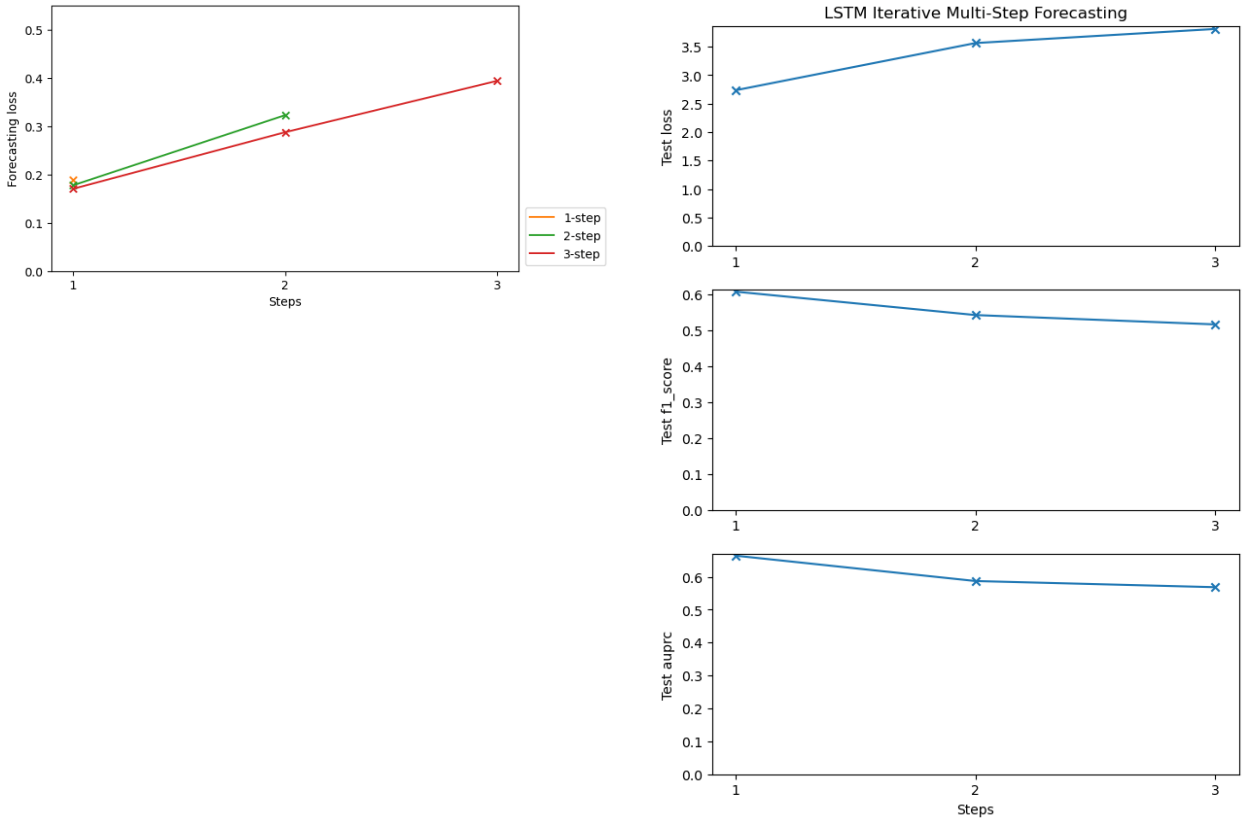
Regarding the Direct forecasting strategy, similar conclusions to those in the previous section can be made about the the impact of the forecast horizon on the metrics (**Table 5.6**, **Figure 5.6**).

Interestingly, looking at **Table 5.3** and **Table 5.6**, we can infer the results for some forecast horizons are slightly better when using 20 days lag instead of the maximum possible lag. This will be investigated in the next section.

The Handoff Forecast LSTM is still very robust to a high number of steps, even when only 20 days as used as lag, as shown in **Table 5.7**. Once again, the classification scores of the Handoff Forecast LSTM exceed those of three baselines used in the Direct forecasting strategy.

Table 5.5: 1, 2 and 3-step Iterative forecasting using the LSTM forecaster and the LSTM baseline as classifier with 20 days lag

	Forecasting test loss (MSE)			Classification loss (CE)
Step	1	2	3	LSTM
1	0.18861	-	-	2.73650 0.60740 0.66435
2	0.17748	0.32275	-	3.56521 0.54230 0.58732
3	0.17017	0.28743	0.39383	3.81283 0.51630 0.56858



(a) **(b)**

Figure 5.5: Comparison of 1, 2 and 3-step Iterative forecasting using the LSTM forecaster and each of the three baselines as classifiers with 20 days lag, based on (a) the forecasting loss and (b) classification metrics

Table 5.6: 1, 2, …, 10-step Direct forecasting using the LSTM, GTN and Transformer baselines with 20 days lag

	LSTM			GTN			Transformer		
Step	Test loss	Test f1-score	Test AUPRC	Test loss	Test f1-score	Test AUPRC	Test loss	Test f1-score	Test AUPRC
1	1.52893	0.76047	0.82468	1.50308	0.75692	0.83061	1.48815	0.76632	0.83822
2	1.60473	0.75808	0.80574	1.59744	0.74604	0.81077	1.58460	0.75016	0.81287
3	1.68430	0.73244	0.79260	1.67424	0.72465	0.79353	1.65947	0.73308	0.79157
4	1.72031	0.73385	0.78836	1.72102	0.72516	0.78565	1.70881	0.72441	0.79309
5	1.74874	0.72396	0.77101	1.77590	0.71423	0.77437	1.77262	0.71755	0.77539
6	1.80253	0.71830	0.76365	1.78922	0.71457	0.76912	1.76901	0.71909	0.77729
7	1.82675	0.71276	0.76134	1.81173	0.71162	0.76387	1.82178	0.70801	0.76900
8	1.84542	0.71109	0.75825	1.81114	0.70867	0.76576	1.86501	0.67863	0.75392
9	1.88877	0.70104	0.74793	1.84027	0.70194	0.76218	1.87817	0.68579	0.74980
10	1.86845	0.70443	0.75590	1.83823	0.70551	0.76464	1.87183	0.69025	0.75138

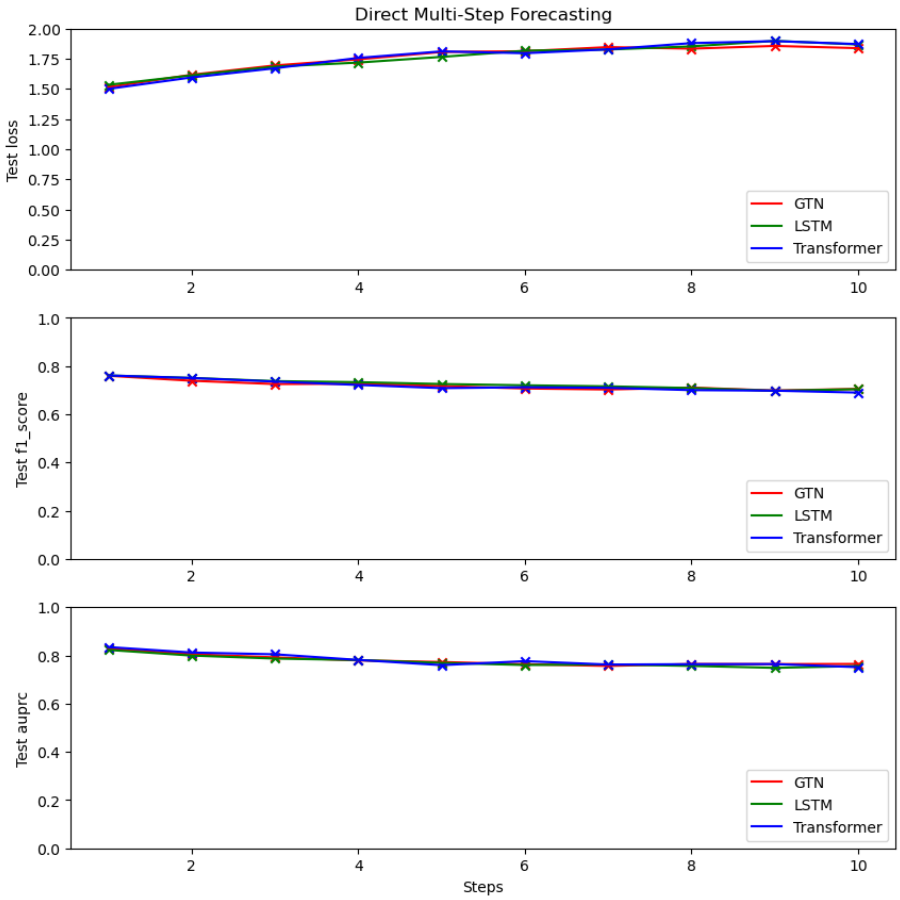


Figure 5.6: Comparison of 1, 2, …, 10-step Direct forecasting using the LSTM, GTN and Transformer baselines with 20 days lag based on classification metrics

Table 5.7: 1, 2, …, 10-step forecasting using the Handoff Forecast LSTM with 20 days lag

Steps	Test loss	Test f1-score	Test AUPRC
1	1.37525	0.78025	0.85333
2	1.40820	0.78117	0.84688
3	1.41101	0.77831	0.84594
4	1.41486	0.77636	0.84425
5	1.41887	0.77672	0.84585
6	1.41389	0.77766	0.84439
7	1.41481	0.77750	0.84738
8	1.41740	0.77445	0.84857
9	1.42286	0.77088	0.84723
10	1.42801	0.77014	0.84897

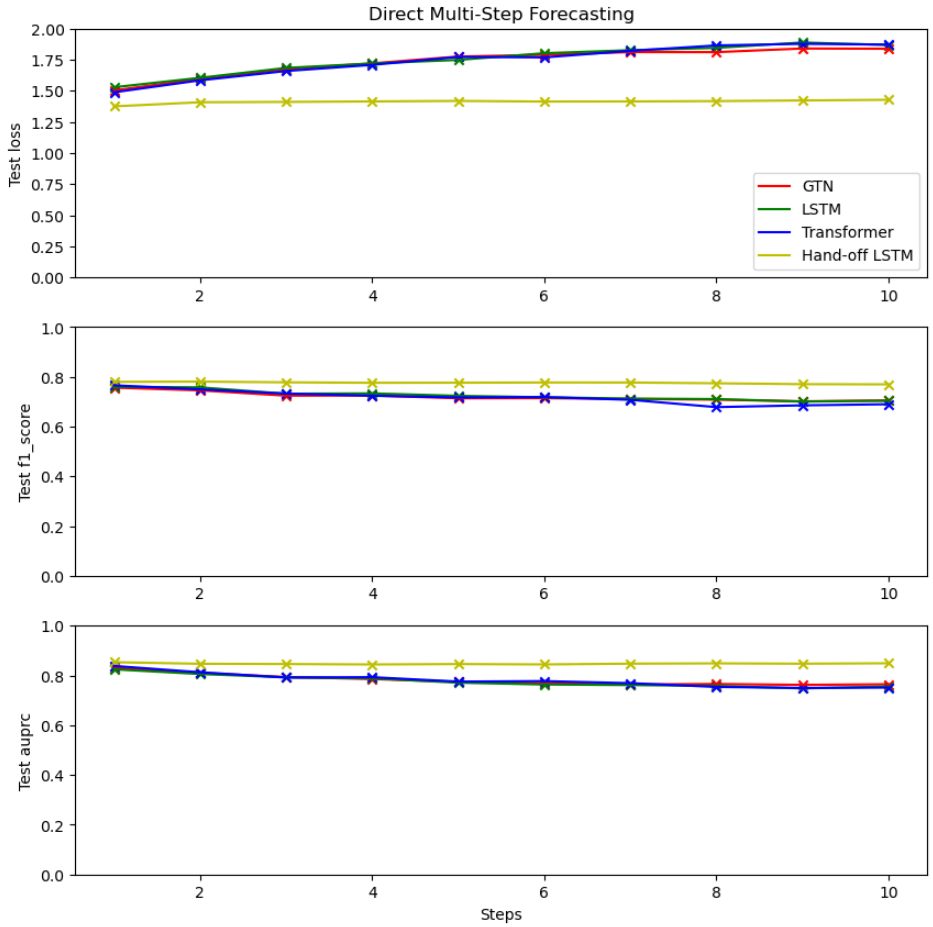


Figure 5.7: Comparison of 1, 2, …, 10-step Direct forecasting using the three baselines and 1, 2, …, 10-step forecasting using the Handoff Forecast LSTM with 20 days lag based on classification metrics

5.1.5 Ablation Study Based on Lag

In Iterative forecasting, the forecasting loss increases as the lag decreases, as seen in **Table 5.8**. Looking at the first three subplots of **Figure 5.8**, it is observed that when the lag is greater than 5 days, the forecasting loss is comparable to that reported when the maximum possible lag was used in all three steps.

In terms of classification performance, the metrics do not follow a specific increasing or decreasing trend but are quite unstable, as they are oscillating in a range of 3 to 5%.

Regarding the Direct forecasting strategy, for 1, 2, 3 and 4-step forecasting, the LSTM baseline is very robust. Even when using just a week of historical data, which is equivalent to reducing the original maximum possible lag by 73-76%, the results produced are still comparable, as seen in **Table 5.9**, while the performance gradually gets worse for lags that are smaller than a week. Furthermore, when the lag is longer or equal to 3 days, the following relation can be observed for two consecutive models using the same lag in our problem:

$$e_h^l \leq e_{h+1}^l, l = 3, \ldots, 30 - h, h = 1,2,3 \tag{5.1}$$

For 5-step forecasting, it is probably best to have more than 10 days of historical data for better results, which corresponds to a roughly 60% reduction on the original lag. However, for longer forecast horizons, both **Table 5.9** and **Figure 5.9** suggest that the results are very unstable, making it almost impossible to come to a conclusion about the optimum lag. This is the reason why we mostly focused on less than 5 steps.

Similarly, we study the 1, 2, 3 and 4-step forecasting using the Handoff Forecast LSTM. Looking at **Table 5.10** and **Figure 5.10**, it is inferred that 5 days of historical data, which translates to a 81-83% reduction on the original lag, are sufficient to secure a good result in all steps.

Again, the Handoff Forecast LSTM is superior to that of the three baselines used in Direct multi-step forecasting, as shown in **Figure 5.11**. This is probably explained by the fact that it leverages both hindcasts and forecasts instead of just forecasts.

Nevertheless, in all strategies and steps, feeding an extensive amount of historical data to a model does not necessarily guarantee an improved performance.

