Міністерство освіти і науки України
Чернівецький національний університет імені Юрія Федьковича

Кваліфікаційна наукова праця
на правах рукопису

СИДОР ПЕТРО ОЛЕГОВИЧ

УДК 004.9:004.8]:504.4

ДИСЕРТАЦІЯ

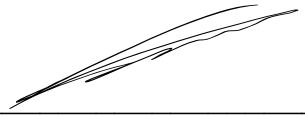
МЕТОДИ ПРОГНОЗУВАННЯ ПРИРОДНИХ КАТАСТРОФ НА ОСНОВІ ТЕХНОЛОГІЙ ШТУЧНОГО ІНТЕЛЕКТУ

121 – інженерія програмного забезпечення

12 – Інформаційні технології

Подається на здобуття наукового ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.



_____ П.О. Сидор

Науковий керівник Виклюк Ярослав Ігорович, доктор технічних наук, професор

Чернівці – 2024

АНОТАЦІЯ

Сидор П.О. Методи прогнозування природних катастроф на основі технологій штучного інтелекту. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 121 – «Інженерія програмного забезпечення» – Чернівецький національний університет імені Юрія Федьковича, Чернівці, 2024.

У сучасних умовах глобалізації та зростаючої мобільності, туризм стає однією з найбільш популярних сфер діяльності, що вимагає підвищеної уваги до питань безпеки туристів. З огляду на збільшення частоти природних катастроф, таких як лісові пожежі, урагани, землетруси та повені, стає очевидною необхідність розробки нових інформаційних технологій, які дозволять туристам і організаторам подорожей ефективно планувати маршрути з урахуванням потенційних ризиків. Особливу актуальність у цьому контексті має створення математичних моделей та методів прогнозування кризових явищ, що здатні передбачати розвиток небезпечних ситуацій та їх вплив на туристичні напрямки.

Штучний інтелект та інструменти аналізу великих даних відкривають нові можливості для побудови таких моделей. Використання історичних даних про кліматичні зміни та реальних показників природних явищ дозволяє значно покращити точність прогнозів і забезпечити своєчасне попередження про можливі загрози. Таким чином, сучасні інформаційні технології на основі систем штучного інтелекту, методів машинного навчання та математичного моделювання можуть стати важливим інструментом для підвищення рівня безпеки туристів, сприяючи більш раціональному та безпечному плануванню подорожей у різні регіони світу.

Дисертація присвячена актуальній проблемі розробки і вдосконалення інформаційних технологій для передбачення природних

катастроф з використанням сучасних досягнень у галузі штучного інтелекту та математичного моделювання. Дослідження зосереджене на аналізі, оцінці та покращенні методів прогнозування таких явищ, як лісові пожежі, урагани та паводки, з метою підвищення точності та надійності інформації про потенційні загрози, що є критично важливим для забезпечення безпеки населення та ефективного реагування на надзвичайні ситуації.

У роботі розглянуто широкий спектр математичних моделей та алгоритмів, включаючи лінійні моделі, системи з нечіткою логікою (ANFIS), нейромережі, в тому числі глибоке навчання і LSTM-мережі, для вирішення поставлених задач. Основна увага приділяється розробці та адаптації цих методів для конкретних умов і особливостей природних явищ, враховуючи великі обсяги даних, їх нестабільність та складність процесів, що моделюються.

Результати дисертації включають в себе розробку нових математичних моделей та алгоритмів для прогнозування природних катастроф, вдосконалення існуючих методів, а також розробку інформаційної технології, що демонструє практичну значимість і ефективність запропонованих підходів. Реалізація і апробація розроблених методів показали їх високу ефективність та потенціал для використання у реальних умовах, що підтверджується застосуванням результатів дослідження у вирішенні практичних завдань з прогнозування природних катастроф.

Мета та задачі дослідження. Метою роботи є розробка інноваційної інформаційної технології для планування безпечних туристичних подорожей, яка базується на передових методах прогнозування природних катастроф з використанням технологій штучного інтелекту. Ця технологія має інтегрувати математичні моделі прогнозування кризових явищ, таких як лісові пожежі, урагани та паводки, в інформаційні системи для визначення рівня загрози та оповіщення користувачів про потенційні

ризики в конкретних географічних локаціях з метою підвищення рівня безпеки та інформованості туристів, забезпечення можливості своєчасної адаптації та коригування туристичних маршрутів відповідно до прогнозованих умов навколишнього середовища, тим самим зменшуючи ризик негативного впливу природних катастроф на туристичний досвід.

В основу дисертаційної роботи покладені методи: лінійні моделі – для аналізу простих залежностей між метеорологічними факторами та інцидентами лісових пожеж; ANFIS – для прогнозування лісових пожеж, з врахуванням різноманітних чинників, таких як погодні умови та ландшафт; нейронні мережі – для моделювання та прогнозування лісових пожеж на основі великих даних про погодні умови та інші впливові фактори; LSTM – прогнозування лісових пожеж, на основі часових рядів даних про погоду та інші екологічні параметри; кореляційний аналіз – для виявлення зв'язків між погодними умовами та виникненням ураганів, а також аналіз взаємозв'язку між різними параметрами, що впливають на паводки та лісові пожежі; R/S аналіз – для дослідження довготривалих залежностей у даних про лісові пожежі; дерево рішень – класифікація та прогнозування паводків; ансамбль моделей – для поліпшення точності прогнозів паводків та лісових пожеж, забезпечуючи більш надійне та точне визначення ризиків.

Отримано наступні наукові результати:

1. Вдосконалено математичні моделі прогнозування лісових пожеж на основі ANFIS, ANN та LSTM шляхом застосування кореляційного та лагового аналізу з використанням сплайн-інтерполяції для виявлення часових затримок між піками сонячної активності та виникненням пожеж, що дозволило підвищити точність прогнозування до 93% для великих та 92% для малих пожеж.
2. Вдосконалено математичні моделі прогнозування ураганів на основі LSTM, нейронних мереж (ANN) та лінійних моделей шляхом застосування лагового аналізу для виявлення взаємозв'язку між піками

сонячної активності та інтенсивністю ураганів, що дозволило досягти високої точності прогнозування до 92% і поліпшити відтворення динаміки ураганів ($R^2 = 0,99$ для LSTM).

3. Вдосконалено ансамблі класифікаційних моделей та моделі на основі дерев рішень для прогнозування паводків шляхом встановлення взаємозв'язку між піками сонячної активності та паводками, що дозволило підвищити точність прогнозування до 97% (на 1 день вперед) і 92% (на 9 днів вперед).
4. Розроблено MLOps технологію для систем з малим і середнім обсягом вхідних даних шляхом впровадження імперативної моделі, що дозволило знизити складність виконання програмних потоків, незважаючи на відхилення від традиційних принципів програмування (DRY).
5. Вдосконалено UML модель через інтеграцію координатора, що забезпечило створення універсальної платформи "все в одному", особливо ефективної для роботи з малими та середніми обсягами даних.

Практичне значення отриманих результатів полягає у тому, що:

1. Розроблені методи прогнозування природних катастроф інтегровано в інформаційні технології, що сприятиме плануванню безпечних туристичних маршрутів. Це дозволяє користувачам уникати потенційно небезпечних районів, забезпечуючи вищий рівень безпеки під час подорожей.
2. Оперативне отримання прогнозів про ризики природних катастроф може сприяти своєчасному вживанню заходів цивільним захистом та службами порятунку для мінімізації наслідків для населення та інфраструктури.
3. Розроблені методи прогнозування забезпечують цінний інструментарій для органів державної влади, екологічних організацій та бізнес-структур у питаннях природоохоронної діяльності та управління ризиками природних катастроф.

4. Інтеграція розроблених методів у інформаційні системи та мобільні додатки сприятиме поширенню важливої інформації серед населення, забезпечуючи краще розуміння ризиків та необхідності підготовки до можливих надзвичайних ситуацій.
5. Результати роботи впроваджено в Управлінні інвестиційної політики та туризму департаменту регіонального розвитку Чернівецької обласної державної адміністрації, ГС «РТО «Гостинна Буковина» та інституті географії Сербської академії наук та мистецтв.

Ключові слова: Методи машинного навчання, Інтелектуальна система, Прийняття рішень, Галузева геоінформаційна система, Система підтримки прийняття рішень, Методи оптимізації, Нейронні мережі, Аналіз даних, Геопросторова симуляційна модель, Ансамбль машинного навчання, Математичне моделювання, Екологічні та техногенні ризики, Складні мережі, Нелінійні задачі.

ABSTRACT

Sydor P.O. Methods of predicting natural disasters based on artificial intelligence technologies. – Qualifying scientific work on manuscript rights.

Dissertation for the Doctor of Philosophy degree in specialty 121 – "Software Engineering" – Yuri Fedkovich Chernivtsi National University, Chernivtsi, 2024.

In today’s globalized world with increasing mobility, tourism has become one of the most popular industries, requiring heightened attention to the safety of travelers. Given the rising frequency of natural disasters such as wildfires, hurricanes, earthquakes, and floods, the need for the development of new information technologies that allow tourists and travel organizers to effectively plan routes with potential risks in mind has become evident. Particularly relevant in this context is the creation of mathematical models and forecasting methods for crisis events, capable of predicting the development of dangerous situations and their impact on tourist destinations.

Artificial intelligence and big data analysis tools open new opportunities for building such models. Using historical data on climate change and real-time indicators of natural phenomena significantly improves the accuracy of predictions and ensures timely warnings about potential threats. Thus, modern information technologies based on artificial intelligence, machine learning and mathematical modeling can become vital tools for enhancing tourist safety, contributing to more rational and secure travel planning across different regions of the world.

The dissertation is devoted to the actual problem of developing and improving information technologies for predicting natural disasters using modern achievements in the field of artificial intelligence and mathematical modeling. Research focuses on analyzing, evaluating, and improving forecasting methods for events such as wildfires, hurricanes, and floods to improve the accuracy and reliability of information about potential threats, which is critical to public safety and effective emergency response.

The paper considers a wide range of mathematical models and algorithms, including linear models, fuzzy logic systems (ANFIS), and neural networks, including deep learning and LSTM networks, for solving the given problems. The main attention is paid to the development and adaptation of these methods for specific conditions and features of natural phenomena, taking into account large volumes of data, their instability, and the complexity of the simulated processes.

The results of the dissertation include the development of new mathematical models and algorithms for predicting natural disasters, the improvement of existing methods, as well as the development of information technology that demonstrates the practical significance and effectiveness of the proposed approaches. The implementation and testing of the developed methods showed their high efficiency and potential for use in real conditions, which is confirmed by the application of the research results in solving practical problems of forecasting natural disasters.

The main goal of the dissertation is to develop an innovative information technology for planning safe tourist trips based on advanced forecasting methods for natural disasters, utilizing artificial intelligence technologies. This technology integrates mathematical models for forecasting crisis events, such as wildfires, hurricanes, and floods, into information systems to assess threat levels and alert users of potential risks in specific geographic locations. The objective is to enhance the safety and awareness of tourists, allowing timely adaptation and adjustment of travel routes according to forecasted environmental conditions, thereby reducing the risk of adverse impacts from natural disasters on the tourism experience.

The dissertation work is based on the following methods: linear models – for the analysis of simple dependencies between meteorological factors and forest fire incidents; ANFIS – for forecasting forest fires, taking into account various factors, such as weather conditions and landscape; neural networks – for modeling and forecasting forest fires based on big data about weather conditions and other influencing factors; LSTM – prediction of forest fires, based on time series of weather data and other environmental parameters; correlational analysis – to identify the relationship between weather conditions and the occurrence of hurricanes, as well as the analysis of the relationship between various parameters affecting floods and forest fires; R/S analysis – for investigating long-term dependencies in forest fire data; decision tree – flood classification and forecasting; an ensemble of models – to improve the accuracy of flood and forest fire forecasts, providing more reliable and accurate risk identification.

The following scientific results were obtained:

- forest fire forecasting methods were developed for the first time, based on the heliocentric hypothesis, integrating machine learning and deep learning technologies to analyze large data on weather conditions, topography, and other environmental factors, to significantly increase the accuracy and speed of forecasting.

- for the first time, hurricane forecasting methods were developed within the framework of the heliocentric hypothesis, which combines atmospheric and oceanic data into one comprehensive model, allowing more accurate estimation of the potential timing of hurricanes.
- for the first time, forecasting methods were developed for forecasting floods within the framework of the heliocentric hypothesis, which includes the integration of data from various sources (meteorological stations, satellite observations, hydrological models) and their analysis using machine learning algorithms to improve the accuracy and efficiency of forecasts.
- information technology was developed for the first time, which combines geo-information systems and intelligent data analysis, which makes it possible to form dynamic recommendations for the safety of tourists.
- the methods of pre-processing distributed heterogeneous big data, which is the basis of forming data sets and choosing optimal models for parallel calculations, have gained further development.
- the reliability of the results is determined by the evaluation of the effectiveness of the developed methods on real data, which includes a comparative analysis with existing approaches for forecasting natural disasters, demonstrating improvements in the accuracy, reliability, and promptness of forecasts.