Table 5.8: : 1, 2, …, 10-step Iterative forecasting using the LSTM forecaster and the LSTM baseline as classifier - Ablation study based on lag

	Forecasting loss (MSE)	Classification metrics			Forecasting loss (MSE)		Classification metrics			Forecasting loss (MSE)			Classification metrics		
Lag\Step	1	Test loss	f1-score	AUPRC	1	2	Test loss	f1-score	AUPRC	1	2	3	Test loss	f1-score	AUPRC
29	0.19134	2.76671	0.59730	0.65256	-	-	-	-	-	-	-	-	-	-	-
28	0.18981	2.81531	0.60220	0.64441	0.17946	0.31756	3.61100	0.53791	0.57165	-	-	-	-	-	-
27	0.19040	2.76952	0.60699	0.65547	0.17926	0.31381	3.63303	0.53913	0.57862	0.17196	0.28568	0.38705	3.95919	0.51587	0.55504
26	0.18992	2.76579	0.60165	0.65372	0.17972	0.31808	3.59835	0.54481	0.57502	0.17169	0.28584	0.38540	4.01370	0.50977	0.54293
25	0.19024	2.81874	0.60331	0.64770	0.17883	0.31895	3.74004	0.53084	0.57058	0.17092	0.28738	0.38692	4.08304	0.50641	0.54287
24	0.19010	2.77373	0.60834	0.64974	0.17864	0.31901	3.55790	0.54462	0.57686	0.17069	0.28725	0.38856	3.92750	0.52627	0.54764
23	0.18955	2.82380	0.61048	0.65620	0.17873	0.31424	3.59545	0.54973	0.57973	0.17041	0.28534	0.38918	3.97592	0.51233	0.54875
22	0.18957	2.84861	0.59807	0.64908	0.17829	0.31942	3.78819	0.52752	0.56746	0.17050	0.28738	0.39562	4.07724	0.50710	0.54739
21	0.18896	2.74948	0.60552	0.65340	0.17760	0.31559	3.48298	0.55715	0.58240	0.17041	0.28087	0.38589	3.79297	0.52623	0.55538
20	0.18861	2.73650	0.60740	0.66435	0.17748	0.32275	3.56521	0.54230	0.58732	0.17017	0.28743	0.39383	3.81283	0.51630	0.56858
19	0.18795	2.72258	0.60965	0.66357	0.17804	0.32636	3.57554	0.54195	0.58511	0.17007	0.29133	0.39915	3.94656	0.51671	0.55848
18	0.18811	2.80998	0.60817	0.66351	0.17752	0.32618	3.60466	0.54697	0.59340	0.17033	0.29108	0.39830	3.89411	0.52075	0.56356
17	0.18902	2.92161	0.58937	0.63479	0.17772	0.32811	3.76330	0.52611	0.55880	0.17021	0.29425	0.40617	4.08370	0.50039	0.53770
16	0.18815	2.79924	0.59579	0.65460	0.17792	0.32848	3.71838	0.53266	0.57161	0.16999	0.29226	0.40420	4.12884	0.49647	0.53788
15	0.18724	2.77338	0.61115	0.65619	0.17757	0.32227	3.54262	0.55233	0.59563	0.17023	0.29645	0.41365	3.87450	0.52817	0.57149
14	0.18801	2.74140	0.61304	0.66619	0.17692	0.33139	3.52772	0.55096	0.58812	0.17058	0.29672	0.41371	3.88750	0.52570	0.55491
13	0.18779	2.75907	0.60672	0.65720	0.17740	0.32991	3.53442	0.53849	0.58720	0.17001	0.29904	0.42454	3.84450	0.51266	0.56301
12	0.18816	2.77393	0.61201	0.65513	0.17774	0.33581	3.58412	0.54265	0.58018	0.17041	0.30977	0.43850	3.75776	0.52677	0.56169
11	0.18876	2.85242	0.58884	0.64627	0.17798	0.33737	3.69436	0.52451	0.57632	0.17047	0.30452	0.43935	4.00934	0.51141	0.55436
10	0.18920	2.91705	0.58108	0.63695	0.17864	0.34494	3.80600	0.52118	0.56030	0.17106	0.31345	0.45281	4.02634	0.49865	0.53594
9	0.18956	2.80348	0.60420	0.64500	0.17911	0.34431	3.65147	0.53664	0.56560	0.17111	0.31272	0.46217	4.00731	0.50312	0.53666
8	0.19074	2.71596	0.60972	0.65642	0.17964	0.34810	3.50843	0.55706	0.57981	0.17220	0.31403	0.46975	3.81998	0.53276	0.55440
7	0.19070	2.76161	0.60195	0.65919	0.18124	0.36650	3.61208	0.55030	0.58335	0.17374	0.33121	0.50404	3.86151	0.52479	0.56204
6	0.19246	2.80751	0.59431	0.64562	0.18260	0.37738	3.60404	0.53921	0.57317	0.17540	0.33992	0.52905	3.89242	0.50703	0.54625
5	0.19563	2.77932	0.61206	0.65001	0.18494	0.38798	3.55451	0.55015	0.57975	0.17722	0.35535	0.57982	3.89333	0.52716	0.55525
4	0.19829	2.72780	0.61838	0.67367	0.18979	0.42223	3.52118	0.55639	0.59122	0.18208	0.39354	0.67529	3.82106	0.53726	0.55995
3	0.20266	2.71694	0.61077	0.66267	0.19313	0.46875	3.38632	0.55219	0.59314	0.18710	0.44191	0.84639	3.72515	0.52761	0.55767
2	0.21201	2.73574	0.62051	0.66315	0.20181	0.63297	3.47103	0.55667	0.59295	0.19505	0.60159	1.31013	3.78828	0.52386	0.55468
1	0.23381	2.82926	0.60521	0.65556	0.22459	1.38427	3.33335	0.56324	0.59266	0.21720	1.35293	1.35053	3.59933	0.52554	0.56309

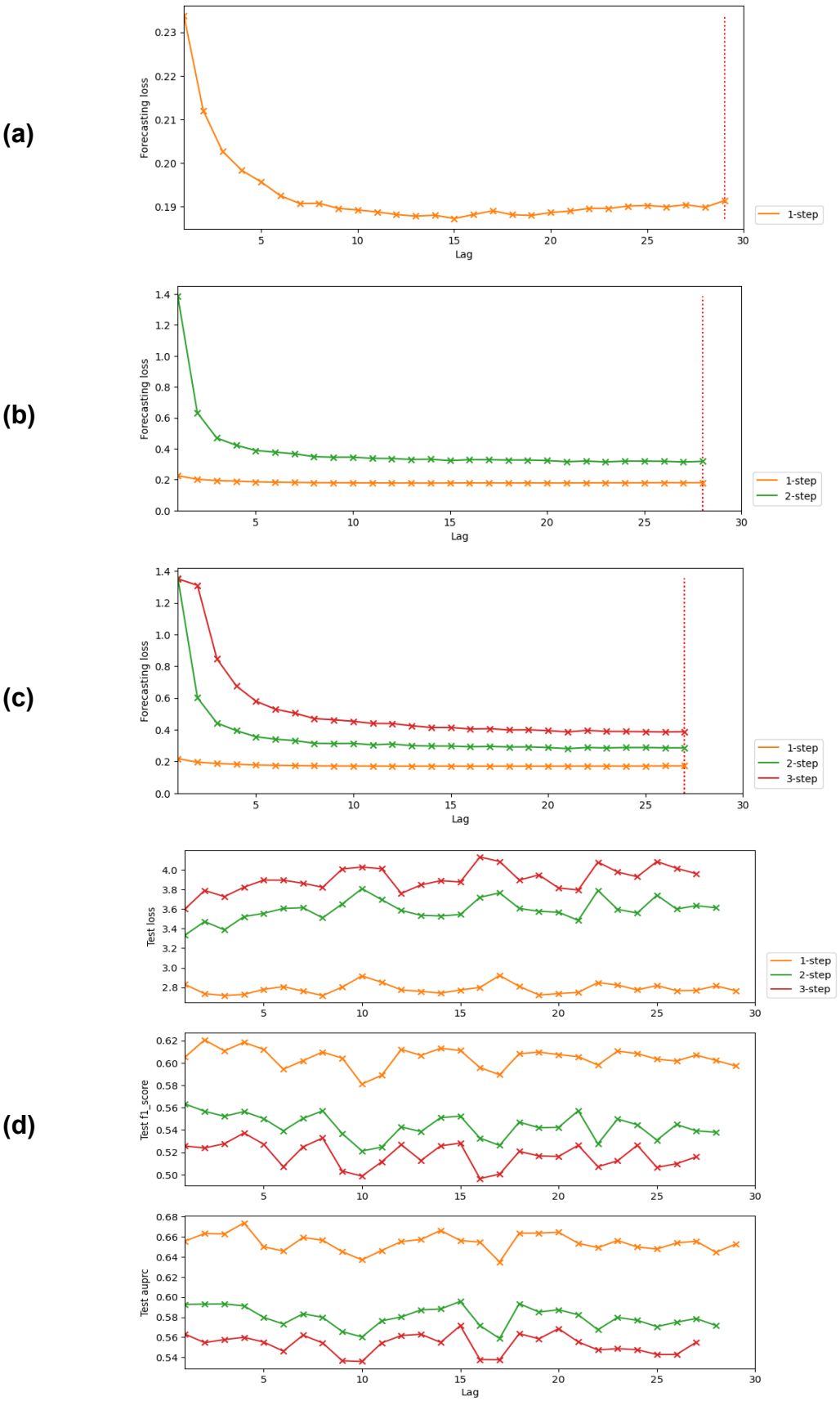


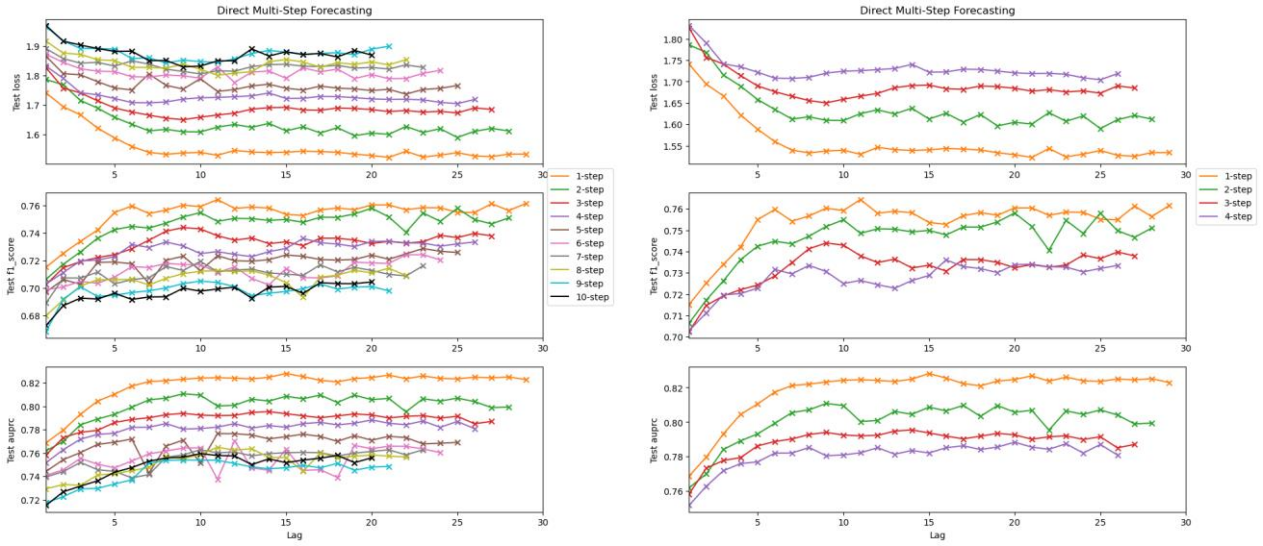
Figure 5.8: Comparison of 1, 2 and 2-step Iterative forecasting using the LSTM forecaster and each of the three baselines as classifiers with different lags, based on (a, b, c) the forecasting loss and (d) classification metrics

Table 5.9: 1, 2, …, 10-step Direct forecasting using the LSTM baseline – Ablation study based on lag

	1-step			2-step			3-step			4-step			5-step		
Lag	Test loss	f1-score	AUPRC	Test loss	f1-score	AUPRC	Test loss	f1-score	AUPRC	Test loss	f1-score	AUPRC	Test loss	f1-score	AUPRC
29	1.53381	0.76147	0.82293	-	-	-	-	-	-	-	-	-	-	-	-
28	1.53378	0.75640	0.82517	1.61215	0.75102	0.79941	-	-	-	-	-	-	-	-	-
27	1.52523	0.76134	0.82446	1.62054	0.74669	0.79894	1.68500	0.73795	0.78705	-	-	-	-	-	-
26	1.52758	0.75491	0.82497	1.61072	0.74982	0.80412	1.68973	0.73980	0.78509	1.71836	0.73349	0.78091	-	-	-
25	1.53858	0.75515	0.82356	1.59011	0.75803	0.80703	1.67291	0.73666	0.79149	1.70434	0.73209	0.78700	1.76562	0.72585	0.76910
24	1.53029	0.75838	0.82394	1.61957	0.74854	0.80450	1.67826	0.73837	0.78992	1.70871	0.73048	0.78205	1.75612	0.72671	0.76836
23	1.52377	0.75850	0.82605	1.60715	0.75463	0.80649	1.67572	0.73365	0.79204	1.71707	0.73272	0.78745	1.75232	0.72875	0.76783
22	1.54332	0.75698	0.82372	1.62707	0.74063	0.79536	1.68107	0.73270	0.79146	1.71923	0.73286	0.78424	1.73651	0.72488	0.77323
21	1.52259	0.76046	0.82678	1.60064	0.75177	0.80682	1.67759	0.73389	0.79009	1.71875	0.73407	0.78548	1.75283	0.72089	0.77430
20	1.52893	0.76047	0.82468	1.60473	0.75808	0.80574	1.68430	0.73244	0.79260	1.72031	0.73385	0.78836	1.74874	0.72396	0.77101
19	1.53353	0.75703	0.82373	1.59628	0.75391	0.80956	1.68809	0.73494	0.79346	1.72464	0.73016	0.78552	1.75429	0.72076	0.77483
18	1.53995	0.75826	0.82102	1.62304	0.75144	0.80332	1.68966	0.73623	0.79184	1.72831	0.73202	0.78412	1.75685	0.72032	0.77001
17	1.54207	0.75683	0.82230	1.60526	0.75152	0.80965	1.68190	0.73624	0.79027	1.72890	0.73297	0.78626	1.76365	0.72073	0.77419
16	1.54354	0.75275	0.82549	1.62584	0.74780	0.80652	1.68334	0.73082	0.79190	1.72251	0.73609	0.78515	1.75051	0.72308	0.77618
15	1.54024	0.75364	0.82813	1.61238	0.74991	0.80864	1.69186	0.73359	0.79369	1.72205	0.72881	0.78209	1.75647	0.72407	0.77399
14	1.53883	0.75821	0.82491	1.63760	0.74924	0.80450	1.69071	0.73233	0.79548	1.74022	0.72641	0.78355	1.77000	0.72059	0.77213
13	1.54077	0.75892	0.82350	1.62418	0.75050	0.80616	1.68570	0.73645	0.79476	1.73061	0.72282	0.78140	1.76427	0.71954	0.77547
12	1.54614	0.75797	0.82415	1.63390	0.75067	0.80093	1.67236	0.73485	0.79233	1.72787	0.72448	0.78516	1.75203	0.71991	0.77657
11	1.52987	0.76438	0.82455	1.62496	0.74857	0.80033	1.66595	0.73801	0.79205	1.72549	0.72646	0.78219	1.74577	0.72347	0.77661
10	1.53930	0.75925	0.82423	1.60882	0.75488	0.80958	1.65862	0.74302	0.79249	1.72421	0.72505	0.78098	1.78865	0.71261	0.75135
9	1.53759	0.76031	0.82325	1.60945	0.75175	0.81079	1.65007	0.74402	0.79404	1.71977	0.73063	0.78048	1.75356	0.72317	0.77109
8	1.53316	0.75671	0.82200	1.61746	0.74723	0.80708	1.65544	0.74133	0.79282	1.70996	0.73351	0.78527	1.76637	0.72024	0.76597
7	1.53961	0.75424	0.82118	1.61229	0.74365	0.80550	1.66551	0.73484	0.79021	1.70719	0.72955	0.78208	1.80448	0.70323	0.74274
6	1.55953	0.75971	0.81739	1.63469	0.74483	0.79927	1.67633	0.72855	0.78874	1.70792	0.73162	0.78201	1.75068	0.71772	0.77199
5	1.58830	0.75498	0.81049	1.65794	0.74253	0.79318	1.68974	0.72430	0.78623	1.72187	0.72285	0.77694	1.75728	0.71893	0.76929
4	1.62197	0.74221	0.80446	1.68936	0.73623	0.78913	1.71401	0.72214	0.77943	1.73436	0.72032	0.77596	1.77873	0.71864	0.76735
3	1.66637	0.73401	0.79319	1.71515	0.72629	0.78416	1.74091	0.71925	0.77776	1.74130	0.71970	0.77183	1.80171	0.70279	0.76039
2	1.69390	0.72525	0.77969	1.76825	0.71725	0.76996	1.75665	0.71476	0.77328	1.78998	0.71128	0.76285	1.80594	0.70611	0.75434
1	1.74056	0.71518	0.76876	1.78576	0.70647	0.76191	1.82549	0.70259	0.75836	1.83087	0.70291	0.75178	1.86417	0.69725	0.74425

	6-step			7-step			8-step			9-step			10-step		
Lag	Test loss	f1-score	AUPRC	Test loss	f1-score	AUPRC	Test loss	f1-score	AUPRC	Test loss	f1-score	AUPRC	Test loss	f1-score	AUPRC
29	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
28	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
27	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
26	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
25	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
24	1.81698	0.72047	0.76037	-	-	-	-	-	-	-	-	-	-	-	-
23	1.80687	0.72423	0.76351	1.82679	0.71617	0.76279	-	-	-	-	-	-	-	-	-
22	1.78952	0.72426	0.76570	1.83544	0.70879	0.75840	1.85290	0.70916	0.75661	-	-	-	-	-	-
21	1.78869	0.71785	0.76589	1.82128	0.70996	0.76276	1.83556	0.71438	0.75740	1.89847	0.69794	0.74854	-	-	-
20	1.80253	0.71830	0.76365	1.82675	0.71276	0.76134	1.84542	0.71109	0.75825	1.88877	0.70104	0.74793	1.86845	0.70443	0.75590
19	1.78944	0.71878	0.76652	1.82447	0.71506	0.75994	1.83733	0.71281	0.75722	1.87006	0.70061	0.74530	1.88410	0.70315	0.75175
18	1.82127	0.70882	0.73907	1.83370	0.71176	0.75748	1.84257	0.70934	0.75615	1.87653	0.69927	0.75111	1.86215	0.70322	0.75798
17	1.81171	0.70739	0.74561	1.82980	0.71691	0.76024	1.82773	0.70791	0.76001	1.87441	0.70290	0.74732	1.87409	0.70381	0.75554
16	1.82555	0.70759	0.74518	1.83106	0.70925	0.76057	1.84514	0.69327	0.74431	1.87017	0.69948	0.74950	1.87056	0.69610	0.75394
15	1.79009	0.71418	0.76327	1.83745	0.71005	0.76027	1.85375	0.70392	0.75609	1.87891	0.69748	0.74736	1.87958	0.70119	0.75172
14	1.81468	0.70224	0.74531	1.83619	0.71101	0.75948	1.84615	0.70954	0.75619	1.88400	0.69619	0.74655	1.86539	0.70060	0.75499
13	1.81119	0.70697	0.74718	1.82966	0.71375	0.75757	1.81242	0.71250	0.76366	1.87402	0.69449	0.74810	1.88940	0.69262	0.75008
12	1.77604	0.71531	0.77053	1.81395	0.71300	0.76086	1.80741	0.71203	0.76245	1.85551	0.70109	0.75099	1.84961	0.70067	0.75751
11	1.82835	0.71135	0.73760	1.81546	0.71189	0.76019	1.80017	0.71267	0.76498	1.84440	0.70398	0.75381	1.84870	0.69931	0.75787
10	1.79060	0.71692	0.76451	1.80496	0.71945	0.76279	1.82075	0.71219	0.75879	1.84628	0.70503	0.75387	1.83147	0.69770	0.75948

9	1.79836	0.71710	0.76428	1.81308	0.71275	0.75836	1.83812	0.71045	0.75516	1.85158	0.70318	0.75407	1.82851	0.69989	0.75646
8	1.80130	0.71791	0.76169	1.82084	0.71563	0.75641	1.82430	0.70708	0.75690	1.84204	0.70006	0.75372	1.85032	0.69366	0.75576
7	1.79410	0.71495	0.75927	1.83796	0.70750	0.74154	1.82664	0.70217	0.74772	1.85926	0.69797	0.75143	1.84977	0.69348	0.75297
6	1.79552	0.71558	0.75316	1.84912	0.70612	0.73873	1.82719	0.70629	0.74489	1.85689	0.69669	0.73704	1.88173	0.69168	0.74750
5	1.81266	0.70795	0.74734	1.83121	0.70302	0.74428	1.84983	0.70607	0.74235	1.88890	0.69479	0.73346	1.88104	0.69615	0.74354
4	1.81413	0.70348	0.75041	1.84386	0.71152	0.74565	1.85312	0.70600	0.74106	1.89066	0.69387	0.72969	1.89058	0.69199	0.73616
3	1.82226	0.70483	0.75538	1.84117	0.70726	0.75200	1.87084	0.70143	0.73238	1.89102	0.70085	0.72930	1.90249	0.69264	0.73161
2	1.84380	0.70082	0.74561	1.85666	0.70733	0.74404	1.87520	0.69126	0.73298	1.91660	0.69198	0.72262	1.91570	0.68718	0.72679
1	1.87120	0.69871	0.74087	1.89018	0.68910	0.73948	1.91537	0.68005	0.72933	1.96377	0.66845	0.71737	1.96816	0.67273	0.71518



(a) **(b)**

Figure 5.9: Comparison of (a) 1, 2, …, 10-step and (b) 1, 2, 3 and 4-step Iterative forecasting using the LSTM baseline with different lags, based on classification metrics

Table 5.10: 1, 2, 3, 4-step forecasting using the Handoff Forecast LSTM – Ablation study based on lag

	1-step			2-step			3-step			4-step		
Lag	Test loss	f1-score	AUPRC	Test loss	f1-score	AUPRC	Test loss	f1-score	AUPRC	Test loss	f1-score	AUPRC
29	1.36042	0.78129	0.85505	-	-	-	-	-	-	-	-	-
28	1.37026	0.78203	0.85523	1.39057	0.78100	0.84976	-	-	-	-	-	-
27	1.35951	0.78018	0.85689	1.37654	0.78218	0.85204	1.40826	0.78043	0.84679	-	-	-
26	1.36120	0.78134	0.85905	1.38621	0.77808	0.85105	1.40275	0.78190	0.84773	1.39917	0.77793	0.84881
25	1.36805	0.78447	0.85522	1.39284	0.78160	0.84989	1.39468	0.78224	0.84967	1.41211	0.77863	0.84870
24	1.37508	0.78183	0.85656	1.39732	0.78185	0.84945	1.40808	0.78016	0.84758	1.42244	0.77565	0.84655
23	1.38652	0.77977	0.85297	1.39516	0.78148	0.85002	1.40153	0.78106	0.84742	1.41464	0.77291	0.84734
22	1.38016	0.78227	0.85384	1.40345	0.77392	0.84706	1.40590	0.77874	0.84735	1.42464	0.77275	0.84547
21	1.38166	0.77970	0.85331	1.40614	0.77980	0.84804	1.40392	0.77973	0.84655	1.42432	0.77290	0.84489
20	1.37525	0.78025	0.85333	1.40820	0.78117	0.84688	1.41101	0.77831	0.84594	1.41486	0.77636	0.84425
19	1.38685	0.77954	0.85181	1.41124	0.77995	0.84480	1.40660	0.78119	0.84594	1.42720	0.77327	0.84281
18	1.37345	0.78493	0.85434	1.41421	0.78128	0.84659	1.40416	0.78168	0.84530	1.42339	0.77430	0.84228
17	1.37242	0.78097	0.85526	1.41032	0.77835	0.84654	1.41535	0.77942	0.84409	1.42340	0.77426	0.84260
16	1.38013	0.78275	0.85381	1.41297	0.77914	0.84635	1.41589	0.77617	0.84446	1.41890	0.77465	0.84402
15	1.38496	0.77977	0.85366	1.42080	0.77720	0.84542	1.41879	0.78108	0.84445	1.41897	0.78048	0.84569
14	1.38462	0.78121	0.85258	1.43065	0.77669	0.84369	1.42824	0.78000	0.84137	1.40638	0.78284	0.84747
13	1.37825	0.77937	0.85331	1.41581	0.78004	0.84567	1.43089	0.78323	0.84118	1.43163	0.77686	0.84326
12	1.38806	0.77925	0.85234	1.41239	0.77732	0.84627	1.43438	0.77675	0.83884	1.43410	0.78224	0.84161
11	1.39580	0.77819	0.85094	1.41928	0.77962	0.84552	1.43602	0.77766	0.83935	1.43358	0.77728	0.84104
10	1.39798	0.78005	0.85078	1.43095	0.77666	0.84422	1.44404	0.77808	0.83808	1.44317	0.77952	0.83855
9	1.42177	0.77471	0.84723	1.43522	0.77128	0.84342	1.44354	0.77736	0.83928	1.44147	0.77754	0.84163
8	1.42922	0.77043	0.84658	1.45061	0.77027	0.84037	1.45067	0.77647	0.83959	1.46207	0.77247	0.83928
7	1.44019	0.76943	0.84410	1.45196	0.77507	0.84138	1.46481	0.77303	0.83764	1.46801	0.76991	0.83850
6	1.44455	0.76611	0.84223	1.47080	0.76854	0.83661	1.46823	0.77096	0.83816	1.48101	0.76248	0.83610
5	1.46541	0.75611	0.83976	1.45949	0.77289	0.83789	1.47383	0.76865	0.83703	1.48521	0.76190	0.83600
4	1.51164	0.75551	0.82971	1.48673	0.76386	0.83400	1.48993	0.76964	0.83252	1.49508	0.76295	0.83506
3	1.52854	0.75372	0.82321	1.53089	0.75433	0.82519	1.50992	0.75697	0.82944	1.49726	0.76209	0.83355
2	1.55973	0.74814	0.81523	1.56182	0.74670	0.81812	1.54114	0.75098	0.82510	1.51259	0.75810	0.83095
1	1.61310	0.74095	0.80175	1.59132	0.73157	0.81090	1.58296	0.73961	0.815960	1.56061	0.74096	0.82272

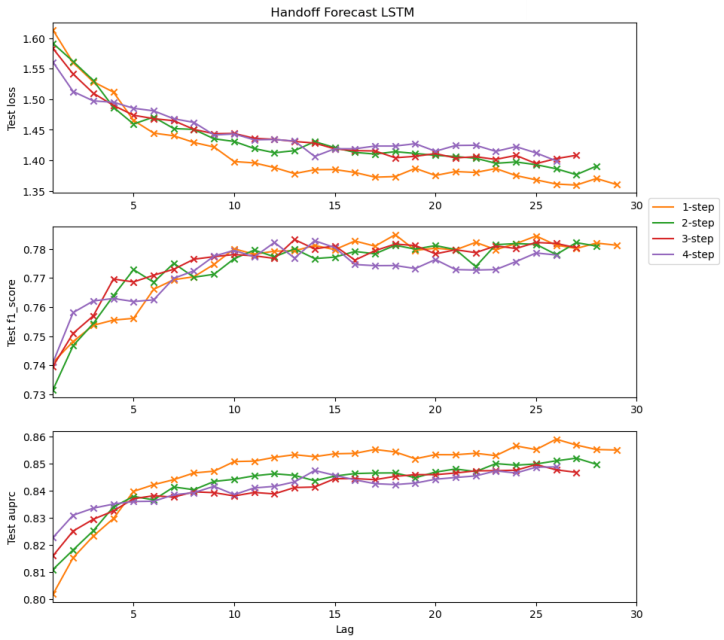


Figure 5.10: Comparison of 1, 2, 3 and 4-step forecasting using the Handoff Forecast LSTM with different lags, based on classification metrics

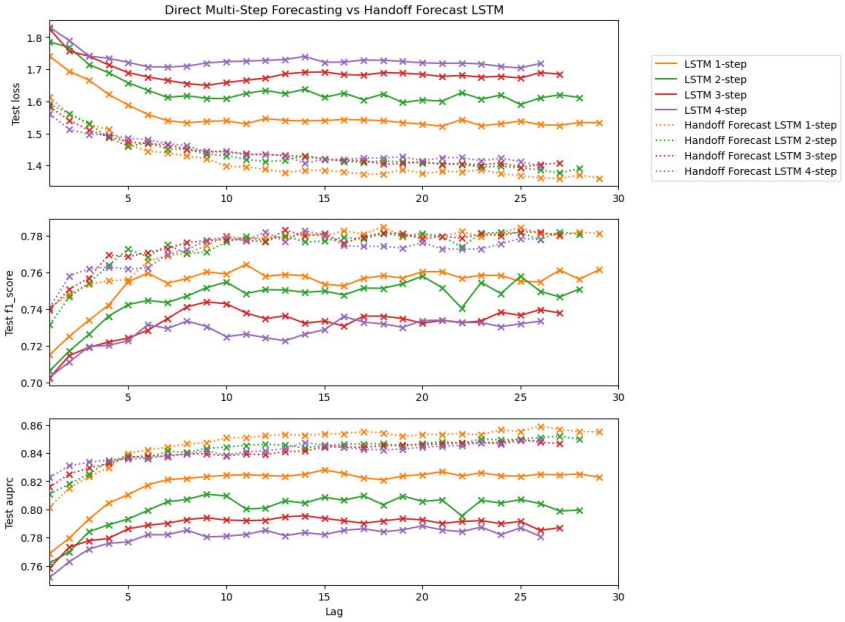


Figure 5.11: Comparison of 1, 2, 3 and 4-step Direct forecasting using the LSTM baseline and 1, 2, 3 and 4-step forecasting using the Handoff Forecast LSTM with different lags based on classification metrics

5.1.6 Feature Ablation in Groups

5.1.6.1 LSTM

Removing the satellite-derived data leads to worse scores in next day forecasting using the LSTM baseline, as seen in **Table 5.11**, meaning that these features are important for the model's predictions.

In 1, 5 and 10-step Direct forecasting with 20 days lag, the difference in loss of both cases is increasing as the forecast horizon expands, indicating that the importance of the satellite-derived data is also increasing, as shown in **Table 5.12**.