The practical significance of the obtained results is that:

1. The developed methods of forecasting natural disasters are integrated into information technologies, which will contribute to the planning of safe tourist routes. This allows users to avoid potentially dangerous areas, providing a higher level of safety when traveling.
2. Promptly obtaining forecasts about the risks of natural disasters can contribute to the timely implementation of measures by civil protection and rescue services to minimize the consequences for the population and infrastructure.

3. The developed forecasting methods provide a valuable toolkit for state authorities, environmental organizations, and business structures in matters of environmental protection and natural disaster risk management.
4. Integration of the developed methods into information systems and mobile applications will contribute to the dissemination of important information among the population, providing a better understanding of risks and the need to prepare for possible emergencies.
5. The results of the work have been implemented in the Department of Investment Policy and Tourism of the Department of Regional Development of the Chernivtsi Regional State Administration, the GS "RTO "Hostynna Bukovyna" and the Institute of Geography of the Serbian Academy of Sciences and Arts.

Keywords: Machine learning methods, Intelligent System, Decision-making, Industry Geographic Information System, Decision Support System, Optimization Methods, Neural Networks, Data Mining, Geo-spatial Simulation Model, ML-ensemble, Mathematical Modelling, Ecological and Technogenic Risks, Complex Networks, Nonlinear Problems.

СПИСОК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМОЮ ДИСЕРТАЦІЇ:

Наукові праці, в яких опубліковані основні наукові результати дисертації

[1] Malinović-Milićević S., Vyklyuk Y., Radovanović M. M., Milenković M., Pešić A.M., Milovanović B., Popović T., Sydor P., Petrović M. D., Applying machine learning in the investigation of the link between the high-velocity streams of charged solar particles and precipitation-induced floods. Environmental Monitoring and Assessment 2024. V196. 400. ISSN: 01676-369 (Scopus, **Q2).** *(Особистий внесок: розробка математичної моделі прогнозування паводків)*

[2] Шаховська Н., Сидор П., Розроблення архітектури системи планування безпечних туристичних подорожей Вісник Хмельницького національного університету. Технічні науки. 2022. №1. (305). С.96-101 *(Особистий внесок: розробка інформаційної технології)*

[3] Сидор П.О., Виклюк Я.І., Ансамблеві моделі прогнозування повеней у Великій Британії на основі сонячної активності. Вісник Хмельницького національного університету. Технічні науки. 2024. №2. (333). С. 218-231 *(Особистий внесок: розробка математичної моделі прогнозування паводків*

Наукові праці, які засвідчують апробацію матеріалів дисертації

[4] Vyklyuk Y., Radovanović M. M., Sydor P. Hurricane Forecasting Using by Parallel Calculations & Machine Learning 2018 IEEE 1st International Conference on System Analysis and Intelligent Computing, SAIC 2018 – Proceedings 31 October 2018 Kyiv, 2018, Article number 8516872 *(Особистий внесок: Створення математичної моделі прогнозування ураганів)*

[5] Виклюк Я.І., Сидор П.О., Кунанець Н. Е., Пасічник В.В. Прогнозування лісових пожеж на основі ANFIS та паралельних розрахунків. Інтелектуальні системи прийняття рішень і проблеми обчислювального інтелекту: Матеріали міжнародної наукової конференції. Херсон: Видавництво ФОП Вишемирський В. С. с. Залізний Порт 21- 27 травня 2018 р. С.41-42 *(Особистий внесок: створення математичної моделі прогнозування лісових пожеж*

[6] Виклюк Я.І. Сидор П.О. Прогнозування лісових пожеж в Португалії. С. 25-32.

[7] Виклюк Я.І. Сидор П.О. Комп'ютерне моделювання та програмне забезпечення інформаційних систем і технологій (КМПЗ_2024) – : зб. наук. праць (тези доповідей та вибрані статті) IV Міжнародної науково–практичної конф. КМПЗ_2024. – (Чернівці, 30 травня – 01 червня 2024) / наук. ред. і відп. за вип. проф. В.М Заяць. – Львів: ЛНУ імені Івана Франка, 2024. – 342 с. *(Особистий внесок: створення математичної моделі прогнозування лісових пожеж)*

Наукові праці, які додатково відображають наукові результати дисертації

[8] Сидор П.О., Виклюк Я.І. Мобільна система інформаційної підтримки з рекомендаціями для безпечних подорожей. Науковий вісник НЛТУ України 2024. том 34. №3. С.103-109 *(Особистий внесок: алгоритм створення мобільного додатку)*

ЗМІСТ

ВСТУП

Актуальність роботи. Актуальність розробки інформаційних технологій для планування безпечних туристичних подорожей, зокрема в контексті необхідності розробки математичних методів прогнозування кризових явищ, визначається кількома ключовими факторами. Перш за все, це стрімке зростання глобальної мобільності та популярності індивідуального туризму, які вимагають від організаторів подорожей та туристів високого рівня готовності до швидкого реагування на потенційні загрози безпеці. Зміни клімату призводять до збільшення частоти та інтенсивності природних катастроф, таких як лісові пожежі, урагани, паводки, що вимагає від суспільства розробки надійних інструментів прогнозування таких кризових явищ.

Розвиток технологій штучного інтелекту, аналізу даних та математичного моделювання відкриває нові можливості для створення передових методів прогнозування, що можуть значно покращити точність та надійність інформації про потенційні ризики для туристів. Використання комплексних математичних моделей, заснованих на історичних даних та реальних показниках змін клімату, дозволяє зі значною точністю прогнозувати виникнення кризових явищ та їх потенційний вплив на конкретні туристичні напрямки.

Актуальність таких розробок полягає також у забезпеченні доступності цієї важливої інформації для широкого кола користувачів через інтеграцію у сучасні інформаційні технології, такі як мобільні додатки, веб-платформи та соціальні мережі. Це не лише сприятиме підвищенню обізнаності туристів про потенційні ризики, але й дозволить їм оперативно адаптувати свої плани подорожей з урахуванням актуальної ситуації.

Крім того, розробка математичних методів прогнозування кризових явищ має велике значення не лише для туризму, але й для загальної готовності суспільства до надзвичайних ситуацій, ефективного управління

ризиками та зниження потенційних втрат. Це вимагає міждисциплінарного підходу, об'єднання зусиль спеціалістів у галузі комп'ютерних наук, математики, метеорології, екології та інших сфер знань для створення комплексних і надійних систем прогнозування та інформування.

Отже, актуальність розробки інформаційних технологій, інтегрованих з математичними методами прогнозування кризових явищ, обумовлена зростаючими викликами глобальних змін, потребою забезпечення безпеки та комфорту туристів, а також необхідністю підвищення загальної резилієнтності суспільства до надзвичайних ситуацій.

Зв’язок роботи з науковими програмами, планами, темами. Робота виконується відповідно до планів науково-дослідницьких робіт кафедри програмного забезпечення комп’ютерних систем Чернівецького національного університету імені Юрія Федьковича за держбюджетною тематикою: «Дослідження, моделювання та розробка програмного забезпечення складних динамічних систем» (Державний реєстраційний номер 0121U109232).

Мета та задачі дослідження. Метою роботи є розробка інноваційної інформаційної технології для планування безпечних туристичних подорожей, яка базується на передових методах прогнозування природних катастроф з використанням технологій штучного інтелекту. Ця технологія має інтегрувати математичні моделі прогнозування кризових явищ, таких як лісові пожежі, урагани та паводки, в інформаційні системи для визначення рівня загрози та оповіщення користувачів про потенційні ризики в конкретних географічних локаціях з метою підвищення рівня безпеки та інформованості туристів, забезпечення можливості своєчасної адаптації та коригування туристичних маршрутів відповідно до прогнозованих умов навколишнього середовища, тим самим зменшуючи ризик негативного впливу природних катастроф на туристичний досвід.

Для досягнення вказаної мети у даній дисертаційній роботі вирішуються такі задачі:

1. Провести глибокий огляд літератури та існуючих підходів до прогнозування лісових пожеж, ураганів, паводків та інших природних катастроф з метою виявлення їхніх переваг і недоліків та існуючих програмних засобів для планування туристичних подорожей, зокрема аналіз їх функціональності та обмежень у контексті забезпечення безпеки та інформування користувачів про ризики природних катастроф.
2. Розробка нових методів прогнозування лісових пожеж, заснованих на методах штучного інтелекту, зокрема машинного навчання та глибокого навчання.
3. Розробка нових методів прогнозування ураганів за допомогою штучного інтелекту, які б враховували велику кількість факторів, включаючи атмосферні та океанічні умови.
4. Розробка нових методів прогнозування паводків, яка інтегрує дані з різних джерел та використовує прогресивні технології аналізу даних для підвищення точності прогнозів.
5. Дослідження й оцінка ефективності розроблених математичних методів та алгоритмів штучного інтелекту на реальних даних для оцінки їх ефективності та точності прогнозування.

Інтеграція розроблених методів прогнозування в інформаційну технологію для планування безпечних туристичних подорожей, що дозволить користувачам отримувати актуальну інформацію про ризики природних катастроф в обраних регіонах.

Об'єктом дослідження є процеси планування та організації безпечних туристичних подорожей з урахуванням ризиків природних катастроф.

Предметом дослідження є методи прогнозування природних катастроф (лісові пожежі, урагани, паводки) на основі технологій штучного інтелекту.

Методи дослідження. В якості апарату досліджень застосовувалися наступні методи: лінійні моделі – для аналізу простих залежностей між метеорологічними факторами та інцидентами лісових пожеж; ANFIS – для прогнозування лісових пожеж, з врахуванням різноманітних чинників, таких як погодні умови та ландшафт; нейронні мережі – для моделювання та прогнозування лісових пожеж на основі великих даних про погодні умови та інші впливові фактори; LSTM – прогнозування лісових пожеж, на основі часових рядів даних про погоду та інші екологічні параметри; кореляційний аналіз – для виявлення зв'язків між погодними умовами та виникненням ураганів, а також аналіз взаємозв'язку між різними параметрами, що впливають на паводки та лісові пожежі; R/S аналіз – для дослідження довготривалих залежностей у даних про лісові пожежі; дерево рішень – класифікація та прогнозування паводків; ансамбль моделей – для поліпшення точності прогнозів паводків та лісових пожеж, забезпечуючи більш надійне та точне визначення ризиків.

Наукова новизна отриманих результатів полягає в наступному:

вперше:

1. Розроблено MLOps технологію для систем з малим і середнім обсягом вхідних даних шляхом впровадження імперативної моделі, що дозволило знизити складність виконання програмних потоків, незважаючи на відхилення від традиційних принципів програмування (DRY).

набуло подальшого розвитку:

2. Вдосконалено математичні моделі прогнозування лісових пожеж на основі ANFIS, ANN та LSTM шляхом застосування кореляційного та лагового аналізу з використанням сплайн-інтерполяції для виявлення

часових затримок між піками сонячної активності та виникненням пожеж, що дозволило підвищити точність прогнозування до 93% для великих та 92% для малих пожеж.

3. Вдосконалено математичні моделі прогнозування ураганів на основі LSTM, нейронних мереж (ANN) та лінійних моделей шляхом застосування лагового аналізу для виявлення взаємозв'язку між піками сонячної активності та інтенсивністю ураганів, що дозволило досягти високої точності прогнозування до 92% і поліпшити відтворення динаміки ураганів ($R^2 = 0,99$ для LSTM).
4. Вдосконалено ансамблі класифікаційних моделей та моделі на основі дерев рішень для прогнозування паводків шляхом встановлення взаємозв'язку між піками сонячної активності та паводками, що дозволило підвищити точність прогнозування до 97% (на 1 день вперед) і 92% (на 9 днів вперед).
5. Вдосконалено UML модель через інтеграцію координатора, що забезпечило створення універсальної платформи "все в одному", особливо ефективної для роботи з малими та середніми обсягами даних.

Практичне значення отриманих результатів.

- Розроблені методи прогнозування природних катастроф інтегровано в інформаційні технології, що сприятиме плануванню безпечних туристичних маршрутів. Це дозволяє користувачам уникати потенційно небезпечних районів, забезпечуючи вищий рівень безпеки під час подорожей.
- Оперативне отримання прогнозів про ризики природних катастроф може сприяти своєчасному вживанню заходів цивільним захистом та службами порятунку для мінімізації наслідків для населення та інфраструктури.
- Розроблені методи прогнозування забезпечують цінний

інструментарій для органів державної влади, екологічних організацій та бізнес-структур у питаннях природоохоронної діяльності та управління ризиками природних катастроф.