Regarding the ablation of static data in groups in the same forecasting strategy and lag, looking at **Table 5.12** and **Table 5.13**, it is once again inferred that ablating any of the three groups worsens the model's performance and that the importance of each group of features is greater for larger forecast horizons. Using **Table 5.13**, we can extract the following relation for the classification loss of 1 and 5-step Direct forecasting:

$$loss^{h}_{humanindicators} < loss^{h}_{topography} < loss^{h}_{landcover}, h = 1{,}5 \tag{5.2}$$

The same ranking applies to the importance of these feature groups. Similarly, for 10-step Direct forecasting, the following relation can be extracted:

$$loss^{h}_{topography} < loss^{h}_{humanindicators} < loss^{h}_{landcover}, h = 10 \tag{5.3}$$

which suggests that the land cover features are the most important for all three forecast horizons.

As shown in **Table 5.12** and **Table 5.14**, ablating the static data in pairs of groups leads to worse performance. In addition to this, the **Table 5.14** leads to the following ranking for the classification loss of 1-step Direct forecasting:

$$loss^{h}_{\{humanindicators,topography\}} < loss^{h}_{\{topography,landcover\}} < loss^{h}_{\{humanindicators,landcover\}}, h = 1 \tag{5.4}$$

The same ranking applies to the importance of these feature groups. Similarly, for 5 and 10-step Direct forecasting, the following relation can be extracted:

$$loss^{h}_{\{humanindicators,topography\}} < loss^{h}_{\{humanindicators,landcover\}} < loss^{h}_{\{topography,landcover\}}, h = 5{,}10 \tag{5.5}$$

which suggests that the land cover features are the most important for all three forecast horizons.

Removing the static data completely worsens the classification performance only in 5 and 10-step forecasting, compared to ablating the features in pairs of groups, as seen in **Table 5.15**.

Table 5.11: Next day forecasting using the LSTM baseline – ablating satellite-derived data

Satellite-derived	Test loss	f1-score	AUPRC
Y	1.40834	0.78696	0.84756
N	1.50464	0.75300	0.82897

Table 5.12: 1, 5 and 10-step Direct Forecasting using the LSTM with 20 days lag – ablating satellite-derived data

	Satellite-derived			Without satellite-derived		
Step	Test loss	f1-score	AUPRC	Test loss	f1-score	AUPRC
1	1.52893	0.76047	0.82468	1.67843	0.72504	0.78253
5	1.74874	0.72396	0.77101	1.91904	0.68309	0.71973
10	1.86845	0.70443	0.75590	2.02655	0.65737	0.71172

Table 5.13: 1, 5 and 10-step Direct Forecasting using the LSTM with 20 days lag – ablating static data in groups

	Without group 1 (human indicators)			Without group 2 (topography)			Without group 3 (land cover)		
Step	Test loss	f1-score	AUPRC	Test loss	f1-score	AUPRC	Test loss	f1-score	AUPRC
1	1.53843	0.75686	0.82089	1.56048	0.74665	0.81504	1.66578	0.73460	0.79317
5	1.78594	0.70994	0.76787	1.80499	0.70535	0.75930	1.92014	0.68219	0.72114
10	1.90344	0.69025	0.74819	1.89352	0.68329	0.74867	2.00777	0.65768	0.70603

Table 5.14: 1, 5 and 10-step Direct Forecasting using the LSTM with 20 days lag – ablating static data in pairs of groups

	Without groups 1 & 2 (human indicators & topography)			Without groups 1 & 3 (human indicators & land cover)			Without groups 2 & 3 (topography & land cover)		
Step	Test loss	f1-score	AUPRC	Test loss	f1-score	AUPRC	Test loss	f1-score	AUPRC
1	1.56789	0.73702	0.81413	1.68097	0.73158	0.78553	1.74020	0.71283	0.76821
5	1.83080	0.69037	0.75864	1.94117	0.67203	0.71122	1.91971	0.67458	0.72645
10	1.96041	0.67233	0.73084	2.01140	0.65502	0.70304	1.98793	0.65260	0.70237

Table 5.15: 1, 5 and 10-step Direct Forecasting using the LSTM with 20 days lag – ablating all static data

	Without static data		
Step	Test loss	f1-score	AUPRC
1	1.69397	0.72135	0.77953
5	1.97383	0.65942	0.70836
10	2.04675	0.65214	0.69594

5.1.6.2 Handoff Forecast LSTM

Removing the satellite-derived data from the hindcasts as well results in worse scores in forecasting using the Handoff Forecast LSTM, as seen in **Table 5.16**. This implies that satellite-derived features carry a lot of useful information even when they are completely absent from the forecasts.

In 1, 5 and 10-step forecasting using the Handoff Forecast LSTM with 20 days lag, the loss difference between using and removing the satellite-derived data increases as the number of steps increase (**Table 5.17**). This points out the great importance of these features for larger forecast horizons.

When ablating the static data in groups, it is observed that the removal of any of the three groups limits the model's prediction ability (**Table 5.16**, **Table 5.18**). Moreover, the importance of each group increases as the forecast horizon expands. Looking at **Table 5.18**, we can extract the following relations for the classification loss of all three forecasting steps:

$$loss^{h}_{humanindicators} < loss^{h}_{topography} < loss^{h}_{landcover}, h = 1,5,10 \quad (5.6)$$

The same ranking applies to the importance of these feature groups. At this point we should note that the land cover feature group retains its importance even when the static feature encoding is dropped.

According to **Table 5.16** and **Table 5.19**, ablating the static data in pairs of groups also increases the classification loss. Similarly, we can conclude the following relations for 1 and 5-step forecasting loss:

$$loss^{h}_{\{humanindicators,topography\}} < loss^{h}_{\{humanindicators,landcover\}} < loss^{h}_{\{topography,landcover\}}, h = 1,5 \quad (5.7)$$

and the following relation for 10-step forecasting loss:

$$loss^{h}_{\{humanindicators,topography\}} < loss^{h}_{\{topography,landcover\}} < loss^{h}_{\{humanindicators,landcover\}}, h = 10 \quad (5.8)$$

Also, without static feature encoding, (5.7) is observed for all three forecasting steps.

The relations (5.6), (5.7) and (5.8) suggest that the land cover feature group is the most important for this model.

Removing the static data completely worsens the classification performance only in 5 and 10-step forecasting, compared to ablating the features in pairs of groups, as seen in **Table 5.20**.

Table 5.16: Next day forecasting using the Handoff Forecast LSTM – ablating satellite-derived data

Satellite-derived	Test loss	f1-score	AUPRC
Y	1.36042	0.78129	0.85505
N	1.56151	0.74402	0.81676

Table 5.17: 1, 5 and 10-step forecasting using the Handoff Forecast LSTM 20 days lag – ablating satellite-derived data

	Satellite-derived			Without satellite-derived		
Step	Test loss	f1-score	AUPRC	Test loss	f1-score	AUPRC
1	1.37525	0.78025	0.85333	1.58925	0.73881	0.81286
5	1.41887	0.77672	0.84585	1.71784	0.71535	0.78797
10	1.42801	0.77014	0.84897	1.73999	0.71035	0.78483

Table 5.18: 1, 5 and 10-step Direct Forecasting using the Handoff Forecast LSTM with 20 days lag – ablating static data in groups

	Without group 1 (human indicators)			Without group 2 (topography)			Without group 3 (land cover)			Without encoding		
Step	Test loss	f1-score	AUPRC	Test loss	f1-score	AUPRC	Test loss	f1-score	AUPRC			
1	1.39453	0.77526	0.85143	1.42558	0.77437	0.84249	1.45809	0.76426	0.83244	1.42704	0.76995	0.83926
5	1.41387	0.77750	0.84584	1.42645	0.77477	0.84006	1.48964	0.75716	0.82935	1.46444	0.76187	0.83182
10	1.43569	0.76838	0.84665	1.44720	0.76604	0.83693	1.51551	0.75488	0.82332	1.46225	0.76513	0.83250

Table 5.19: 1, 5 and 10-step Direct Forecasting using the Handoff Forecast LSTM with 20 days lag – ablating static data in pairs of groups

	Without groups 1 & 2 (human indicators & topography)			Without groups 1 & 3 (human indicators & land cover)			Without groups 2 & 3 (topography & land cover)			Without encoding			Without encoding		
Step	Test loss	f1-score	AUPRC	Test loss	f1-score	AUPRC	Test loss	f1-score	AUPRC						
1	1.41766	0.77272	0.84636	1.45206	0.76127	0.82803	1.49148	0.76423	0.81983	1.48600	0.83905	0.82738	1.48688	0.76070	0.82591
5	1.43824	0.76705	0.83664	1.49887	0.75341	0.82718	1.53634	0.75326	0.81680	1.47936	0.75497	0.82856	1.48624	0.75854	0.82609
10	1.45322	0.76975	0.83875	1.53361	0.75296	0.82054	1.53299	0.75578	0.81638	1.47651	0.75508	0.83152	1.52760	0.75598	0.81847

Table 5.20: 1, 5 and 10-step Direct Forecasting using the Handoff Forecast LSTM with 20 days lag – ablating all static data

	Without static data		
Step	Test loss	f1-score	AUPRC
1	1.52944	0.75284	0.81413
5	1.52834	0.75281	0.81576
10	1.55826	0.74731	0.81007

5.2 xAI

5.2.1 Feature Ablation

In **Figure 5.12**, the plots demonstrate the ranking of features based on their absolute average feature importance, while the scatter plots show the distribution of samples according to the importance of their feature values. First, all samples are used and then the positive and negative features are used separately to get a clearer picture.

It is evident that the most important features in all three plots are tp, ndvi, smi, lst_day, lst_night and rh, as expected. For a sample that is positively correlated, increasing its value is very likely to increase the wildfire danger, while for negatively correlated features, an increase in their value will lead to the opposite effect.

Looking at the three plots in detail, we can observe that for some features the interpretation is what we would expect. For instance, ndvi is highly negatively correlated with wildfire danger, i.e. healthier vegetation reduces the wildfire risk. Unfortunately, this does not apply to all features. For example, in all three plots suggest tp is highly positively correlated, i.e. more precipitation is leads to an increased wildfire danger. This is counterintuitive.

The usage of a different baseline, such as the mean of negative samples, did not seem to fix this problem, which is why this should be investigated by the use of other xAI techniques, as seen in other sections later.

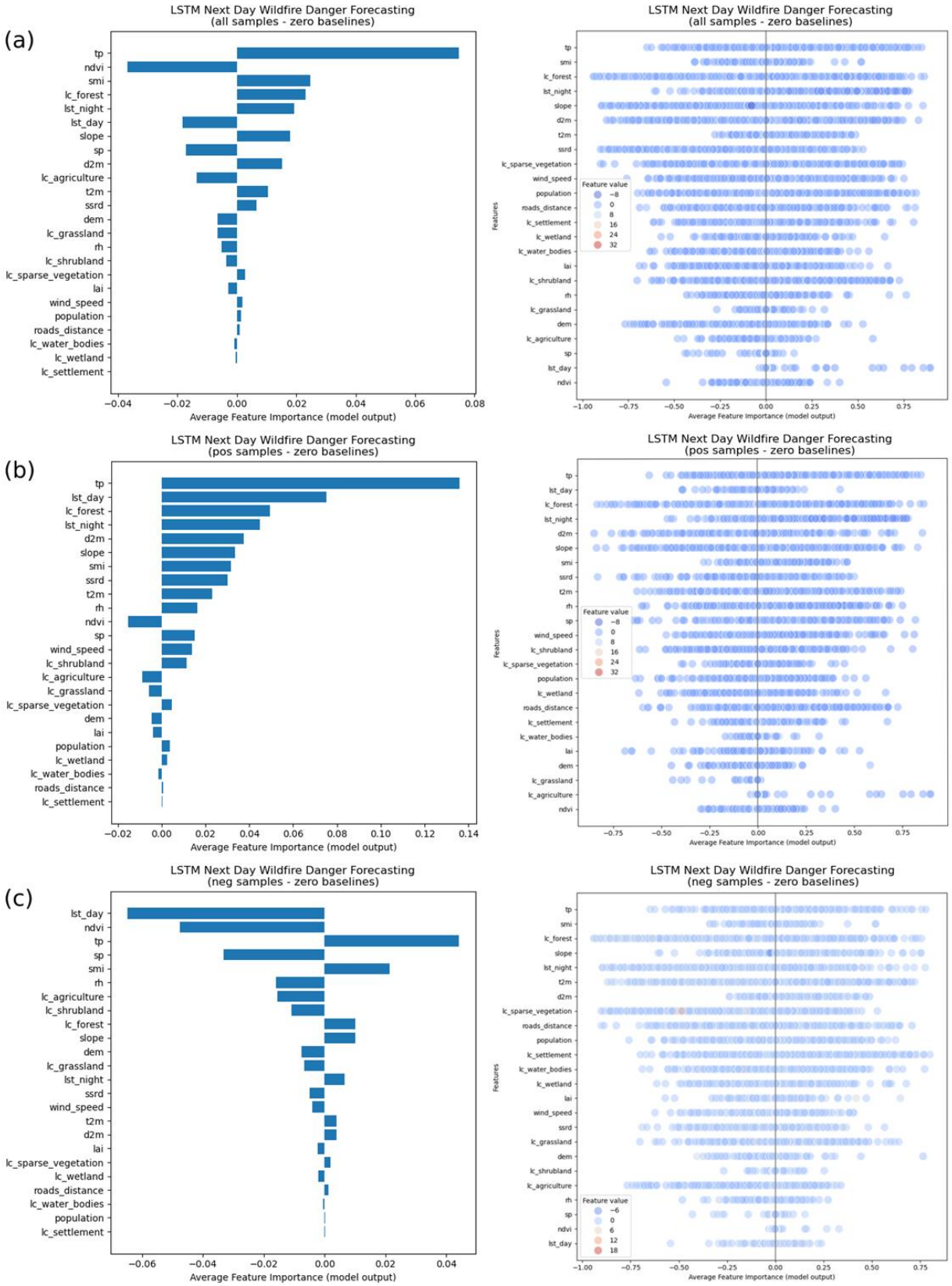


Figure 5.12: Feature ablation bar plots and scatter plots for next day forecasting using the LSTM baseline with (a) all, (b) only positive and (c) only negative test samples

5.2.1.1 LSTM Next Day Forecasting With 5, 10 and 30 Days Lag

The zero baseline is used on all samples.

For all three lags, as shown in **Figure 5.13**, the tp, ndvi, smi, lc_forest and sp seem to be the most important features. However, tp is still interpreted as a positively correlated variable irrespective of the lag used, while other variables that are conventionally connected to a high wildfire danger, such as lst_day or wind_speed, exhibit a very small to minor effect on wildfire risk.

Again, using different baselines did not bring any changes.

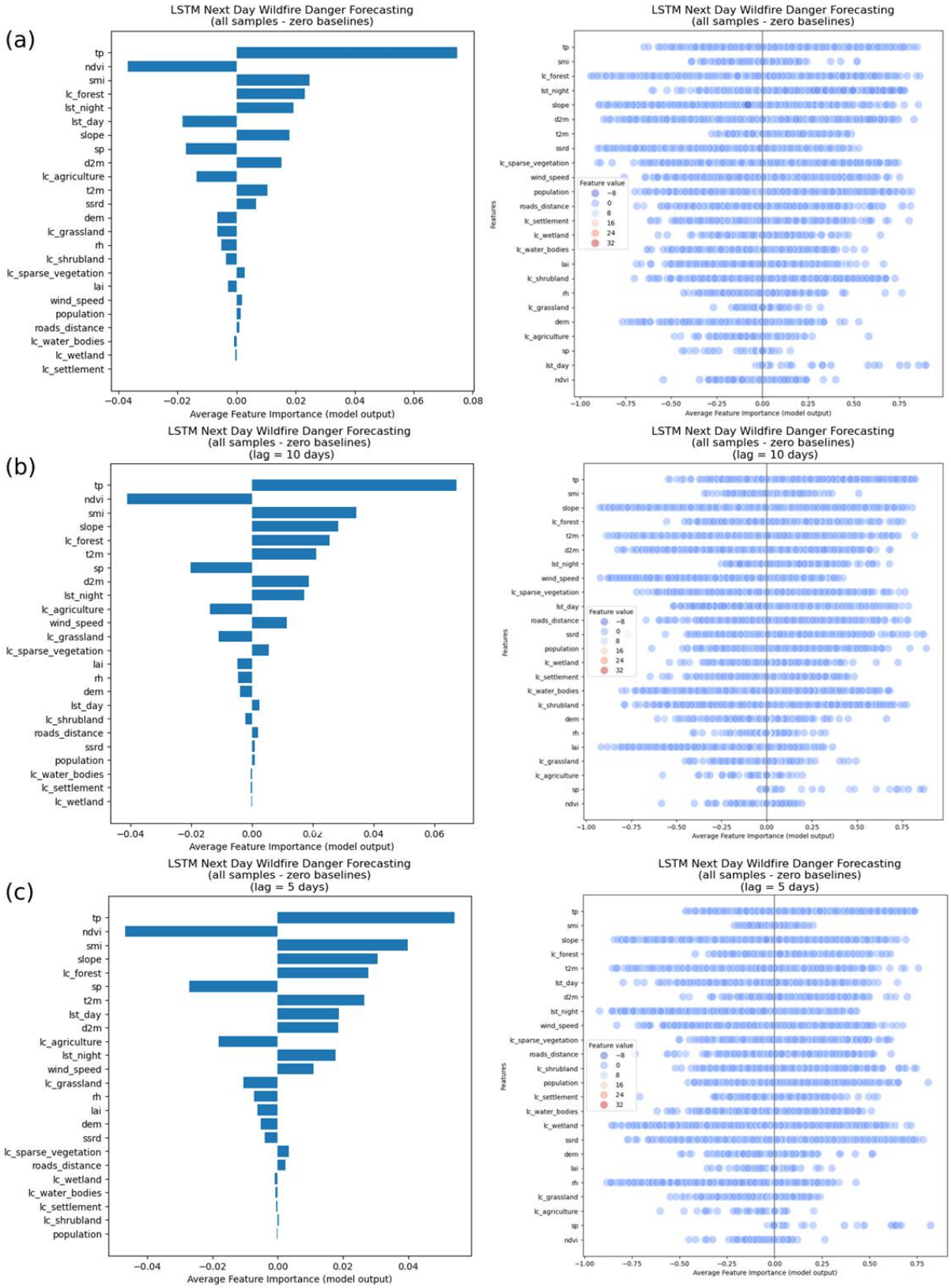


Figure 5.13: Feature ablation bar plots and scatter plots for next day forecasting using the LSTM baseline with (a) 30, (b) 10 and (c) 5 days lag

5.2.1.2 LSTM 1, 5 and 10-step Direct Forecasting with 20 Days Lag

The zero baseline is used on all samples.

As shown in **Figure 5.14**, for all three forecast horizons, the ndvi, tp, smi, slope and ssrd are the most important features.

Moreover, the importance of tp decreases as the forecast horizon expands, however, the correlation attributed to tp is once again illogical. The same applies to smi, though its importance is increasing as the number of forecasting steps increases.

Regarding conventionally important fire drivers, the lst_day is mistakenly interpreted as negatively correlated, while the attribution given to wind_speed constantly decreases, from slightly positively correlated in 1-step to slightly negatively correlated in 10-step forecasting.

Just like in previous experiments, the usage of a different baseline was not helpful.

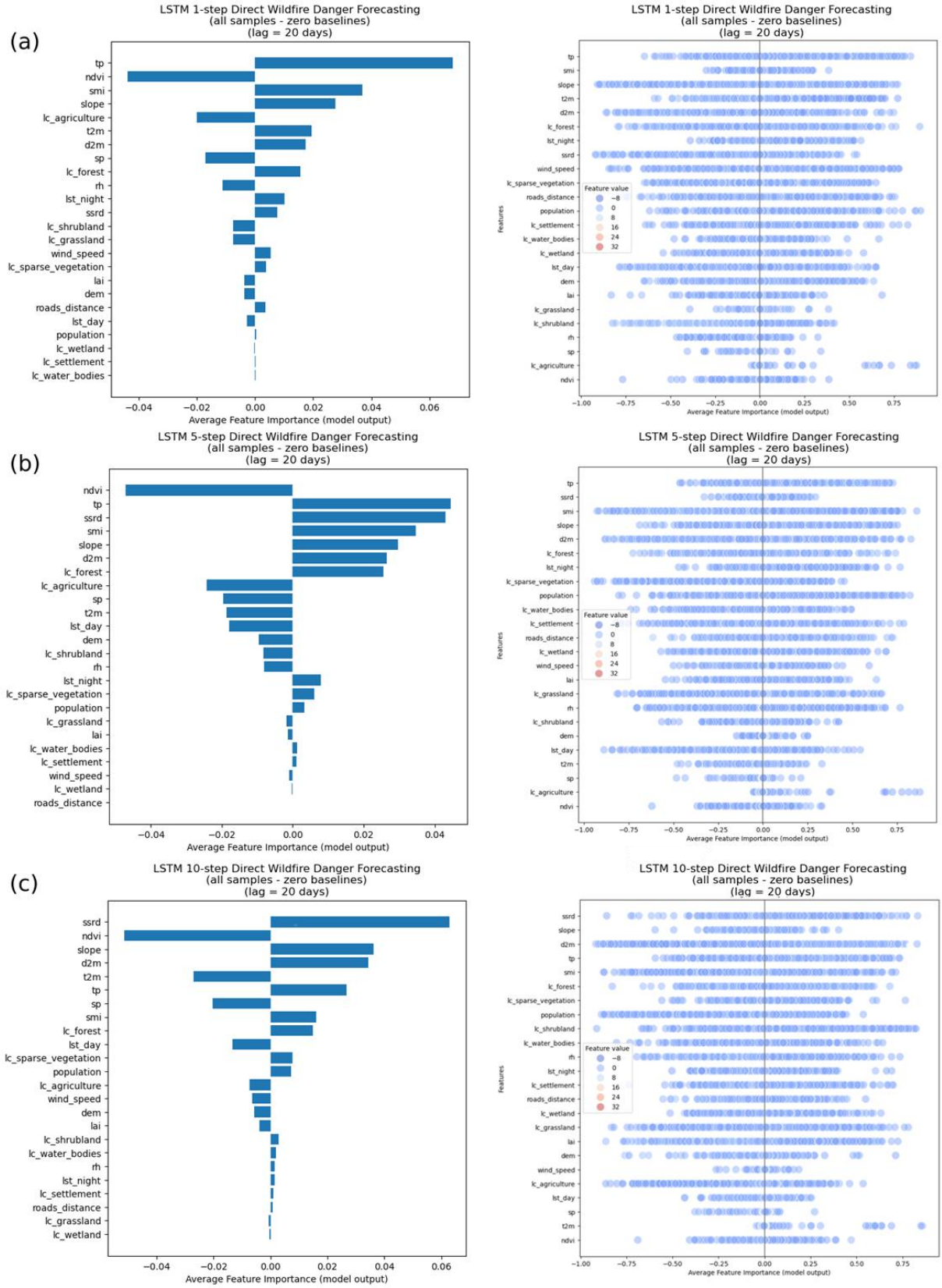


Figure 5.14: Feature ablation bar plots and scatter plots for (a) 1, (b) 5 and (c) 10-step Direct forecasting using the LSTM baseline with 20 days lag

5.2.1.3 LSTM with Uncorrelated Variables

In **Figure 5.15**, all samples are firstly used and then the positive and negative features are used separately.

Unlike in previous experiments, the tp and ndvi are now perceived correctly. Conversely, the smi and rh are not interpreted correctly, while the model seems to be indifferent to different wind_speed values.

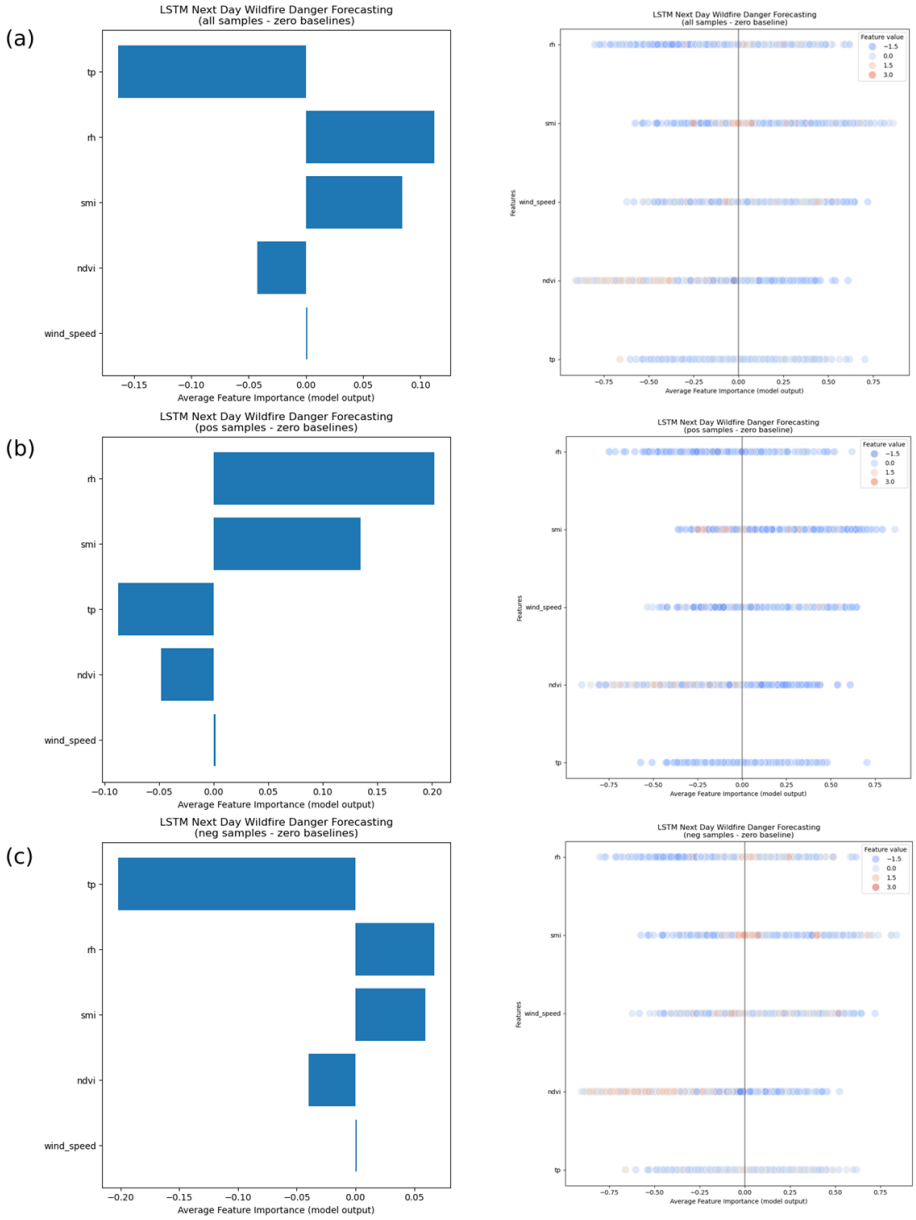


Figure 5.15: Feature ablation bar plots and scatter plots for next day forecasting using the LSTM baseline with 5 uncorrelated variables with (a) all, (b) only positive and (c) only negative test samples

5.2.2 Partial Dependence Plots (PDPs)

5.2.2.1 LSTM Next Day Forecasting With 5, 10 and 30 Days Lag

For this and the following two subsections, we focus only on the Partial Dependence Plots of 4 satellite-derived and 4 meteorological features that are the most important fire drivers conventionally: lst_day, lst_night, ndvi, smi, t2m, tp, rh and wind_speed.

First, regarding the lst_day plots in **Figure 5.16**, it is evident that they come into agreement with the ones presented in [12]. When the lag is reduced from 30 days to 10 or 5 days, not many changes are reported. Nevertheless, in all cases, the lst_day feature seems to be of great importance as it can increase the wildfire danger from almost 0 to 60%.

Similarly, the t2m plots show that higher temperatures increase the wildfire risk, as expected, but not as much as lst_day.

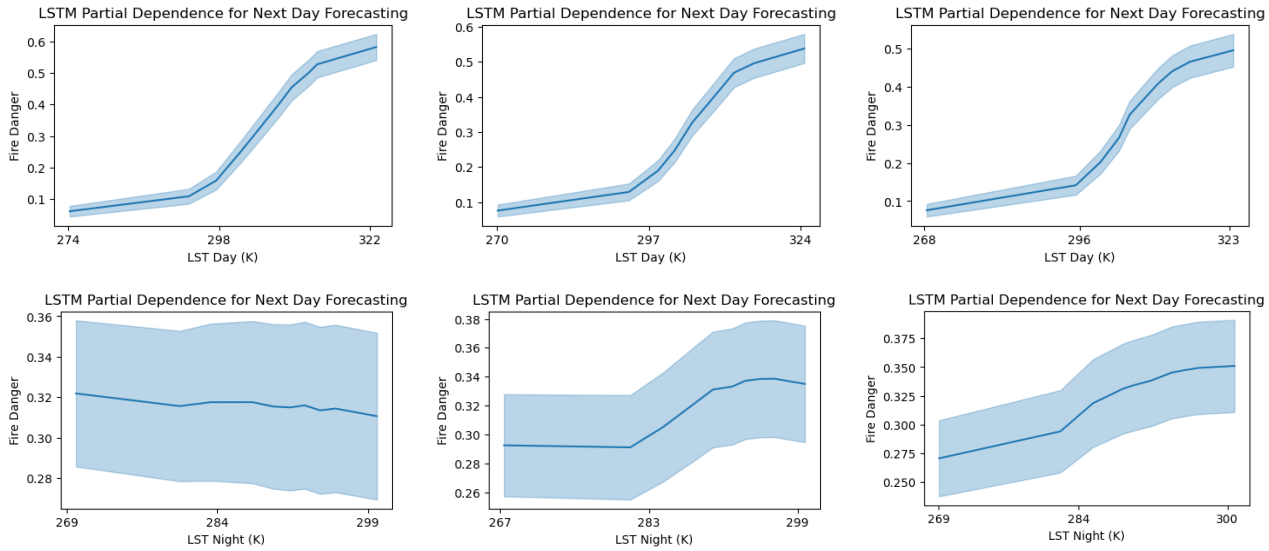
The lst_night plots gradually get closer to what we would expect as the lag decreases, however, it is not as important, as the wildfire danger constantly stays under 40% for all values. Nonetheless, lst_night influences the interactions of other variables.