– Інтеграція розроблених методів у інформаційні системи та мобільні додатки сприятиме поширенню важливої інформації серед населення, забезпечуючи краще розуміння ризиків та необхідності підготовки до можливих надзвичайних ситуацій.

Результати дисертаційної роботи впроваджено:

– в Управлінні інвестиційної політики та туризму департаменту регіонального розвитку Чернівецької обласної державної адміністрації, де застосовано комплекс методів, включаючи лінійні моделі, ANFIS та нейронні мережі для аналізу і прогнозування ризиків таких природних явищ як лісові пожежі, урагани та паводки. Ці методи допомагають оцінювати потенційні небезпеки в різних регіонах та формувати рекомендації щодо безпечних туристичних маршрутів. (акт впровадження від 14 травня 2024 р.);

– у ГС «РТО «Гостинна Буковина», де впроваджено інформаційну систему, розроблену на основі алгоритмів дисертації Сидора П.О., яка аналізує потенційні ризики природних катастроф (лісові пожежі, урагани, паводки) в регіонах пропонованих туристичних маршрутів. Система використовує прогнозні моделі для оповіщення менеджерів та клієнтів про можливі небезпеки. (акт впровадження від 2 травня 2024 р.);

– інституті географії Сербської академії мистецтв та наук де застосування передових методів прогнозування підвищило якість наукових публікацій співробітників Інституту, збільшило цитованість і міжнародне визнання, посилило міждисциплінарну взаємодію в рамках наукових проектів географії та екології. (акт впровадження від 29 квітня 2024 р.).

Результати теоретичних досліджень викладені у [1] – [8].

Особистий внесок здобувача. Усі наукові результати дисертації одержано автором самостійно.

У працях, надрукованих у співавторстві, автору належить наступне: [1], [3] – розробка математичної моделі прогнозування паводків; [2] – розробка інформаційної технології, [4] – Створення математичної моделі прогнозування ураганів, [5], [6], [7] – створення математичної моделі прогнозування лісових пожеж, [8] – алгоритм створення мобільного додатку.

Апробація матеріалів дисертації.

Матеріали дисертаційної роботи обговорювалися на міжнародних наукових конференціях. Зокрема на наукових конференціях: IEEE 1st International Conference on System Analysis and Intelligent Computing, SAIC (31 October 2018 Kyiv), Інтелектуальні системи прийняття рішень і проблеми обчислювального інтелекту (Залізний Порт 21- 27 травня 2018 р.), Комп'ютерне моделювання та програмне забезпечення інформаційних систем і технологій, КМПЗ_2024 (Чернівці, 30 травня – 01 червня 2024).

Публікації. По темі дисертації опубліковано 8 робіт, де представлені результати досліджень. З них 4 статті у рецензованих виданнях (3 – в українських фахових виданнях, 1 – у періодичному ***науковому*** виданні***, проіндексованих у наукометричних базах даних Web of Science Core Collection та/або Scopus*** Q2,), 1 – за матеріалами міжнародної конференції, що індексується у базі даних SCOPUS, в збірниках матеріалів міжнародних наукових конференцій – 3 роботи.

Структура та обсяг дисертації. Дисертація складається зі вступу, чотирьох розділів, висновків, списку використаних джерел і додатків. Дисертаційна робота має 49 рисунки, 48 таблиць, 4 додатки. Список використаних джерел містить 140 найменувань. Загальний обсяг роботи складає 239 сторінок, обсяг основного тексту – 177 сторінок.

РОЗДІЛ 1
ОГЛЯД ЛІТЕРАТУРИ ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1.Огляд програмних засобів для планування туристичних подорожей

Туризм є основною метою економічного відновлення, оскільки він може приносити великий дохід завдяки низьким інвестиціям та високій прибутковості. Багато країн, які мають туристичні визначні пам'ятки, ставлять важливі цілі для відродження економіки та постійного розвитку, зосереджені на своїх туристичних пам'ятках. Туристи з кількох країн створюють нові стилі подорожей та нові експериментальні методи пошуку. Готельний бізнес розраховує на успішних туристів для зростання. Планування туру необхідне, коли стилі групових або сімейних поїздок стають популярними та збільшуються в розмірі чи складності [1].

Туристичні інформаційні системи є невід'ємною частиною прийняття рішень щодо відвідування привабливих туристичних місць, які також називають «орієнтирами». Ранжування, послідовність, маршрутизація та вибір пам’яток для огляду протягом певного періоду часу є складними процесами з точки зору задоволення потреб усіх членів групи [2].

Туристична сфера потребує сучасних мобільних інформаційних технологій для запровадження індивідуальних туристичних маршрутів. Мобільні персональні інформаційні системи використовуються для підтримки прийняття рішень туриста під час планування подорожі. Зокрема, йде мова про побудову маршрутів. Індивідуальний туризм передбачає самостійний вибір міста, готелю, транспорту чи закладу харчування. Подорож буде сплановано самостійно за обраним маршрутом і в зручні терміни, поїздка буде включати цікаві екскурсії для кожного. Побудова туристичних маршрутів розглянута у [1 – 4].

В умовах зростаючої глобалізації та інтенсифікації міжнародного туризму, питання забезпечення безпеки під час подорожей набуває особливої

актуальності, особливо у контексті України. Враховуючи збільшення кількості туристичних поїздок, актуальним стає розроблення ефективних інформаційних систем, здатних забезпечити туристів необхідною інформацією про безпеку. Незважаючи на існування численних технологій, спрямованих на поліпшення вражень від подорожей, мало уваги приділяється систематизованому аналізу ризиків та інформуванню мандрівників про потенційні загрози. Більшість існуючих досліджень зосереджена на покращенні туристичного досвіду без звернення належної уваги до аспектів безпеки. Така ситуація вимагає розробки цілеспрямованих інструментів, які допомогли би мандрівникам оцінювати та мінімізувати потенційні ризики.

Багато дослідників вивчали планування туризму та розробку маршрутів подорожі протягом визначеного періоду часу для створення програмного забезпечення систем планування туризму. Програмне забезпечення для планування турів та розробки маршруту поїздки зосереджено на збалансуванні послідовностей відвідувань у межах часового вікна для підтримки відвідування найбільшої кількості орієнтирів з найменшою відстанню та витратами. Це допомагає мандрівникам легше приймати рішення щодо маршруту поїздки та вибору місця [3–8].

Суттєвою проблемою в плануванні туризму є задоволеність усіх учасників поїздки, яку важко вирішити. Тому різні туристичні стилі членів сім'ї створюють складну ситуацію з точки зору програмного забезпечення створення системи планування туризму. Туристичний стиль є причиною кількох факторів, що стосуються задоволеності туром. Багато факторів впливають на задоволеність учасників подорожі, наприклад, їхні різні інтереси, культура, бюджет, часові обмеження, а також уподобання в їжі та напоях [9]. Відвідування найбільшої кількості визначних пам’яток, низькі витрати, цікаві види діяльності, зручне транспортне сполучення, доступні засоби, а також низькі відстані та витрати часу є важливими факторами для планування туризму [7].

З іншого боку, безпека подорожі є тим фактором, який слабо врахований у дослідженнях. Йде мова про визначення ймовірності настання природних та антропогенних явищ та корекцію маршруту.

Складність побудови туристичного маршруту полягає у наданні користувачеві можливості його прокладання цікавими туристичними пам’ятками. Такого роду завдання можна класифікувати як задачу комбінаторної оптимізації, рішенням якої буде задача комівояжера в її незамкнутому варіанті [8]. Вирішення задач оптимізації виконують за допомогою різноманітних алгоритмів, але вони мають наступні недоліки: – всі алгоритми часто мають обмеження локальних рішень; – у якості вихідного використовується лише один варіант рішення; – кожен метод є досить чутливим до вибору умов.

Сьогодні послуги з планування самостійних поїздок надають кілька компаній, серед яких – Free Travel [2], TripAdvisor [3], а також численні невеликі сервіси та приватні особи. Сервіс Free Travel дозволяє планувати та організувати індивідуальні подорожі. На сторінках сайту можна знайти довідкову інформацію для мандрівника про транспорт, місця проживання чи визначні пам’ятки. Окрім цього сайт надає можливість оформити паспорт чи візу, придбати квиток на літак, автобус чи потяг, а також забронювати готель посилаючись на сайтипартнери. Сервіс Free Travel вважають самовчителем для початкових мандрівників. З матеріалів користувачмає змогу дізнатися як подорожувати самостійно, оскільки вся інформація заснована на особистому досвіді його засновників. На сторінках сайту можна дізнатися як заощадити час і гроші, є можливість знайти дешеві авіаквитки, квитки на автобуси чи поїзди, переглянути інформацію про готелі й інші варіанти житла, а також ознайомитися з рекомендаціями щодо відвідування міст і культурних пам’яток з їх розташуванням.

Ще одним туристичним сервісом є американський сайт подорожей TripAdvisor. Він допоможе продумати основні нюанси поїздки, подивитися

перелік визначних пам’яток й забронювати готельний номер чи столик у закладі громадського харчування. Послуги на сайті є безкоштовними для користувачів завдяки створенню ними більшої частини контенту. Саме тому сервіс можна вважати міжнародною базою експертних і користувальницьких відгуків.

Обрані додатки позиціонують себе як туристичні сервіси, але кожен має свої недоліки. Сервіс Free Travel важко назвати планувальником маршруту, оскільки на його сторінках присутня інформація рекомендаційного характеру, що не дає повної картини для організації подорожі. На сторінках сайту TripAdvisor можна ознайомитися з туристичними пам’ятками, переглянути до них відгуки реальних користувачів, а також додати деякі місця до обраних для майбутньої подорожі, що дозволяє попередньо продумати основні деталі поїздки. Головним недоліком цих додатків є те, що з їх допомогою немає можливості розробити туристичний маршрут. Підтримка вибору туристичного маршруту дозволила б користувачам досить швидко зрозуміти траєкторію свого руху, спланувати час та можливості.

Сьогодні існує багато науково-інформаційних розробок, спрямованих на потреби туриста. Зокрема у статті [10] пропонується платформа з мобільними та веб-базованими додатками для підтримки розумного та сталого туризму, що дозволяє відвідувачам та місцевій владі мінімізувати негативний вплив на довкілля та спільноту. У роботі [11] розробляється система для планування туристичних поїздок людьми з аутизмом, враховуючи їхні інтереси та когнітивні можливості, з метою зниження стресу та адаптації інтерфейсу з уникненням перевантаження інформацією. Автори дослідження [12] створили мобільний додаток для популяризації та організації культурного туризму, що виявився ефективним для надання рекомендацій подорожей і може слугувати рекомендацію для подальшого розвитку туристичних активностей. Технологія та застосунок, призначені

для підтримки глухих туристів у місцях культурної спадщини, що сприяє розвитку інклюзивного туризму та відкриває нові можливості для інновацій у секторі описується у статті [13]. Комплексна та масштабована система підтримки управління туристичними напрямками, розроблену з урахуванням збереження природи та потреб відвідувачів, засновану на принципах вантажопідйомності та системному підході до туризму та місць призначення, що включає моніторинг відвідуваності та інші географічні дані описано в роботі [14]. Однак більшість наукових досліджень в галузі розробки інформаційних систем уникають аналізу ризиків, з якими може зустрітись турист.

Ризики поділяються на природні, виробничі та соціальні. Загалом існує більше ніж 150 видів ризиків, проте цей перелік не є вичерпним. До природних ризиків належать екстремальні погодні умови, природні пожежі, отруйні рослини, небезпечні тварини, комахи, бактерії та інше. Як приклади досліджень та симуляцій серед найсерйозніших природних катастроф останніх років можна навести лісові пожежі [15], землетруси на Чілі [16] та цунамі [17]. Також тварини можуть становити загрозу, як показують дані про напади акул [18] та випадки укусів змій [19].

Промислові ризики охоплюють загрози, пов’язані з використанням транспортних засобів, підйомно-транспортного обладнання, горючих та вибухонебезпечних матеріалів, а також процесів, що відбуваються при високих температурах та тисках, електроенергії, хімікатів, різних видів радіації тощо. Багато промислових катастроф пов'язані з використанням ядерної енергії, як для військових, так і для цивільних цілей. Однією з останніх промислових катастроф стала аварія на АЕС Фокусіма, яка сталася через цунамі в Японії [20]. Однак, забруднення не обмежується тільки атомною енергетикою. Наприклад, забруднення повітря є причиною смерті людей щорічно по всьому світу [21].

Соціальні загрози також містять в собі не тільки тероризм, війни та злочинність, але й культурне розмаїття, що впливає на поведінкові норми, великі скупчення людей, бідність та інше. Варто підкреслити, що в останні роки спостерігається збільшення ризику терористичних актів, заснованих на фактичних подіях[22]. Іншим важливим прикладом суспільної загрози є військова агресія, наприклад російське вторгнення на півдні України, яке продовжується і на сьогодні [23]. Як зазначено раніше, навіть натовп може нести загрозу[24].

1.2. Інформаційні системи для визначення рівня загрози та оповіщення користувачів

Сучасні технології та мобільні застосунки відіграють ключову роль у підвищенні безпеки людей, інформуючи їх про різноманітні небезпеки, включаючи погодні умови та терористичні загрози. Ось декілька мобільних інформаційних систем, які забезпечують користувачів актуальними сповіщеннями про потенційні небезпеки:

- *Попередження про небезпеку погоди:* Мобільні застосунки, такі як *NOAA Weather Radar, AccuWeather* та *Windy*, надають користувачам сповіщення про різні погодні умови, включаючи урагани, бурі, зливи та інше. Ці застосунки використовують різні джерела даних для надання точних та своєчасних прогнозів.
- *Попередження про лісову пожежу:* Застосунки, як *BC Wildfire* та *Wildfire Analyst Pocket,* спеціалізуються на наданні інформації про лісові пожежі, їх місцеположення, інтенсивність та напрямок поширення.
- *Попередження про небезпеку тероризму:* Застосунки *Terror Alert*, *TerrorMate* та *Новини про тероризм WTA* фокусуються на сповіщенні користувачів про терористичні загрози та атаки, надаючи важливу інформацію, яка може допомогти уникнути потенційно небезпечних районів.

Один з аналізованих застосунків, *NOAA Weather Radar and Alerts* [25], розроблений компанією Apalon, є потужною системою для прогнозування погоди та оповіщення про погодні небезпеки. Він надає користувачам детальну інформацію про погодні умови, включаючи суворі погодні умови та відстеження ураганів, та володіє функціоналом, який дає змогу користувачам отримувати сповіщення про екстремальні погодні явища у своїй місцевості або вказаних локаціях (таблиця 1.1).

Ці мобільні інформаційні системи забезпечують важливий засіб для підвищення обізнаності та підготовки до потенційних небезпек, надаючи користувачам вживати необхідних заходів безпеки заздалегідь.

AccuWeather [26] відома своєю високою точністю у прогнозуванні погоди, що робить її незамінним інструментом для мільйонів користувачів по всьому світу. Ця платформа використовує передові технології та алгоритми для аналізу погодних умов, забезпечуючи детальні та актуальні прогнози. Її девіз – "Щоб врятувати життя, захистити власність і допомогти людям процвітати, одночасно розширюючи AccuWeather як здоровий і прибутковий бізнес" – підкреслює зобов'язання компанії щодо надання важливої інформації, яка може допомогти людям вжити необхідних заходів для зниження ризиків, пов'язаних з погодними умовами.

Система сповіщень AccuWeather (таблиця 1.1.) активується в наступних випадках, забезпечуючи користувачів важливою інформацією про екстремальні погодні явища:

- *Дощ:* Коли очікується понад 12,7 мм опадів, що може вказувати на значний дощ або зливу, спричиняючи можливі затоплення або інші проблеми.
- *Сніг:* Сповіщення надсилаються, коли прогнозується понад 2,54 мм снігу, що може призвести до ускладнень у дорожньому русі та інших зимових проблем.
- *Ожеледь:* Сповіщення про ожеледь надсилаються, коли очікується більше ніж 0,254 мм льоду, що значно збільшує ризик аварій та падінь.

Таблиця 1.1

Функціональне порівняння інформаційних систем оповіщення про погоду

Ім'я	Компанія	Сповіщення про погоду	Прогнози ураганів	Карта	Сповіщення електронно
NOAA Weather Radar	Apalon	+	+	+	–
AccuWeather	AccuWeather	+	–	+	–
Windyty	Windyty	+	+	+	+

• *Постійний вітер:* Сповіщення про сильний вітер видаються, коли швидкість вітру перевищує 48 км/год, що може спричинити шкоду будівлям та інфраструктурі.

• *Пориви вітру:* Якщо очікується, що пориви вітру перевищать 64 км/год, надсилаються сповіщення про ризик сильних поривів, які можуть завдати ще більшої шкоди.

• *Імовірність грози:* Коли існує 75% ймовірність грози, користувачі отримують сповіщення, оскільки грози можуть супроводжуватися небезпечними явищами, такими як сильні дощі, блискавки та навіть торнадо.

AccuWeather використовує ці критерії для сповіщення своїх користувачів про потенційні небезпеки.

Windy [26], також відомий як Windyty, є високо цінуваним інструментом для візуалізації погодних умов, який забезпечує користувачів детальними прогнозами погоди та сповіщеннями про погодні умови. Цей інструмент є надзвичайно корисним для широкого спектра користувачів, від любителів природи до професіоналів, які залежать від точних погодних прогнозів для своєї діяльності, включаючи пілотів, парапланеристів та парашутистів (таблиця 1.1.).

Ось декілька ключових особливостей Windy, які роблять його таким корисним інструментом [27]:

• *Візуалізація погодних умов:* Windy надає інтерактивні карти, що дають можливість користувачам візуально оцінити погодні умови в різних частинах світу. Це містить інформацію про тип небезпечних погодних умов, швидкість вітру, кількість опадів, температуру, хмарність та час або тривалість погодних явищ.

• *Прогноз хвиль:* Особливо цінною є здатність Windy надавати прогнози хвиль, що допомагає користувачам визначити, чи безпечно займатися діяльністю на воді, як-от плавання, серфінг або вітрильний спорт, у морі, океані, озерах чи річках.

• *Інформація про різну висоту:* Однією з унікальних особливостей Windy є здатність надавати інформацію про погодні умови на різних висотах, що є надзвичайно корисним для людей, зайнятих у повітряних видів спорту або діяльності, таких як пілотування літаків, парапланеризм та парашутизм.

Використання Windy може значно підвищити безпеку та ефективність планування діяльності на відкритому повітрі, надаючи користувачам доступ до точних і актуальних погодних даних. Ця платформа є важливим ресурсом не тільки для тих, хто шукає розваги на свіжому повітрі, але й для професіоналів, чия робота або хобі залежать від погодних умов.

1.3. Огляд методів прогнозування лісових пожеж

Лісові пожежі є важливою екологічною проблемою, особливо тому, що адекватних заходів їх запобігання не існує, точніше, здатності запобігти поширенню вогню. Немає спільної думки про походження багатьох лісових пожеж. Згідно аналізу даних FAO, в Європі на період з 1999 по 2001 рік зареєстровано 42,7 % випадків виникнення пожеж, причини яких невідомі [28].

Як показано в роботі [29], в середньому причинами спалахів 58,8 % загальної кількості лісових пожеж в країнах Балканського півострова на період з 1988 по 2004 рік є людський фактор, 3,3 % — природний і 37,9 % мають невідоме походження. Найбільший відсоток виникнення пожеж людського походження був зафіксований в Хорватії (75,3 %) і найменший — в Болгарії (30,4 %). З іншої сторони, Болгарія має найбільший відсоток невідомих причин виникнення лісових пожеж (67,9 %).

Загальна кількість лісових пожеж, зафіксованих у Португалії, склала 25 221 в 2011 році, при чому 40 % зафіксованих випадків – пожежі з невідомих причин, а у Німеччині з 888 пожеж в тому ж році – 48 %. З іншого боку, в Угорщині причиною 95 % пожеж є діяльність людини. В цілому, дослідження щодо лісових підпалів у 2011 році здійснювалося Територіальним Гарнізоном італійського лісового корпусу, в результаті чого відбулися судові засідання над 455 людьми, в тому числі 9 було заарештовано або взято під варту за звинуваченням у скоєні підпалу лісу. Загальна кількість лісових пожеж, зафіксованих в Італії у 2011 році, склала 8 181 [30].

Джерела, з яких дані були завантажені для вивчення (число пожеж в США), показують, що всі пожежі, які сталися, належать до людської діяльності (85,5%) чи удару блискавки (14,5%). Відомо, що блискавка може бути також важливим фактором, яка спричиняє виникнення пожежі [31-33]. Тим не менш, існують багато інших гіпотез. Адже блискавки, в основному, з’являються з появою опадів, а кількість опадів у таких ситуаціях визначає, чи вогонь поширюватиметься чи згасне [32]. Однак, відсутність більш детальних досліджень на цю тему залишає відкритим наступне питання: в якій мірі електричні розряди відіграють роль у початковій фазі явища пожежі? В роботі [33] стверджується: «В період з 1990 по 1998 рік в Арізоні і Нью-Мексико на федеральній землі США під час пожежонебезпечного сезону з квітня по жовтень спостерігалося більше 17 000 природніх пожеж. Удари блискавок, пов’язаних з цими пожежами, склали менше, ніж 0,35%

усіх зареєстрованих випадків загоряння, що сталися під час пожежонебезпечного сезону протягом цього часу».

З іншого боку, згідно [34], в період з 1961 по 1993 рік в змішаних лісах в провінції Альберта (Канада) 67.6% пожеж були викликані блискавками. Останні дослідження показали, що блискавки є причиною майже ½ аналізованих випадків спалахів лісових пожеж. «В Канаді в період з 1991 по 2000 рік з близько 8000 лісових пожеж, що сталися за рік, для 48% випадків причиною займання стала блискавка» [35], (згідно даних Канадської ради міністрів лісового господарства 2003 року). В роботі [33] висунута гіпотеза, що в Західному Сибіру майже всі пожежі викликані блискавками. Отже, можна прийти до висновку, що наявні дані в науковій літературі про вплив блискавок на спалахи лісових пожеж є суперечливими.

Як відомо, існує прямий зв'язок між відносно високими температурами повітря і місцерозташуванням пожежі, але цей факт також не має повного пояснення. Добре відомо, що необхідно, як мінімум, 300 °C для згаданої початкової фази пожежі [36]. Варто відзначити, що така висока температура повітря ніколи не спостерігається на Землі за допомогою стандартних метеорологічних пристроїв, навіть у випадку дослідження температури ґрунту.

Ґрунтуючись на цих результатах, можна побачити, що відсоток пожеж, спричинений людським фактором, є невеликим. Згідно аналізу літературних джерел, відсоток непояснених причин пожеж варіюється від ~ 43% випадків до 95% (у випадку Угорщини). Дані, які свідчать, що удари блискавок є причинами пожеж, також суперечливі. Відсотки коливаються від 0.35% у випадку Арізони і Нью-Мексико до майже 100% у Західному Сибіру. Припущення, що звичайними явищами ми можемо пояснити більшість пожеж, приведено в роботі [37]: «Крім того, ми виявили, що, при відносно великих просторових і часових масштабах (тобто, штати і століття), зміна частоти та місцерозташування пожеж практично не залежать від місцевих факторів, таких як тип рослинності, рельєф, випас худоби і підпали».

Беручи до уваги результати, представлені в роботі [38], ми висунули «геліоцентричну гіпотезу», згідно якої лісові пожежі без встановлених причин виникають в результаті масового спалювання рослин під дією заряджених частинок, що приходять до нас від сонця. Ми припустили, що місцерозташування джерела пожежі корелює з раптовим надходженням зазначених частинок до нашої планети.

1.4. Огляд методів прогнозування ураганів

Наприкінці серпня та на початку вересня 2017 року прилади на супутнику Advanced Composition Explorer (ACE) виміряли надзвичайно потужний потік частинок високої енергії. Потім, у геоефективному положенні, була коронарна діра, що піднімалася від північної полярної області Сонця через його екватор, а також енергетичні області 12671 і 12672 [39] В іншому випадку супутник розташований у точці Лагранжа, тому в реальному часі він вимірює параметри сонячного вітру (SW). Протягом першої половини вересня в геоефективному положенні з'явилося більше десятка спалахів класу M, спалах рівня X-9 і пов'язана з ним помірна подія сонячних частинок (SPE). 7 і 8 вересня 2017 року раннє настання викиду корональної маси (CME), пов'язане зі спалахом X-9, викликало сильну геомагнітну бурю. Одночасно з цими процесами на Сонці в атмосфері над Атлантикою відбуваються збурення, які переросли в урагани IRMA, JOSE та KATIA, де IRMA була одним із найбільш руйнівних ураганів, коли-небудь зареєстрованих [40].

Можна сказати, що існують численні занепокоєння як щодо виникнення циклонів, так і щодо їх поведінки в часі та просторі [41]. Аналіз коливань приземного тиску після SPEs та зниження Форбуша для Євразійського регіону показав значні варіації цього атмосферного тиску протягом принаймні перших п'яти днів після подій. Ці варіації відрізняються залежно від широти та довготи. Розрізняють клітини підвищеного та зниженого поверхневого тиску [42].