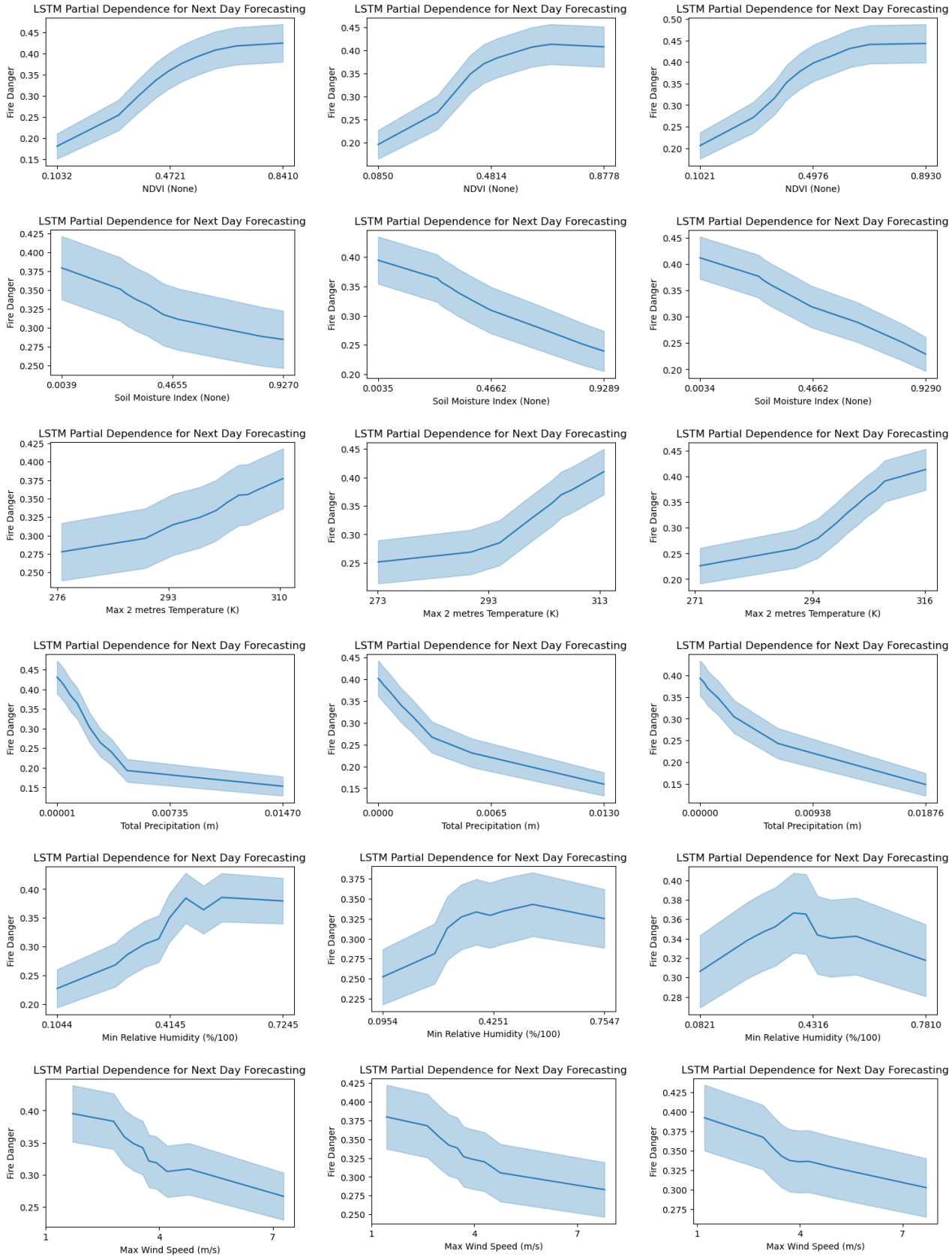
Looking at the tp plots, we can observe that the interpretation given makes sense, as high values of tp eliminate wildfire risk, coming into conflict with our previous results.

For rh, for larger lags, the dependence curve does not express what we would normally expect. However, for 5 days lag, rh, while not being able to eliminate fire danger, it sure does play a role in decreasing it for values higher than 0.25, like in [12].

The results for wind_speed are very counterintuitive, unlike in [12] and for ndvi it is very abnormal that the plot implies that healthier vegetation increases the wildfire danger. This could be caused by interactions and correlations between other fire drivers.

Lastly, for smi, the results are similar to those in [12].





(a) **(b)** **(c)**

Figure 5.16: Partial Dependence Plots (mean and 95% confidence interval) for next day forecasting using the LSTM baseline with (a) 30, (b) 10 and (c) 5 days lag

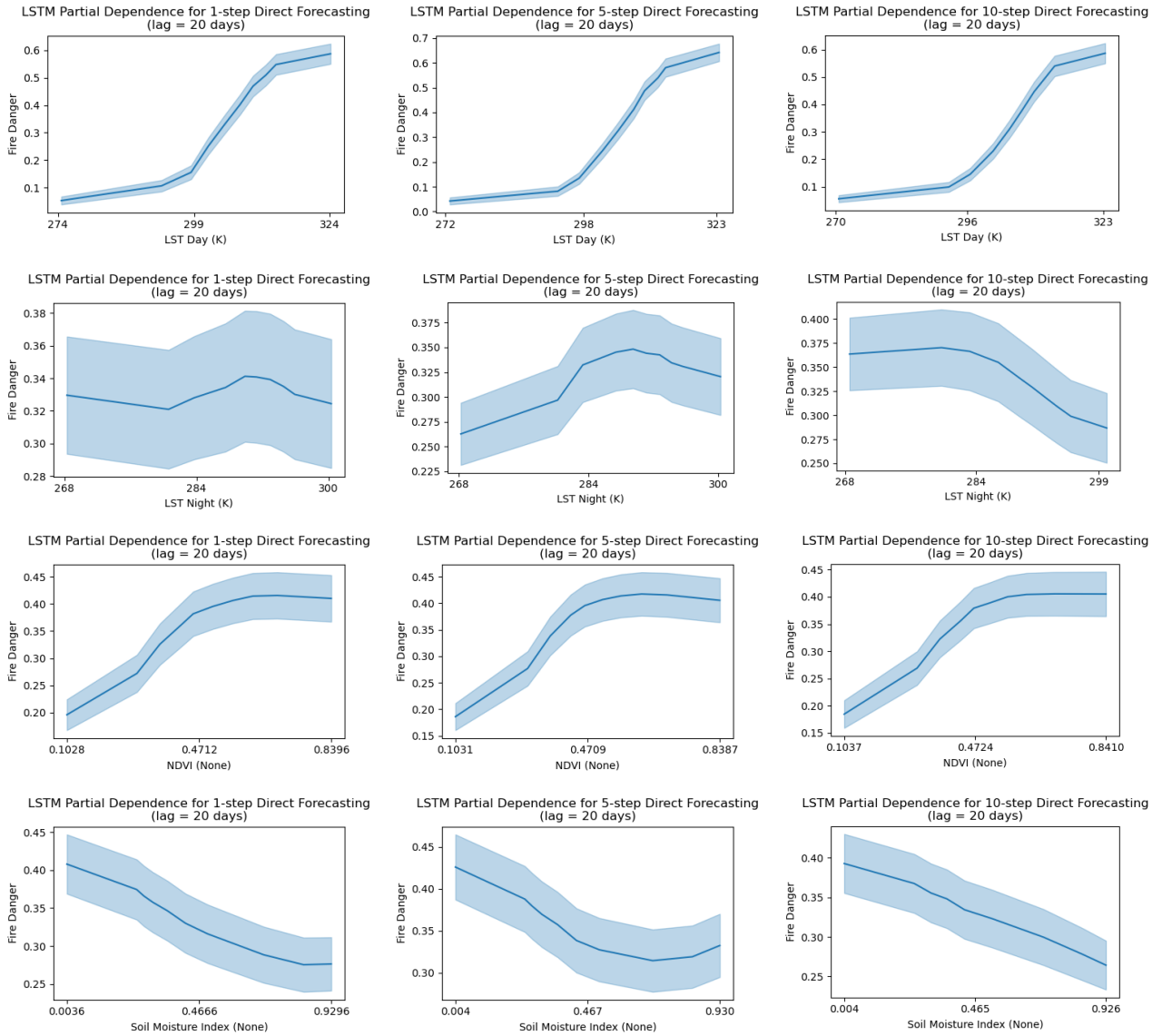
5.2.2.2 LSTM 1, 5 and 10-step Direct Forecasting with 20 Days Lag

The lst_day partial dependence and importance is not affected by expanding the forecast horizon, as seen in **Figure 5.17**.

Regarding the lst_night plots, only for 5-step forecasting the partial dependence is the one that is expected. Also, regardless of the number of forecasting steps, the lst_night still is a less important feature. Similarly, for t2m, only the partial dependence plot for 1-step forecasting makes sense.

The plots for tp are similar to those in previous subsections. For rh and wind_speed, while the partial dependencies are not affected by changing the forecast horizon, the produced plots are still faulty and counterintuitive. Hence, it is inferred that various interactions between variables are mostly retained, even when trying to forecast multiple steps ahead.

The plots for smi slightly change as the number of steps increase, but the results are still acceptable.



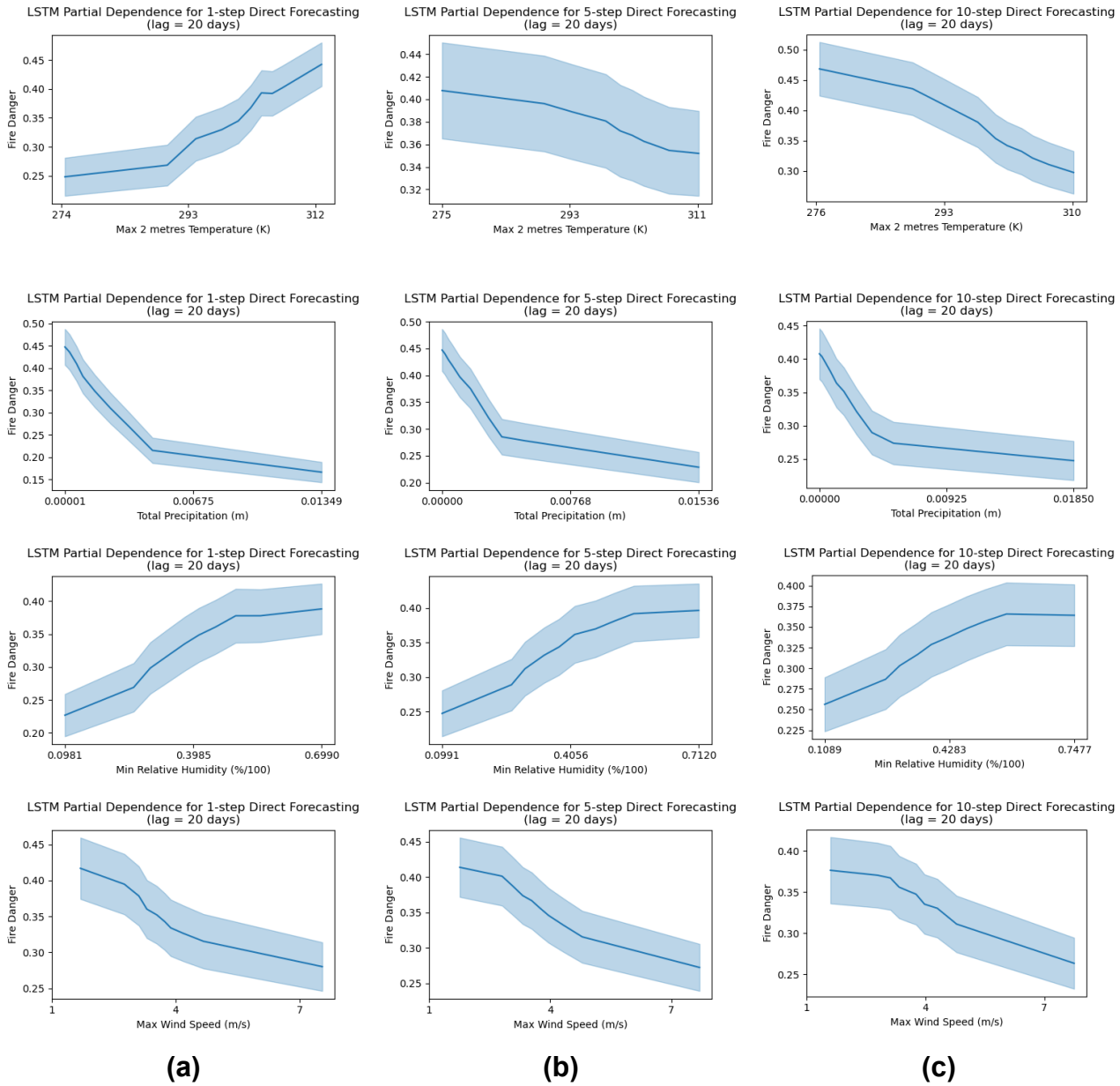


Figure 5.17: Partial Dependence Plots (mean and 95% confidence interval) for a) 1, (b) 5 and (c) 10-step Direct forecasting using the LSTM baseline with 20 days lag

5.2.2.3 LSTM with Uncorrelated Variables

The plots in **Figure 5.18**, show how the partial dependencies for a set of 5 uncorrelated variables: ndvi, rh, wind_speed, smi, tp.

In particular, the plots in **Figure 5.15** and **Figure 5.18**, the plots for ndvi and smi are quite similar, while the plot for wind_speed still follows a decreasing trend. Conversely, plots for rh and tp have improved and now can be logically explained.

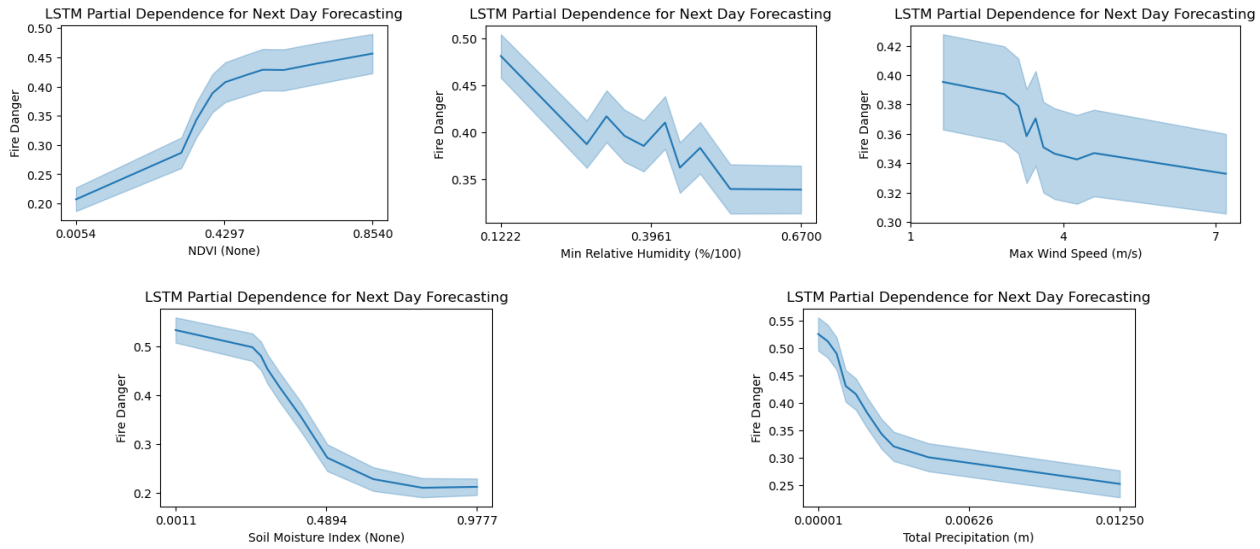


Figure 5.18: Partial Dependence Plots (mean and 95% confidence interval) for next day forecasting using the LSTM baseline with 5 uncorrelated variables

5.2.3 Integrated Gradients (IGs)

5.2.3.1 LSTM Next Day Forecasting With 5, 10 and 30 Days Lag

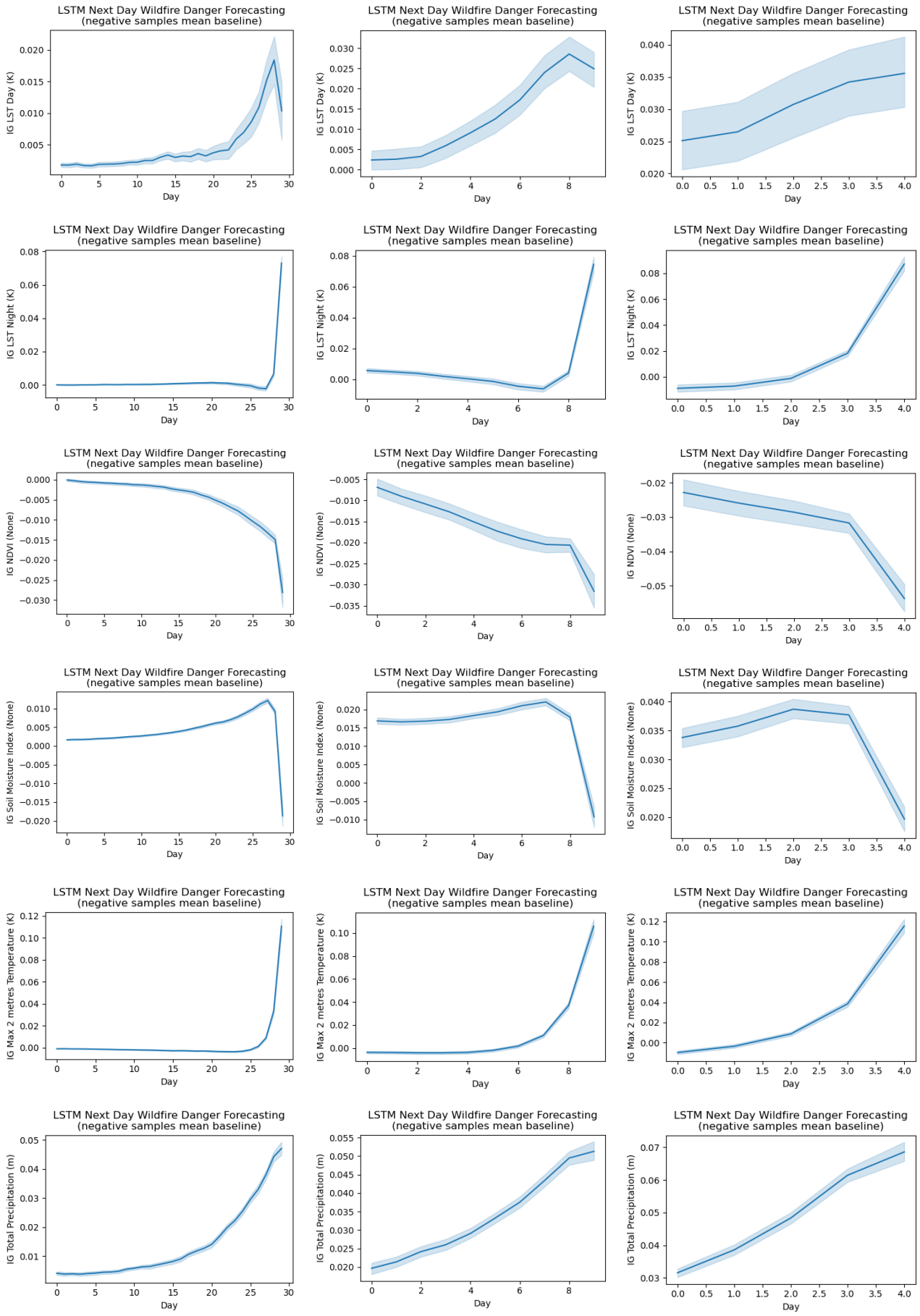
Just like in the previous section, for this and the following two subsections, we focus only on the Integrated Gradients of the 8 most conventionally important fire drivers.

Looking at the plots in **Figure 5.19**, we observe that the Integrated Gradients for most variables stay close to zero for a long time, until they abruptly increase or decrease during the last 2 or 3 days.

For the tp and ndvi variables, the Integrated Gradients are constantly changing as we get closer to the day of ignition.

At this point, comparing the lst_day and t2m plots, it is evident that, while lst_day becomes less important during the last few days, the model pays a lot more attention to t2m instead. Therefore, it is possible that the most recent meteorological data might be more important than satellite-derived data for the model's prediction.

Additionally, since the Partial Dependence Plots of lst_night imply that it is not an important feature, the Integrated Gradients are probably not that reliable.


LSTM Next Day Wildfire Danger Forecasting
(negative samples mean baseline)
IG LST Day (K)
IG LST Night (K)
IG NDVI (None)
IG Soil Moisture Index (None)
IG Max 2 metres Temperature (K)
IG Total Precipitation (m)
Day

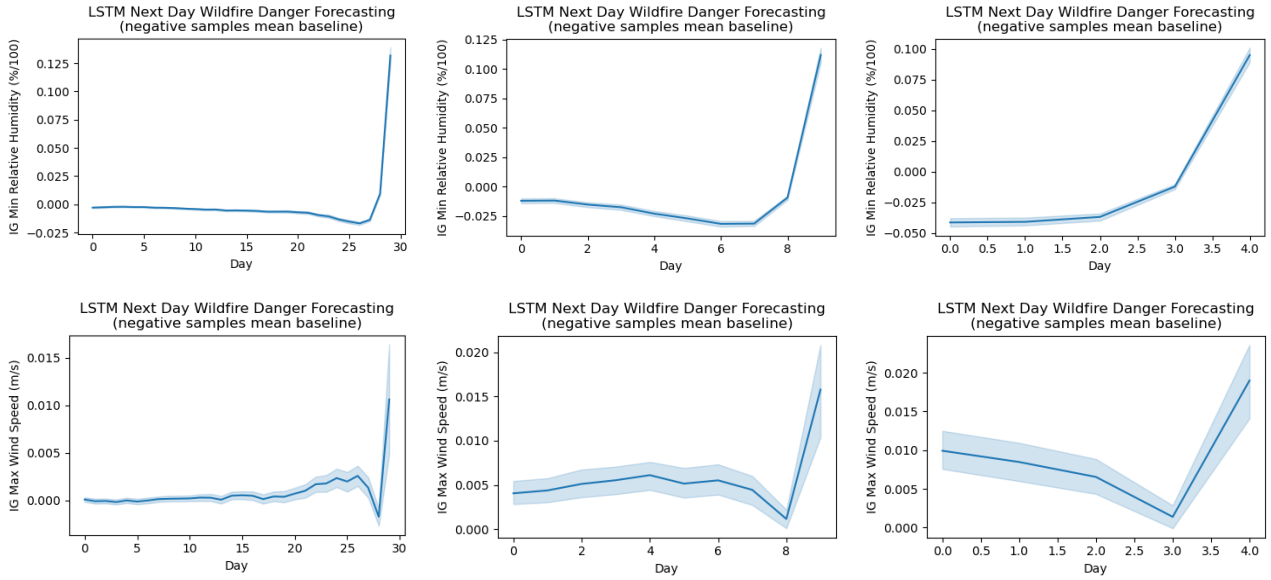
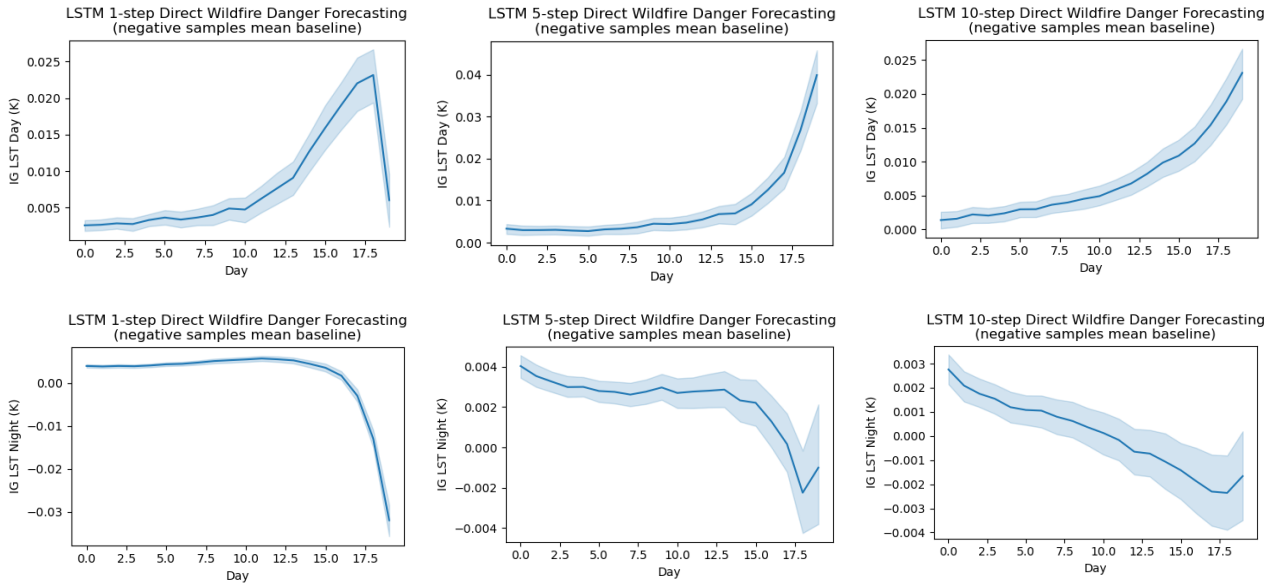


Figure 5.19: Integrated Gradients (mean and 95% confidence interval) for next day forecasting using the LSTM baseline with (a) 30, (b) 10 and (c) 5 days lag

5.2.3.2 LSTM 1, 5 and 10-step Direct Forecasting with 20 Days Lag

In **Figure 5.19**, for 1-step forecasting, the lst_day feature appears to be less important than t2m, while the opposite is observed for 5 and 10-step forecasting. Thereby, the model probably relies more on satellite-derived data for 1-step forecasting and on meteorological data for longer forecast horizons.

Moreover, it is evident that fire drivers, with fast temporal evolution, such as lst_day, gain more higher attributions than features that are changing at a slower rate, such as smi.



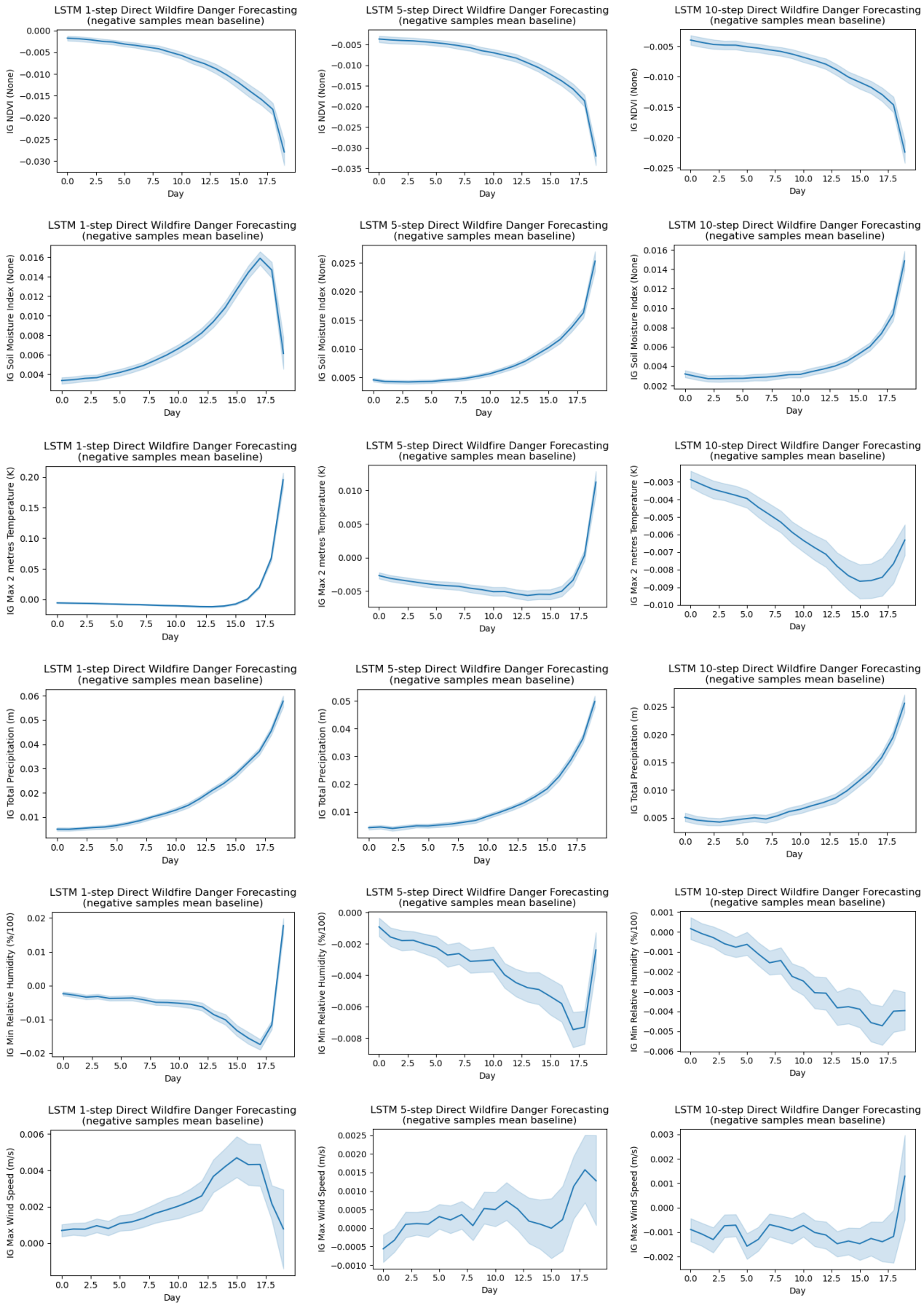


Figure 5.20: Integrated Gradients (mean and 95% confidence interval) for (a) 1, (b) 5 and (c) 10-step Direct forecasting using the LSTM baseline with 20 days lag

5.2.3.3 LSTM with Uncorrelated Variables

According to **Figure 5.21**, the Integrated Gradients for tp are constantly decreasing until 5 days before the ignition, where they start increasing again.