У статті [43] наводять список літератури, яка підтверджує ідею причинно-наслідкового зв’язку процесів на Сонці з ураганами, починаючи з XIX ст.

У відповідь на зміни сонячної активності спостерігається просторово неоднорідна реакція інтенсивності та частоти ураганів [44] Ходжес і Елснер [45] стверджували, що регіональна частота ураганів з 1851 по 2010 рік вказує на меншу кількість ураганів у Карибському басейні та вздовж східного узбережжя США, коли сонячних плям багато. Навпаки, менше ураганів спостерігається в центральній і східній частині Північної Атлантики, коли сонячних плям мало. Значна позитивна кореляція між усередненим *Kp*-індексом глобальної геомагнітної активності та інтенсивністю урагану, виміряною максимальною стійкою швидкістю вітру, виявлена для бароклінічних ураганів [46].

Давно помічено, що кліматичні аномалії в тропосфері, пов’язані із сонячним впливом, переважно мають стратосферне походження [47]. Помічено, що значні погодні явища, особливо якщо вони викликані системами низького тиску, мають тенденцію слідувати за надходженням високошвидкісного сонячного вітру [48]. Раніше опубліковані статистичні дані про те, що вибухонебезпечні екстратропічні циклони в північній півкулі, як правило, виникають протягом кількох днів після надходження високошвидкісних потоків сонячного вітру з корональних дір [49-51], підтверджуються для південної півкулі.

Встановлено, що спалахи сонячного космічного випромінювання призводять до збільшення тривалості елементарних синоптичних процесів в атлантико-європейському секторі Північної півкулі. Було припущено, що спостережувані варіації тривалості елементарних синоптичних процесів зумовлені впливом короткоперіодних варіацій космічного випромінювання на інтенсивність циклонічних процесів у середніх і високих широтах [52-54].

Застосовуючи вейвлет-спектральний аналіз до часових рядів ураганів, Мендоза та Пасос [55] виявили періодичності, які збігаються з основними сонячними плямами та магнітними сонячними циклами. В Атлантичному океані спостерігаються піки біля 11 і 22 років. Їх результати вказують на те, що найвищі значні кореляції виявлені між атлантичними та тихоокеанськими ураганами та *Dst* індекс. Найважливіше те, що обидва океани представляють найвищі кореляції ураганів і *Dst* під час висхідної частини непарних сонячних циклів і спадної фази парних сонячних циклів.

По-перше, можна сказати, напрочуд вдалі прогнози оприлюднив П. Корбін на 6–11 місяців вперед. Методи, які він використовував, стосувалися виключно варіацій у поведінці Сонця, його магнітного поля, корональних вивержень і флуктуаційного характеру сонячного вітру. В результаті в період з жовтня 1995 по вересень 1997 чотири з п'яти сильних штормів були точно прогнозовані. П'ятий помилився на 48 годин [56].

Виклюк та ін. [57] спробували за допомогою моделі ANFIS визначити, чи існує математичний зв’язок між потоком високоенергетичних частинок від Сонця та появою ураганів. За період 1999–2013 рр. (добові значення з травня по жовтень) із фазовим зсувом 0–3 дні було виявлено, що моделі можуть пояснити в найкращому випадку 22–26 % потенційної зв’язності. В іншій спробі Виклюк та ін. [58], за той самий період часу, використовуючи краще комп’ютерне обладнання та подовжуючи фазовий зсув від 0–10 днів, отримують кращі результати (до 39%). Автори приходять до висновку, що отримані результати не можна ігнорувати і що потрібні додаткові зусилля для пояснення причинно-наслідкових зв'язків. У цьому сенсі ми вважали, що також необхідно вивчити причинно-наслідковий зв’язок між потоком частинок від Сонця та утворенням ураганів Irma, Katya та Jose.

1.5. Огляд методів прогнозування паводків

Екстремальні погодні умови, такі як інтенсивні опади, які спричиняють повені, визнані як одна з найбільших природних загроз з серйозними

соціальними, економічними та екологічними наслідками [59]. Повені можуть спричинити втрати життя, руйнування майна, знищення врожаю та худоби. Довгострокові ефекти включають збої у постачанні питної води та електроенергії, руйнування транспортної та комунікаційної інфраструктури, а також негативний вплив на фізичне та психічне здоров'я людей через переміщення населення. Попри прогрес у розумінні процесів, що призводять до сильних опадів та можливих повеней, потреба у вдосконаленні прогнозування екстремальних погодних та гідрологічних явищ збільшується через їхні значні негативні наслідки.

Зв'язок між сонячною активністю та кліматом Землі досліджується вже понад 200 років [60]. Сонячна енергія, що досягає Землі, варіюється на різних часових масштабах і корелює з атмосферними параметрами [61], проте оцінити її вплив на кліматичні та екологічні процеси складно. Незважаючи на те, що остаточного визнання зв'язку між сонячним вітром та тропосферою досягнуто не було, численні дослідження вказують на вплив сонячної активності на клімат, включно з атмосферною циркуляцією, температурою, опадами та екстремальними погодними умовами [62-69].

Основні досліджені механізми включають прямий нагрів Землі сонячним випромінюванням і вплив УФ-випромінювання на озоновий шар стратосфери, що веде до змін в атмосферній циркуляції та кліматі [70]. Інший важливий механізм стосується впливу галактичних космічних променів, які можуть сприяти утворенню ядер конденсації хмар і таким чином впливати на хмарність [71-74].

Сонячна активність та збурення міжпланетного середовища мають прогностичне значення для розвитку екстраполічних циклонів, що є ключовими для погоди в середніх широтах. Дослідження показують зміни в циклонічній активності відповідно до сонячної активності, а також вплив сонячних протонних подій на інтенсивність опадів [75-77].

Паводки зустрічаються по всьому світу зокрема, територія Сполученого Королівства має густу дренажну мережу з приблизно 200,000 км водотоків, що дренують близько 1500 окремих басейнів [78, 79]. Ці численні водотоки переважно короткі, мілководні та чутливі до значних антропогенних змін. Режим річок визначається кліматичними умовами (особливо опадами, температурою повітря, інсоляцією), геологічними особливостями кожного водозбірного басейну (такими як їхня проникність), морфологією території та антропогенним фактором (змінами в руслах річок, використанням води, змінами у землекористуванні тощо).

У роботі [80] зазначили, що зимовий NAO (North Atlantic Oscillation) впливає на річкові потоки, контролюючи перенос вологи та тепла над Великою Британією. Laizé та Hannah [81] підкреслили, що вищий NAO індекс збільшує західні повітряні потоки через Велику Британію, що призводить до вищих, ніж в середньому, рівнів опадів і температур, а отже, і до більших річкових потоків. В той час як високогірні басейни отримують значні кількості опадів, низинні райони випробовують менші обсяги, тому на режим стоку впливають також інші фактори, такі як проникність, висота і фізичні характеристики басейну.

Водотоки Великої Британії різноманітні: від гірських потоків, що отримують до п'яти метрів опадів на рік, до низинних річок з підземним живленням на південному сході, де рівень опадів нижчий [79]. Опади у Великій Британії розподілені відносно рівномірно протягом року, але із схильністю до осінньо-зимового піку, особливо в західних басейнах. Сезонні коливання температури повітря та сонячного світла спричиняють високу випаровуваність в літній період (квітень-вересень), що впливає на внутрішньорічний розподіл стоків у річках з природними режимами. Зазвичай максимальні витрати води реєструються взимку, а мінімальні – влітку або восени. Варто зауважити, що міські водотоки були суттєво змінені та не завжди слідують цьому взірцю. Наприклад, низькі витрати

можуть штучно збільшуватись через переливання резервуарів або перекидання води між басейнами.

Вивчаючи тенденції річкових потоків за чотири стандартні сезони у період з 1969 по 2008 рік у 89 басейнах з майже природними режимами стоку у Великій Британії, Hannaford і Buys [82] зробили висновок, що спостерігається загальне збільшення зимових річкових потоків (із найбільшим зростанням у північних та західних високогірних басейнах, тоді як низькі потоки зменшились у деяких західних басейнах); регіонально послідовне зниження весняних потоків; збільшення літніх потоків (у північних та західних басейнах); і в основному слабкі позитивні та негативні тенденції (в англійських низинах); збільшення осінніх потоків (особливо для високих потоків у центральній і південно-західній Британії та на північному сході Шотландії). Спостережувані тенденції, такі як збільшення зимового стоку та зниження весняного стоку, можуть впливати на управління водними ресурсами і вказувати на збільшення ризику повеней.

1.6. Висновки до розділу 1

Було проведено всебічний огляд існуючих досліджень, програмних засобів та методів прогнозування природних катастроф, зокрема лісових пожеж, ураганів та паводків. Зіставлення різноманітних підходів і технологій, які використовуються в цих сферах, дозволило ідентифікувати ключові тренди розвитку та прогалини у дослідженнях, вказавши на потенціал для подальших інновацій. Аналіз підкреслив значення інтеграції новітніх технологій обробки даних, машинного навчання та глибокого навчання в прогнозуванні та управлінні природними ризиками, визначивши це як перспективний напрямок для зміцнення глобальної безпеки та реагування на надзвичайні ситуації.

У результаті проведених досліджень:

– Було проведено аналіз існуючих інформаційних систем, що дозволяють

індивідуалізувати маршрути відповідно до уподобань користувачів, виявляючи при цьому ключові вимоги до функціональності та інтерфейсу, це дало змогу встановити відсутність інформаційних систем які би інтегрували в плануванні подорожі аналіз потенційних кризових явищ.

- Розглянуто сучасні технологічні рішення, які забезпечують швидке попередження про потенційні небезпеки, включаючи природні катастрофи та інші ризики, з особливим акцентом на інноваційні методи аналітики та їх відмінності від традиційних систем, що дало змогу визначити необхідність покращення точності прогнозу та оперативності оповіщень.
- Досліджено використання різноманітних даних, включаючи супутникові спостереження та математичні моделі, для ідентифікації факторів, що сприяють виникненню лісових пожеж, і це дало змогу встановити відсутність інтеграції таких математичних моделей в системи планування туристичних подорожей, а також недостатню точність та низький горизонт прогнозів.
- Вивчено методи прогнозування ураганів які включали аналіз залежностей між сонячною активністю та формуванням ураганів, застосовуючи дані супутникового моніторингу для виявлення передумов таких явищ, і це дозволило визначити нові можливості для покращення методів прогнозування ураганів.
- Проведений аналіз методів прогнозування паводків показав важливість розгляду кліматичних змін та їх впливу на виникнення екстремальних погодних умов, що можуть спричинити паводки, з особливим наголосом на взаємозв'язок між сонячною активністю та кліматичними факторами, що дало змогу краще зрозуміти механізми виникнення паводків та визначити шляхи покращення точності їх прогнозування.

Основні наукові результати розділу опубліковані в працях [83, 84].

РОЗДІЛ 2
МЕТОДИ ПРОГНОЗУВАННЯ ПРИРОДНИХ КАТАСТРОФ

2.1. Лісові пожежі

Кризові явища на землі можуть бути спричинені не тільки людським фактором, а і сонячною активністю. Геліоцентрична гіпотеза припускає, що зміни в сонячній активності, такі як сонячні спалахи та викиди корональної маси, можуть мати безпосередній вплив на атмосферні явища на Землі, спричиняючи або посилюючи кризові явища, такі як лісові пожежі, урагани, торнадо та інші екстремальні погодні умови. Не вдаючись у деталі проходження частинок через атмосферу Землі, в дисертаційному дослідженні ми тестували геліоцентричну гіпотезу, щоб знайти будь-які відповідні кореляції та підтвердити чи спростувати цю гіпотезу. Для цього ми вивчали різні набори даних, про лісові пожежі в різних країнах світу за різні часові періоди. А також використовували різні моделі машинного навчання для підтвердження геліоцентричної гіпотези.

2.1.1. Лінійні та ANFIS моделі для прогнозування лісових пожеж в Португалії

2.1.1.1 Попередній аналіз структури даних

Зважаючи на те, що неможливо було безпосередньо зафіксувати можливе поширення частинок на землю, як потенційну причину, що викликає початкову фазу полум’я в палаючій рослинній масі, ми вирішили перевірити геліоцентричну гіпотезу опосередковано. Наступні усереднені за годину дані в реальному часі використовувалися як вхідні параметри: диференціальний електрон (діапазони енергій 38-53 і 175-315 кеВ) і потік протонів (діапазони енергій 47-68, 115-195, 310-580, 795-1193) і 1060-1900 кеВ), усереднені за годину об’ємні параметри густини протонів плазми сонячного вітру (р/сс), об’ємної швидкості (км/с) і температури іонів (градуси К)[15]. ACE Satellite –

Solar Wind Electron Proton Alpha Monitor розташований у точці Ла Гранж, щоб вимірювати дані в реальному часі, які надходять від Сонця до нашої планети. Погодинні метеорологічні дані, що стосуються станції Monte Real, були використані як вихідні дані (широта: 39° 49' 52" пн. ш., довгота: 8° 53' 14" зх. д.). Ця станція розташована на військовій авіабазі, розташованій поблизу Лейрії. Дані включають температуру повітря (°C), вологість (%) і тиск повітря (гПа). Усі дані, використані в роботі, стосуються періоду 15-19 червня 2017 року. Цю станцію було обрано, оскільки вона розташована поблизу охопленої пожежею території, а дані доступні в Інтернеті.