Regarding the rest of plots, only the rh is increasing in terms of attribution the last 5 days.

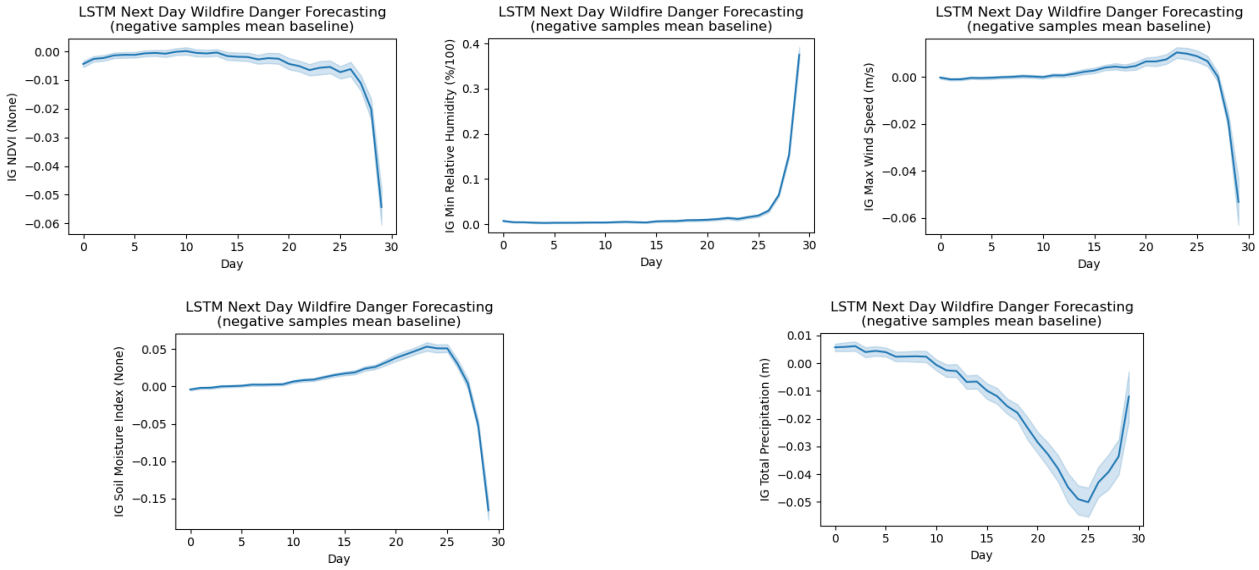


Figure 5.21: Integrated Gradients (mean and 95% confidence interval) for next day forecasting using the LSTM baseline with 5 uncorrelated variables

6. CONCLUSIONS

In this work, we approached the multi-step wildfire forecasting task in three distinct ways: using the Iterative and Direct forecasting strategies and using an alternative model, the Handoff Forecast LSTM, inspired by the Google Flood Hub system.

The findings suggest that the Iterative method is very erroneous and would be better avoided for our problem. Considering the Direct forecasting strategy, it yields much better results, though it is still prone to error for very small lags or very long forecast horizons. Nevertheless, the Handoff Forecast LSTM was proved to be pretty robust to changes in both lag and number of steps, which states the benefit of using forecasts for meteorological features, instead of solely relying on hindcasts.

The ablation study based on lag showed that we can reduce it, while still achieving scores that are comparable to the initial ones. For Iterative Forecasting, the optimum lag is at least 5 days, though, as mentioned earlier, the error accumulation is very high. For Direct forecasting with less than 4 steps ahead, the optimum lag is at least 7 days, which is roughly 25% of the original lag, while for 5-step forecasting, the ideal lag is at least 10 days, 40% of the initial lag. For the Handoff Forecast LSTM, the best lag was found to be at least 5 days for forecasting a maximum of 4-steps ahead, which is approximately 20% of the original.

From the results of ablation study based on satellite-derived and static data for 1, 5 and 10-step Direct forecasting using the LSTM and 1, 5 and 10-step forecasting using the Handoff Forecast LSTM, it is inferred that the satellite-derived data are of great importance and the most important static features are the land cover data for both models and all steps.

Regarding the xAI methods applied to the LSTM, the feature ablation results indicate that the model fails to correctly interpret some conventionally important fire drivers, such as total precipitation and wind speed. These same wildfire covariates are neither portrayed correctly in Partial Dependence Plots, regardless of lag or forecast horizon. Applying the same methods to the same model trained using 5 uncorrelated variables did not fix the problem, which implies that the cause of this might be lying in the model's architecture.

The Integrated Gradient plots show that the last 2 or 3 days before the wildfire ignition are the most important days for predicting the wildfire danger, regardless of lag or forecast horizon. Also, from the same plots we can conclude that the LSTM sometimes relies more on satellite-derived data than on meteorological data for its predictions.

7. FUTURE WORK

There is still room for improvement for the Iterative forecasting task, by introducing various error correction methods to ensure more accurate predictions of feature values. However, this method is still the most error prone by definition than the rest of methods described.

The Handoff Forecast LSTM seems very promising so far in forecasting the wildfire danger for multiple steps ahead. Apart from its very robust architecture, it is also very modular, allowing for easy extension by introducing new heads for regression or classification subtasks. In this direction, apart from predicting the wildfire danger at the last step, we could also make use of the last output of the hindcast LSTM and all outputs of the forecast LSTM to predict satellite-derived feature values at each forecast step. This could open new opportunities for the prediction of other wildfire-related extreme events at the same time, such as heatwaves or droughts.

Concerning the xAI techniques, we might want to explore different baselines that could be more suitable for our dataset, as they could help improve the feature importance results and thus, investigate the wildfire covariates in depth. Alternatively, we might want to try out more sophisticated xAI methods, specifically designed for time series models.

ABBREVIATIONS – ACRONYMS

ANN	Artificial Neural Network
AUPRC	Area Under Precision-Recall Curve
CE	Cross Entropy
CNN	Convolutional Neural Network
ConvLSTM	Convolutional Long Short-term Memory
FP	Feature Permutation
FWI	Fire Weather Index
GRU	Gated Recurrent Unit
GTN	Gated Transformer Network
IG	Integrated Gradients
LOAN	Location-aware Adaptive Normalization Layer
LSTM	Long Short-term Memory
MLP	Multi-layer Perceptrons
MSE	Mean Squared Error
NDVI	Normalized Difference Vegetation Index
NKUA	National and Kapodistrian University of Athens
NOA	National Observatory of Athens
NTUA	National Technical University of Athens
PDP	Partial Dependence Plot
RF	Random Forest
SHAP	Shapley Values
SVM	Support Vector Machine
xAI	Explainable Artificial Intelligence

REFERENCES

[1] J. Ruffault, T. Curt, V. Moron, R. M. Trigo, F. Mouillot, N. Koutsias, F. Pimont, N. Martin-StPaul, R. Barbero, J. – L. Dupuy, A. Russo, and C. Belhadj-Khedher. Increased likelihood of heat-induced large wildfires in the Mediterranean Basin. *Scientific Reports*, vol. 10, no. 1, p. 13790, Aug. 2020.

[2] M. Turco, J. J. Rosa-Cánovas, J. Bedia, S. Jerez, J. P. Montávez, M. C. Llasat, and A. Provenzale. Exacerbated fires in Mediterranean Europe due to anthropogenic warming projected with nonstationary climate-fire models. *Nature Communications*, vol. 9, no. 1, p. 3821, Oct. 2018.

[3] P. Jain, S. C. P. Coogan, S. G. Subramanian, M. Crowley, S. Taylor, and M. D. Flannigan. A review of machine learning applications in wildfire science and management. *Environmental Reviews*, vol. 28, no. 4, pp. 478–505, Dec. 2020.

[4] Y. Zhang, P. Tino, A. Leonardis, and K. Tang. A Survey on Neural Network Interpretability. *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 5, no. 5, pp. 726-742, Oct. 2021.

[5] C. E. Van Wagner. Structure of the Canadian forest fire weather index. *Information Canada Department of the Environment, Canadian Forest Service*, Publication No. 1333, 1974.

[6] M. Reichstein, G. Camps-Valls, B. Stevens, M. Jung, J. Denzler, N. Carvalhais, and K. Prabhat. Deep learning and process understanding for data-driven Earth system science. *Nature*, vol. 566, no. 7743, pp. 195–204, Feb. 2019.

[7] G. Zhang, M. Wang, and K. Liu. Forest Fire Susceptibility Modeling Using a Convolutional Neural Network for Yunnan Province of China. *International Journal of Disaster Risk Science*, vol. 10, no. 3, pp. 386–403, Sep. 2019.

[8] G. Zhang, M. Wang, and K. Liu. Deep neural networks for global wildfire susceptibility modelling. *Ecological Indicators*, vol. 127, p. 107735, Aug. 2021.

[9] F. Huot, R. L. Hu, M. Ihme, Q. Wang, J. Burge, T. Lu, J. Hickey, Y.-F. Chen, and J. Anderson. Deep Learning Models for Predicting Wildfires from Historical Remote-Sensing Data. arXiv:2010.07445 [cs.CV], Feb. 2021.

[10] I. Prapas, S. Kondylatos, I. Papoutsis, G. Camps-Valls, M. Ronco, M. – Á. Fernández-Torres, M. P. Guillem, and N. Carvalhais. Deep Learning Methods for Daily Wildfire Danger Forecasting. arXiv:2111.02736 [cs], Nov. 2021.

[11] I. Prapas, S. Kondylatos, and I. Papoutsis. FireCube: A Daily Datacube for the Modeling and Analysis of Wildfires in Greece. Zenodo, 2021; https://zenodo.org/records/6475592 [Accessed 12/07/2024]

[12] S. Kondylatos, I. Prapas, M. Ronco, I. Papoutsis, G. Camps-Valls, M. Piles, M. – Á. Fernández-Torres, and N. Carvalhais. Wildfire Danger Prediction and Understanding with Deep Learning. *Geophysical Research Letters*, vol. 49, no. 17, p. e2022GL099368, Sep. 2022.

[13] S. Kondylatos, I. Prapas, G. Camps-Valls, and I. Papoutsis. Mesogeos: A multi-purpose dataset for data-driven wildfire modeling in the Mediterranean. *In Advances in Neural Information Processing Systems*, vol. 36, pp. 50661–50676, Curran Associates, Inc, Dec. 2023.

[14] S. Kondylatos, I. Prapas, G. Camps-Valls, and I. Papoutsis. Mesogeos: A multi-purpose dataset for data-driven wildfire modeling in the Mediterranean. Zenodo, 2023; https://zenodo.org/records/8036851 [Accessed 12/07/2024]

[15] M. H. Shams Eddin, R. Roscher, and J. Gall. Location-aware Adaptive Normalization: A Deep Learning Approach for Wildfire Danger Forecasting. *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–18, 2023.

[16] K. Boyd, K. H. Eng, and C. D. Page. Area under the Precision-Recall Curve: Point Estimates and Confidence Intervals. *Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2013. Lecture Notes in Computer Science*, vol 8190. Springer, Berlin, Heidelberg, 2013.

[17] S. G. Makridakis, S. C. Wheelwright, and R. J. Hyndman, Forecasting: Methods and Applications. Wiley, 3rd edition, 1998. ISBN: 978-0-471-53233-0.

[18] S. Ben Taieb, G. Bontempi, A. Atiya, and A. Sorjamaa. A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition. *Expert Systems with Applications*, vol. 39, no. 8, pp. 7067-7083, Jun. 2012.

[19] J. De Stefani, "Towards multivariate multi-step-ahead time series forecasting: A machine learning perspective", Doctoral Dissertation, Université Libre de Bruxelles, 2022; https://dipot.ulb.ac.be/dspace/bitstream/2013/340052/4/De_Stefani_Thesis_Published.pdf [Accessed 12/07/2024]

[20] R. J. Hyndman, and G. Athanasopoulos. Forecasting: principles and practice, OTexts: Melbourne, Australia, 3rd edition, 2021; https://otexts.com/fpp3/ [Accessed 12/07/2024]

[21] M. Marcellino, J. H. Stock, and M. W. Watson. A comparison of direct and iterated multistep AR methods for forecasting macroeconomic time series. *Journal of Econometrics*, vol. 135, no. 1–2, pp. 499-526, Nov. 2006.

[22] M. Gauch, F. Kratzert, D. Klotz, G. Nearing, J. Lin, and S. Hochreiter. Rainfall–Runoff Prediction at Multiple Timescales with a Single Long Short-Term Memory Network. *Hydrology and Earth System Sciences*, vol. 25, no. 4, pp. 2045–2062, Apr. 2021.

[23] S. Nevo, E. Morin, R. A. Gerzi, A. Metzger, C. Barshai, D. Weitzner, D. Voloshin, F. Kratzert, G. Elidan, G. Dror, G. Begelman, G. Nearing, G. Shalev, H. Noga, I. Shavitt, L. Yuklea, M. Royz, N. Giladi, L. N. Peled, O. Reich, O. Gilon, R. Maor, S. Timnat, T. Shechter, V. Anisimov, Y. Gigi, Y. Levin, Z. Moshe, Z. Ben-Haim, A. Hassidim, and Y. Matias. Flood Forecasting with Machine Learning Models in an Operational Framework. *Hydrology and Earth System Sciences*, vol. 26, no. 15, pp. 4013-4032, Aug. 2022.

[24] G. Nearing, D. Cohen, V. Dube, M. Gauch, O. Gilon, S. Harrigan, A. Hassidim, D. Klotz, F. Kratzert, A. Metzger, S. Nevo, F. Pappenberger, C. Prudhomme, G. Shalev, S. Shenzis, T. Y. Tekalign, D. Weitzner, and Y. Matias. Global Prediction of Extreme Floods in Ungauged Watersheds. *Nature*, vol. 624, no. 8004, pp. 559-563, Mar. 2024.

[25] C. Molnar. Interpretable Machine Learning: A Guide for Making Black Box Models Explainable, 2nd edition, 2022; https://christophm.github.io/interpretable-ml-book [Accessed 12/07/2024]

[26] A. Carrillo, L. F. Cantú, and A. Noriega. Individual Explanations in Machine Learning Models: A Survey for Practitioners. arXiv:2104.04144 [cs:LG], Apr. 2021.

[27] P. Linardatos, V. Papastefanopoulos, and S. Kotsiantis. Explainable AI: A Review of Machine Learning Interpretability Methods. *Entropy*, vol. 23, no. 1, p. 18, Dec. 2020.

[28] N. Kokhlikyan, V. Miglani, M. Martin, E. Wang, B. Alsallakh, J. Reynolds, A. Melnikov, N. Kliushkina, C. Araya, S. Yan, and O. Reblitz-Richardson. Captum: A unified and generic model interpretability library for PyTorch. arXiv:2009.07896 [cs:LG], Sep. 2020.

[29] J. H. Friedman. Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics*, vol. 29, no. 10, Oct. 2001.

[30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, vol. 12, pp. 2825-2830, 2011.

[31] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic Attribution for Deep Networks. *International Conference on Machine Learning*, Mar. 2017.