Метою розрахунків було дослідити функціональні залежності між характеристиками СВ та температурою повітря – $\boldsymbol{T}$, вологістю – $\boldsymbol{H}$ і тиском – P. Тестовані поля введення представлені в лівій частині таблиці 2.1. Крок вимірювання становив 1 година.

Рішення цієї проблеми складається з кількох етапів.

2.1.1.2. Заповнення пропущених даних

Особливістю цього набору даних є наявність пропущених даних (розрив) з максимальною тривалістю 3 години. Сплайн-інтерполяція з використанням умов відсутності вузла була використана для заповнення цих прогалин. Інтерпольоване значення в точці запиту базується на кубічній інтерполяції значень у сусідніх точках сітки в кожному відповідному вимірі [85].

2.1.1.3. Кореляційний аналіз

Проведено кореляційний аналіз для встановлення наявності лінійного зв’язку між вхідним і вихідним полями (табл. 2.1.). Як ви можете бачити з таблиці, коефіцієнти кореляції Пірсона (R) є достатньо малими у всіх випадках, крім $I1$ та $I2$. Це означає, що жодних лінійних залежностей цих даних не спостерігається. Високі значення R для $I1$ і $I2$ вказують на наявність сильно виражених нелінійних залежностей. Наявність затримки (часу) між вхідним і вихідним полями може бути ще однією причиною малих коефіцієнтів кореляції.

Таблиця 2.1

Перевірені поля введення та поля виводу та кореляція між ними

Поля введення		Кореляція (R)		
		T	*X*	*П*
Диференціальний потік частинок/см2-с-стер-МеВ, електронів				
E1	38-53	-0,11	0,14	0,16
E2	175-315	0,06	-0,02	-0,17
Диференціальний потік частинок/см2-с-стер-МеВ, протони				
P1	47-68	0,01	-0,03	0,11
P2	115-195	0,14	-0,12	-0,07
P3	310-580	0,15	-0,13	-0,17
P4	795-1193	0,11	-0,08	-0,30
P5	1060-1900 pp	-0,10	0,06	0,22
Інтегральний потік протонів				
I1	> 10 МеВ	-0,57	0,50	0,85
I2	> 30 МеВ	-0,55	0,48	0,85
Сонячний вітер				
W1	Щільність протонів (п/см3)	-0,11	0,16	0,41
W2	Насипна швидкість (км/с)	0,34	-0,34	-0,47
W3	Іонна температура (градуси K)	0,14	-0,12	-0,06

2.1.1.4. Лаговий аналіз

Для встановлення лагової залежності було проведено перетворення набору даних. Вихідні поля фіксувалися, після чого часовий ряд кожного вхідного поля зсувався вертикально вниз на кількість рядків, що дорівнює досліджуваному лагові. Після цього розраховувався коефіцієнт кореляції

між вхідним і вихідним полями. Ми досліджували лаг від 0 до 5 годин. Результати розрахунку представлені в табл. 2.2.

Таблиця 2.2

Коефіцієнти кореляції для лагового перетворення набору даних

відставання	E1	E2	P1	P2	P3	P4	P5	I1	I2	W1	W2	W3
температура												
0	-0,11	0,06	0,01	0,14	0,15	0,11	-0,10	-0,57	-0,55	-0,11	0,34	0,14
1	-0,05	0,09	0,06	0,14	0,22	0,16	-0,07	-0,58	-0,56	0,02	0,31	0,21
2	0,01	0,11	0,08	0,13	0,23	0,18	-0,06	-0,59	-0,57	0,13	0,28	0,26
3	0,03	0,14	0,09	0,12	0,21	0,18	-0,06	-0,59	-0,58	0,24	0,24	0,28
4	0,05	0,18	0,11	0,19	0,14	0,13	-0,06	-0,59	-0,58	0,34	0,21	0,30
5	0,08	0,19	0,08	0,22	0,10	0,08	-0,06	-0,59	-0,57	0,41	0,20	0,32
Вологість												
0	0,14	-0,02	-0,03	-0,12	-0,13	-0,08	0,06	0,50	0,48	0,16	-0,34	-0,12
1	0,04	-0,07	-0,08	-0,12	-0,21	-0,14	0,01	0,52	0,50	0,05	-0,32	-0,20
2	-0,01	-0,10	-0,12	-0,10	-0,22	-0,17	-0,01	0,52	0,50	-0,07	-0,28	-0,25
3	-0,05	-0,15	-0,13	-0,11	-0,19	-0,16	-0,02	0,52	0,51	-0,19	-0,23	-0,26
4	-0,08	-0,21	-0,13	-0,19	-0,12	-0,11	-0,03	0,53	0,52	-0,30	-0,19	-0,27
5	-0,07	-0,23	-0,12	-0,22	-0,07	-0,06	-0,04	0,53	0,52	-0,39	-0,16	-0,26
Тиск												
0	0,16	-0,17	0,11	-0,07	-0,17	-0,30	0,22	0,85	0,85	0,41	-0,47	-0,06
1	0,16	-0,17	0,07	-0,04	-0,20	-0,29	0,22	0,86	0,85	0,40	-0,48	-0,08
2	0,12	-0,16	0,08	-0,07	-0,22	-0,29	0,21	0,86	0,85	0,39	-0,51	-0,11
3	0,16	-0,18	0,06	-0,06	-0,24	-0,28	0,18	0,87	0,86	0,36	-0,52	-0,14
4	0,16	-0,20	0,01	-0,09	-0,25	-0,27	0,17	0,88	0,86	0,33	-0,55	-0,19
5	0,13	-0,18	0,01	-0,09	-0,25	-0,26	0,17	0,89	0,86	0,30	-0,56	-0,22

Як видно з таблиці 2.2, найменше R спостерігається для потоку електронів і протонів (E і P). Це означає, що ці поля введення не впливають на поля виводу для всіх лагів. Найбільший R спостерігається для $I1$ і $I2$, що стосуються тиску повітря. Як бачите, R слабко зростає зі збільшенням відставання. Це означає, що між цими полями та вихідними полями існують нелінійні інерційні залежності. Ці затримки означають, що можна зробити прогноз вихідних полів на кілька годин вперед. Подібна ситуація спостерігалася для полів $W1$, $W2$, $W3$.

2.1.1.5. Автокореляційний аналіз

Для подальших досліджень слід провести автокореляційний аналіз, щоб узгодити взаємозв'язок між полями введення. Результати цих розрахунків наведені в таблиці 2.3.

Таблиця 2.3

Коефіцієнти автокореляції для полів введення I1, I2 , W1 , W2 , W3

	I1	***I2***	***W1***	***W2***	***W3***
I1	1,00				
I2	0,98	1,00			
W1	0,26	0,27	1,00		
W2	-0,55	-0,55	-0,29	1,00	
W3	-0,27	-0,28	0,14	0,73	1,00

Результати в таблиці 3 показують сильну лінійну залежність між полями *I1* і *I2* . Це означає, що в розрахунках повинен використовуватися тільки один з них.

2.1.1.6. Пошук кращих моделей

Як видно з лаг-кореляційного та автокореляційного аналізу, найкращі моделі для всіх вихідних полів повинні бути залежними від інтегрального потоку протонів і сонячного вітру з лагом=5:

$$T(H, P) = F((I1 \text{ або } I2)_5, W1_5, W2_5, W3_5), \qquad (2.1)$$

де індекс підписки «5» означає відставання=5.

Ми повинні знати, який з них (*I1* чи *I2*) кращий. Тому ми протестували моделі $T(H,P) = F(I1_5, W1_5, W2_5, W3_5)$ і $T(H,P) = F(I2_5, W1_5, W2_5, W3_5)$, (таблиця 2.5).

Для перевірки цього рішення були протестовані моделі з усіма можливими комбінаціями лагів від 0 до 5 ($6^4 = 1296$ моделей). Моделі з полями введення *I1* або *I2* тестувалися окремо. Теоретичні дослідження [86]

показали, що електрони повинні мати нелінійний вплив на вихідні поля. Тому були проведені аналогічні розрахунки для моделей, що містять одне з полів *E1* або *E2 (6* ^{5} = 7776 моделей) . Крім того, були протестовані моделі, які враховують лише диференціальний потік електронів і протонів (6^7 = 279 936 моделей). Лінійний регресійний аналіз і адаптивна нейронечітка система логічного висновку (ANFIS) використовувалися як моделі в цьому дослідженні. Для кожного поля введення в моделях ANFIS було створено дві функції належності Гаусса:

$$f(x,\sigma,c) = e^{\frac{-(x-c)^2}{2\sigma^2}}, \qquad (2.2)$$

де σі c – отримані під час навчання нейронної мережі.

Оскільки ANFIS є системою типу Sugeno, тип вихідної функції належності перевірявся як постійний. Кожна система ANFIS навчалася протягом 100 епох, початковий розмір кроку – 0,01, швидкість зменшення розміру кроку – 0,9, швидкість збільшення розміру кроку – 1,1. Гібридний метод перевірено як метод оптимізації, що використовується при навчанні параметрів функції належності. Для процесу навчання набір даних був розділений на навчальний і тестовий набори в пропорції 90/10. Цей метод є комбінацією оцінки за методом найменших квадратів і зворотного поширення.

Враховуючи, що досліджено 894 240 моделей і всі вони незалежні одна від одної, для вирішення цієї задачі використовується паралельний розрахунок. Це скоротило час розрахунку приблизно в 3,5 рази. Найдовший розрахунок тривав близько 60 годин. Загальний час складав близько 200 годин (~8 днів). Інструменти, які використовувалися в експериментальних середовищах, перераховані в таблиці 2.4. Результати цих розрахунків представлені в таблиці 2.5.

Таблиця 2.4

Інструменти в експериментальних середовищах

Пункт	**Інструмент**
Операційна система	macOS Sierra 10.12.5
комп'ютер	MacBook Pro (Retina, 15 дюймів, середина 2015 р.)
Процесор	2,5 ГГц Intel Core i7
Оперативна пам'ять	16 ГБ 1600 МГц DDR3
Мова програмування середовища	Matlab R2016a (maci64)
Інструмент статистики	MS Excel для Mac 15.36 (2017)

Таблиця 2.5

Порівняння коефіцієнтів кореляції найкращих моделей з моделями з лагом=5 полів введення

Моделі	Лінійний		ANFIS		Кількість моделей	Номер (2.1) моделі (лінійна/ANFIS)
	Лаг=5	Найкращий	Лаг=5	Найкращий		
температура						
F(I1, W1, W2, W3)	0,8199	0,8251	0,8529	0,8621	1296	19 16
F(I2, W1, W2, W3)	0,8113	0,8178	0,8483	0,8540	1296	11 11
F(I1, W1, W2, W3, E1)	0,8337	0,8431	0,8698	0,8839	7776	54 41
F(I1, W1, W2, W3, E2)	0,8239	0,8287	0,8843	0,9051	7776	65 359
F(E1, E2, P1, P2, P3, P4, P5)	0,3372	0,4241	0,3567	0,4733	279936	*95 927* *196 615*
Вологість						
F(I1, W1, W2, W3)	0,7559	0,7680	0,8026	0,8247	1296	20 24
F(I2, W1, W2, W3)	0,7437	0,7597	0,7953	0,8115	1296	26 24
F(I1, W1, W2, W3, E1)	0,7705	0,7861	0,8216	0,8455	7776	46 123

Продовження таблиці 2.5

Моделі	Лінійний		ANFIS		Кількість моделей	Номер (2.1) моделі (лінійна/ANFIS)
	Лаг=5	Найкращий	Лаг=5	Найкращий		
F(I1, W1, W2, W3, E2)	0,7672	0,7831	0,8538	0,8910	7776	50 733
F(E1, E2, P1, P2, P3, P4, P5)	0,3553	0,4363	0,3689	0,4610	279936	56 329 130 930
Тиск						
F(I1, W1, W2, W3)	0,8985	0,9037	0,9483	0,9637	1296	178 247
F(I2, W1, W2, W3)	0,8767	0,8988	0,9338	0,9506	1296	932 512
F(I1, W1, W2, W3, E1)	0,8994	0,9055	0,9521	0,9679	7776	917 929
F(I1, W1, W2, W3, E2)	0,8991	0,9061	0,9576	0,9673	7776	1280 309
F(E1, E2, P1, P2, P3, P4, P5)	*0,4090*	0,5879	0,4338	0,6153	279936	256 184 276 420

Як видно з таблиці 2.5, критерієм точності був коефіцієнт кореляції між реальними даними та даними моделей. Перш за все, слід зазначити, що всі моделі ANFIS мають вищий коефіцієнт кореляції, ніж лінійні. Чітко видно, що моделі на основі диференціального потоку електронів і протонів мають найменше *R.* Це означає, що вони не є основними факторами впливу на вихідні поля.

2.1.2. ANFIS та нейромережеві моделі для прогнозування лісових пожеж в США

2.1.2.1. Попередній аналіз структури даних

Для перевірки гіпотези використовувались статистичні дані по США. Такий вибір обґрунтовується наявністю великого обсягу статистичних даних про пожежі на відносно великій площі і на щоденній основі. В

дослідженні використовувались дані за період з травня по жовтень 2004 – 2007 років. Дані про лісові пожежі на містяться отримані з [15]. Інформацію про кількість нових невеликих пожеж (F^{small}), а також про нові великі пожежі (F^{large}), було використано як вихідні параметри моделей. Відповідно до цього джерела, існують великі (значні) пожежі: які перевищують 300 акрів трави або 100 акрів лісу. Дані про потік протонів, електронів і сонячного потоку знаходяться на [15]. Дані про швидкість сонячного вітру (км/с), містяться на [15]. В розрахунках використовувались максимальні значення на щоденній основі. Отже, вхідні параметри (показники сонячної активності) були обрані наступним чином:

X_1 — потік протонів > *1 MeB;*

X_2 — потік протонів > *10 MeB;*

X_3 — потік протонів > *100 MeB;*

X_4 — потік електронів > *0,6 MeB;*

X_5 — потік електронів > *2 MeB;*

X_6 — індекс F10.7 (міра рівня шуму, генерованого сонцем на довжині хвилі 10,7 см на орбіті Землі)*;*

X_7 — швидкість сонячного вітру.

Випромінювання Сонця в радіодіапазоні довжин хвиль пов’язане, насамперед, з корональною плазмою в пастці магнітних полів, розміщених в активній області. Це чудовий показник загального рівня сонячної активності. Важливо відзначити, що дані, пов’язані з сонячною активністю, завантажуються з ACE супутника, який знаходиться між Землею і Сонцем. Попередні дослідження показали, що в певних ситуаціях існують невеликі причинно-наслідкові зв'язки між різким припливом протонів і/або електронів і виникненням пожежі на відносно великих площах [87-89]. Враховуючи, що деякі райони можуть бути під впливом обох типів заряджених частинок, або одного з них, (X_6) і (X_7) були обрані в якості

показників сонячної активності. Навчальний період відноситься до останньої фази сонячного циклу 23. У квітні 2008 року сонячна активність була мінімальною, отже в роботі була проаналізована ситуація, що характеризується тривалим спадом сонячної активності.

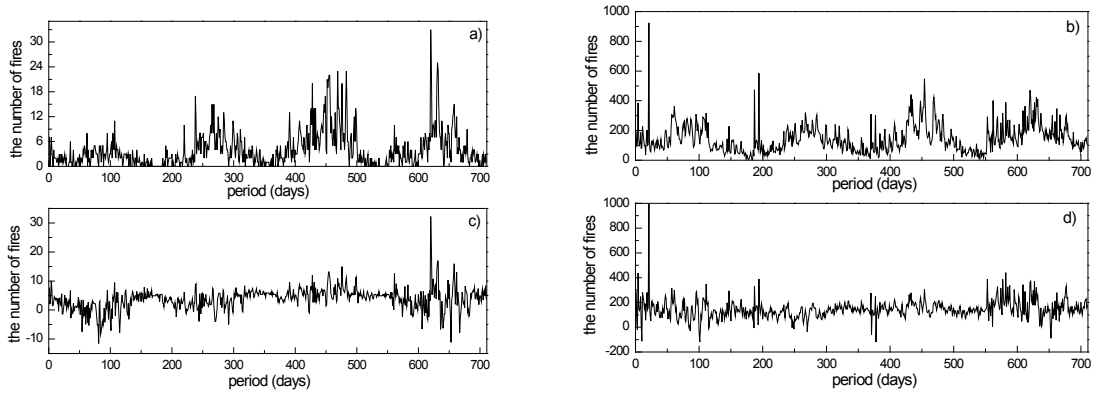


Рис. 2.1. Кількість великих *(a)*, *(c)* і невеликих *(b)*, *(d)* пожеж. Реальні дані *(a)*, *(b)*; дані з усунутою сезонною складовою *(c)*, *(d)*

Як видно з рисунку 2.1 *(a, b)*, на графіках спостерігаються циклічні спалахи пожеж F^{small} і F^{large}. Це пов'язано із сезонним підняттям температури у літні періоди. Крім цього, можна побачити, що протягом всього досліджуваного періоду на фоні сезонних коливань спостерігаються раптові спалахи лісових пожеж. Інтенсивність їх виникнення не залежить від часу. Саме ці спалахи можуть бути пов'язані із сонячною активністю. Тому необхідно спочатку позбутись сезонної компоненти шляхом розкладання часових рядів F^{small} і F^{large} на компоненти з використанням адитивної моделі. Адитивна часова модель у нашому випадку має вигляд [90]: $F^{small(large)} = T^{small(large)} + S^{small(large)} + \widetilde{F}^{small(large)}$, де $T^{small(large)} = \left\{ t_j^{small(large)} \right\}_{j=\overline{1,n}}$ – трендова компонента кількості малих (великих) пожеж; n- кількість спостережень (в нашому випадку $n = 710$ – днів в період з травня по жовтень

2004 – 2007 років); $S^{small(large)} = \left\{ s_j^{small(large)} \right\}_{j=\overline{1,n}}$ – сезонна компонента – кількість малих (великих) пожеж, що пов'язані з підвищенням (зниженням) температури протягом року або з впливом туристів на появу лісових пожеж; $\widetilde{F}^{small(large)} = \left\{ \widetilde{f}_j^{small(large)} \right\}_{j=\overline{1,n}}$ – флуктуаційна компонента, що пов'язана з такими параметрами, як, наприклад, сонячна активність. Видаливши сезонну складову та тренд, ми отримали часові ряди для дослідження впливу сонячної активності на виникнення малих і великих лісових пожеж $\widetilde{F}^{small(large)}$. Для цього був використаний класичний метод індексів сезонності [91]. Методика видалення сезонної та трендової складової полягає в наступному:

Крок 1. Згладжування часових рядів F^{small} і F^{large} за допомогою ковзної середньої.

Крок 2. Розрахунок сезонної компоненти $S^{small(large)}$ наступним чином:

1) знаходження центрованої ковзної середньої. Цей крок необхідний через зміщення отриманих значень середнього арифметичного щодо реальних значень часового ряду
2) розрахунок корегувального коефіцієнта, що передбачає наступне: сума всіх індексів сезонності має бути рівною нулю, таким чином сезонні ефекти для всього річного циклу компенсують один одного в адитивній моделі.

Значення сезонної компоненти, отримані таким чином, представляють відношення кількості пожеж у той чи інший день року до середнього числа пожеж за рік. Таким чином, були отримані як позитивні, так і негативні значення компонент часових рядів.

Крок 3. Видалення сезонної компоненти з вихідних часових рядів. Таким чином, отримано часові ряди числа лісових пожеж без сезонних впливів:

$$\hat{F}^{small(large)} = F^{small(large)} - S^{small(large)} = T^{small(large)} + \widetilde{F}^{small(large)}$$

Крок 4 . Видалення трендової компоненти з $\hat{F}^{small(large)}$ методом найменших квадратів [92]. Таким чином, було отримано часові ряди $\widetilde{F}^{small(large)}$, які були

використали для ідентифікації функціональної залежності між сонячною активністю і появою лісових пожеж.

2.1.2.2. Кореляційний аналіз

Для перевірки гіпотези наявності функціональної залежності між компонентами сонячної активності та спалахами лісових пожеж був проведений кореляційний аналіз між параметрами X_i та кількістю пожеж $\widetilde{F}^{small(large)}$ з урахуванням часу затримки (лагу) між настанням пожеж і сонячною активністю. Результати цього аналізу показані в таблиці 1. Як можна побачити, будь-який коефіцієнт кореляції не перевищує 0,2.

Таблиця 2.6

Коефіцієнти кореляції між вхідними (X*i*, $i=\overline{1,7}$) і вихідними ($\widetilde{F}_L^{small(large)}$) параметрами залежно від лагу $L=\overline{0,5}$

	X_1	X_2	X_3	X_4	X_5	X_6	X_7
$\widetilde{F}_0^{large}$	-0.02	0.01	0.00	0.04	-0.02	-0.15	0.05
$\widetilde{F}_1^{large}$	-0.04	-0.03	-0.01	0.02	-0.04	-0.16	0.04
$\widetilde{F}_2^{large}$	-0.04	-0.02	-0.02	0.00	-0.02	-0.17	0.02
$\widetilde{F}_3^{large}$	-0.04	-0.03	-0.03	-0.01	-0.02	-0.18	0.02
$\widetilde{F}_4^{large}$	-0.05	-0.03	-0.03	-0.01	-0.02	-0.18	0.02
$\widetilde{F}_5^{large}$	-0.02	-0.02	-0.02	0.01	-0.04	-0.19	0.02
$\widetilde{F}_0^{small}$	-0.02	-0.01	-0.01	0.03	-0.02	0.09	-0.04
$\widetilde{F}_1^{small}$	0.01	0.01	-0.01	0.00	-0.02	0.09	-0.03
$\widetilde{F}_2^{small}$	-0.02	0.02	0.01	0.00	-0.01	0.07	-0.03
$\widetilde{F}_3^{small}$	-0.04	-0.02	0.03	0.01	0.02	0.07	-0.02
$\widetilde{F}_4^{small}$	-0.05	-0.04	0.01	0.01	0.04	0.07	-0.07
$\widetilde{F}_5^{small}$	-0.03	-0.03	-0.02	0.00	0.03	0.05	-0.07

Це означає, що немає ніяких лінійних зв'язків між згаданими факторами. Тому необхідно застосовувати методи нелінійного аналізу, щоб перевірити гіпотезу про функціональний взаємозв'язок між виникненням пожеж і сонячною активністю.

2.1.2.3. R/S аналіз

Для визначення ступеня стохастичності часових рядів вхідних і вихідних параметрів був використаний *R/S* аналіз [93-95]. *R/S* аналіз дозволяє встановити факт наявності довгострокової пам'яті у часових рядів. Для цього було використано наступне співвідношення [96]:

$$R/S = c \cdot n^H, \tag{2.3}$$

де *R/S* – нормований розмах, тобто відношення часткових сум відхилень часових рядів від його середнього, масштабований за допомогою стандартного відхилення, *c* – константа, *H* – коефіцієнт Херста.

Це рівняння було розв'язане для кожної із змінних X_i і вихідних часових рядів $\widetilde{F}^{large}$ та $\widetilde{F}^{small}$. В цій роботі наведено приклад аналізу $\widetilde{F}^{large}$. Інші часові ряди аналізуються аналогічно.

Спочатку часовий ряд $\widetilde{F}^{large}$ з довжиною *n* перетворюється в послідовність $F = \{f_j\}_{j=\overline{1,n-1}}$, де $f_j = \ln\left(\frac{\widetilde{f}_j^{large}}{\widetilde{f}_{j-1}^{large}}\right)$. Після цього цей часовий ряд ділиться на A суміжних підперіодів з довжиною l. Кожен півперіод позначений L^a, $a = \overline{1,A}$, кожний елемент підперіоду: $f_{(a-1)l+k}$, $k = \overline{1,l}$. Тоді для кожного підперіоду визначається середнє значення $\overline{f^a} = \frac{1}{l} \cdot \sum_{k=1}^{l} f_{(a-1)l+k}$ і величина накопичених сум:

$$R^a = \max_a\left(\left\{\sum_{k=1}^{l}\left(f_{(a-1)l+k} - \overline{f^a}\right)\right\}\right) - \min_a\left(\left\{\sum_{k=1}^{l}\left(f_{(a-1)l+k} - \overline{f^a}\right)\right\}\right).$$

Стандартне відхилення S^a для кожного підперіоду визначається як:

$$S^a = \sqrt{\frac{1}{l} \cdot \sum_{k=1}^{l} \left(f_{(a-1)l+k} - \overline{f^a} \right)^2} . \quad (2.4)$$

Кожна величина накопичених сум R^a нормалізується шляхом ділення ii на відповідне стандартне відхилення S^a. Тоді середнє значення $(R/S)_l$ для підперіоду довжиною l матиме вигляд:

$$\left(R/S\right)_l = \frac{1}{A} \cdot \sum_{a=1}^{A} \frac{R^a}{S^a} . \quad (2.5)$$

Аналогічні розрахунки проводяться збільшуючи довжини підперіодів з l до $(n-1)/2$. Коефіцієнт Херста (H_l) визначається шляхом розв'язання рівняння лінійної регресії у логарифмічному поданні:

$$\log\left(\left(R/S\right)_l\right) = \log(c) + H_l \cdot \log(l) \quad (2.6)$$

Значення коефіцієнта Херста інтерпретуються наступним чином[97]:

- Якщо $H = 0.5$, часові ряди є стохастичними ("білий шум");
- Якщо $0.5 < H < 1$, часовий ряд характеризується персистентністю, тобто властивістю тривалої пам'яті ("чорний шум").
- Якщо $0 < H < 0.5$, часові ряди є антиперсистентні, тобто часовий ряд змінюється швидше, ніж у випадку випадкового процесу ("рожевий шум").

Використання критеріїв персистентності чи антиперсистентності часових рядів дозволяє прогнозувати розвиток досліджуваного часового ряду у відносно простій формі на базі своєї історії.

На основі коефіцієнту Херста був розрахований інший показник – фрактальна розмірність D:

$$D = 2 - H \tag{2.7}$$

Фрактальна розмірність є кількісною характеристикою, яка характеризує зміну графіка часового ряду залежно від масштабу, тобто ступінь самоподібності. Результати цих обчислень наведені в таблиці 2.7. Як видно з таблиці, середнє значення коефіцієнтів Херста для $X_{1\text{-}5}$ є близьким до 0,5. Це означає, що ці часові ряди описують випадкові процеси.

Таблиця 2.7

Результати *R/S* аналізу для часових рядів

Змінна	X_1	X_2	X_3	X_4	X_5	X_6	X_7	$\widetilde{F}^{small}$	$\widetilde{F}^{large}$
Коефіцієнт Херста	0.58	0.56	0.49	0.56	0.55	0.92	0.69	0.72	0.93
Фрактальна розмірність	1.42	1.44	1.51	1.44	1.45	1.08	1.31	1.28	1.07

На відміну від них, коефіцієнт Херста, що знаходиться у межах 0.69 – 0.72 (для X_7, $\widetilde{F}^{small}$) та 0.92 – 0.93 (для X_6, $\widetilde{F}^{large}$), означає, що для цих часових рядів зміна значень факторів залежить від попередніх періодів. Таке значення коефіцієнту Херста для X_6, X_7, $\widetilde{F}^{small}$, $\widetilde{F}^{large}$ означає, що ці процеси є фракталами і для їх дослідження не може бути використана класична лінійна статистика. Подібність величини фрактальної розмірності (2.7) для X_7 – $\widetilde{F}^{small}$ та X_6 – $\widetilde{F}^{large}$ означає наявність однакових правил у зміні структури графіку часового ряду в залежності від масштабу. Таким чином, можна висунути гіпотезу, що вищезазначені пари часових рядів або строго залежать від однакових «третіх» факторів, або залежні один від одного [98].

В цьому випадку необхідно застосувати методи нелінійної алгебри для встановлення функціональної залежності. Задача пошуку прихованих залежностей у великих базах даних відноситься до задач DataMining. Тому в роботі було досліджено та проведено порівняльний аналіз моделей на основі гібридних нейронних мереж ANFIS та багатошарових нейронних мереж.

2.1.2.4. Формалізація моделей прогнозування лісових пожеж

У загальному випадку, задача зводиться до знаходження залежності у вигляді: $M^{small(large)}: X_1 \times ... \times X_7 \to \widetilde{F}^{small(large)}$. При врахуванні часової затримки (lag), були сформовані дві навчальні множини у вигляді кортежів:

$$Tr^{small} = \left\{ \left\langle \overline{x}_{1,j-L}, ..., \overline{x}_{1,j-L}, \widetilde{f}_j^{small} \right\rangle \right\}_{j=\overline{1,n}} \quad (2.8)$$

$$Tr^{large} = \left\{ \left\langle \overline{x}_{1,j-L}, ..., \overline{x}_{1,j-L}, \widetilde{f}_j^{large} \right\rangle \right\}_{j=\overline{1,n}} \quad (2.9)$$

де L – lag, $\overline{x}_{i,j}$ – нормовані компоненти часових рядів, ($\overline{x}_{i,j} = \frac{x_{i,j} - \min(X_i)}{\max(X_i) - \min(X_i)}$).

Необхідність нормалізації всіх вхідних параметрів зумовлено значною різницею між абсолютними значеннями *max-min* компонентів окремих вхідних векторів, що можуть змінюватися від одного до п’яти порядків (наприклад X_1 і X_6). Також наявною є велика різниця між абсолютними значеннями різних вхідних векторів. Наприклад: $\max(X_4) - \max(X_6) \approx 10^{11}$, $\min(X_4) - \min(X_6) \approx 10^8$ (таблиця 2.8). Комп’ютерний розрахунок без нормалізації цих даних призводить до значних помилок заокруглення, що повністю нівелює адекватність моделі [99-107].

Для визначення часової затримки між спалахами на сонці та настанням пожеж були створені по 6 навчальних вибірок для побудови моделей як для малих лісових пожеж, так і для великих, при $L = \overline{0,5}$ ($\widetilde{f}_j^{small(large)} = M_L^{small(large)}(\overline{x}_{1,j-L}, ..., \overline{x}_{7,j-L})$). Для кожної з вибірок проводилось навчання

окремої нейронної мережі. Після чого проводився аналіз точності для кожного лагу окремо. Це дало змогу визначити найкращу модель та затримку між подіями (в наближенні однакового лагу для всіх вхідних параметрів).

Таблиця 2.8

Статистичні характеристики вхідних та вихідних параметрів

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	$\tilde{F}^{small}$	$\tilde{F}^{large}$
Max	1100000000	74000000	500000	180000000000	9300000000	175	1005	996	32
Min	55000	11000	2100	230000000	650000	65	276	-121	-12
Average	8523106	404424	5487	21438042254	182332930	87	478	144	4
Average of $\overline{X_i}$	0.008	0.005	0.007	0.118	0.020	0.200	0.276	-	-

2.1.2.5. Побудова моделей прогнозування на основі нейронних мереж

В роботі досліджувались два типи багатошарових нейронних мереж:

- Нейронні мережі з прямим поширенням помилки.
- Нейронні мереж із зворотнім поширення помилки.

Для визначення необхідної кількості нейронів були використані емпіричні формули:

$$\frac{mN}{1+log_2 N} \leq L_w \leq m\left(\frac{m}{N}+1\right)(n+m+1)+m \quad (2.10)$$

$$L = \frac{L_w}{n+m} \quad (2.11)$$

$$\frac{N}{10} - n - m \leq L \leq \frac{N}{2} - n - m \quad (2.12)$$

$$2(L+n+m) \leq N \leq 10(L+n+m) \quad (2.13)$$

де N – число елементів навчальної вибірки, m – розмірність вихідного сигналу, n – розмірність вхідного сигналу, L_w – необхідне число синаптичних ваг, L — кількість елементів масиву.

В нашому випадку навчальна вибірка містить 7 входів ($n = 7$) і 1 вихід ($m = 1$). Число елементів навчальної вибірки залежить від лагу, при збільшені якого зменшується кількість елементів навчальної вибірки (табл. 2.9).

Таблиця.2.9

Залежність числа елементів навчальної вибірки від лагу

Лаг	0	1	2	3	4	5
N	710	706	702	698	694	690

У результаті розрахунків було отримано: $63 \leq L \leq 347$. Отже, сумарна кількість нейронів має бути більшою за 63 та меншою за 347. Тому в розрахунках були використані нейронні мережі, що містять 50×50 та 60×60 нейронів в прихованих прошарках для двох типів вищезазначених нейронних мереж (рис.2.2).

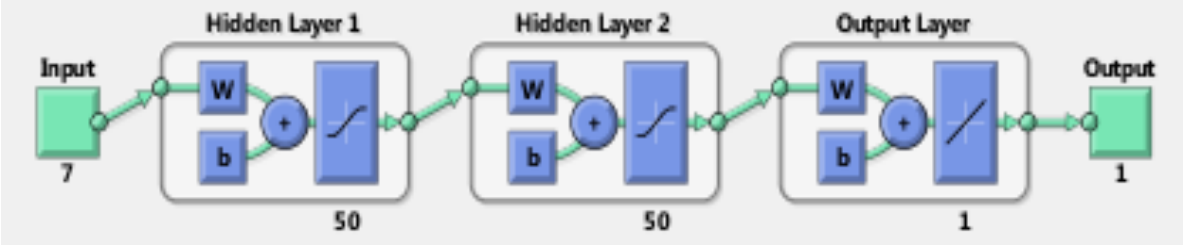


Рис. 2.2. Схема нейронної мережі з 7 входами та 1 входом розмірністю $[50 \times 50]$.

2.1.2.6. Побудова моделей прогнозування на основі гібридних нейронних мереж (ANFIS)

ANFIS — нейронна мережа, що базується на основі нечіткої системи виводу Такагі-Сугено. Її система виведення містить набір нечітких *If-Then* правил, які отримуються при навчанні на великих базах даних на основі

нелінійних функцій [108-109]. Ці методи добре зарекомендували себе в моделюванні складних соціальних систем в наших роботі [110].

Для побудови нечіткої моделі всі вхідні параметри представлені як лінгвістичні змінні. Як було показано вище, в досліджуваній системі наявні нелінійні зв'язки, тому кожний терм у всіх лінгвістичних змінних описується нелінійними Гаусівськими функціями належності. Як показали тестові розрахунки, найкращий результат отримувався при кількості 3-х термів у кожної лінгвістичної змінної для кожного X_i (21 терм для кожної моделі). У випадку 2-х термів моделі не були адекватними. Якщо ж цих термів більше, ніж 3, кількість емпіричних параметрів перевищує обсяг навчальної вибірки, що унеможливлює процес навчання. Як метод виведення нечіткої системи, була обрана функція Сугено нульового порядку. Методом навчання був гібридний спосіб, що об'єднує метод зворотного поширення помилки з методом найменших квадратів. У результаті були отримані продуктивні бази знань, що містять 6561 нечітке правило.

2.1.3. LSTM для прогнозування лісових пожеж в США, Португалії та Греції

2.1.3.1. Аналіз структури даних

Основною задачею дослідження було встановлення функціонального зв'язку між сонячною активністю та кліматичними параметрами повітря, та на основі них провести прогнозування. В якості таких параметрів виступали температура (*T*), вологість (*H*) та атмосферний тиск (*P*) які вимірювались на 5 різних об'єктах. 3 розташовані в Каліфорнії інші дві в Португалії та Греції відповідно. В якості вхідних параметрів виступали: інтегральний потік високоенергетичних протонів сонячного вітру в різних енергетичних діапазонах: >10MeV та >30MeV; диференційні потоки електронів в діапазонах: 38-53 і 175-315 (keV); та протонів: 47-68, 115-195, 310-580, 795-1193 та 1060-

1900 (keV). Також брались до уваги плазми сонячного вітру такі як густина протонів (Proton Density), об'ємна швидкість (Bulk Speed), температура іонів (Ion Temperature), а також радіо потік в діапазоні 10.7 см (10.7 cm Radio Flux).

Наявність великої кількості асинхронних даних, що надходять з різних джерел, та відсутність фізичної моделі процесу дозволяють віднести цю задачу до категорії Big Data. Необхідність врахування часової затримки між сонячною активністю та атмосферними явищами зумовлює використання рекурентних нейронних мереж. Як показали наші попередні дослідження, найкращі результати для таких даних отримуються в рамках ансамблю моделей LSTM.

Класичний алгоритм розрахунку наступний:

- Попередній аналіз вхідних та вихідних даних
- Імпорт та інтеграція даних в розріджену матрицю
- Зведення даних до однакового часового діапазону
- Заповнення пропусків даних
- Зменшення кількості вхідних факторів
- Створення нормалізованих навчальних та тестових вибірок
- Створення та навчання ансамблю рекурентних нейронних мереж типу LSTM
- Перевірка на адекватність та аналіз чутливості ансамблю моделей

Вхідні дані сонячної активності належали до діапазону дат з 16.07.2018 по 16.08.2018. Наявні дані по метеостанціям:

Таблиця 2.10

Діапазони дат метео-спостережень

Станція	Діапазон дат	Прогноз
Каліфорнія	16.07.2018–13.08.2018	3.08.2018–11.08.2018
Португалія	16.07.2018–16.08.2018	3.08.2018–16.08.2018
Греція	16.07.2018–13.08.2018	17.07.2018–26.07.2018

Як видно з таблиці, всі вхідні дані необхідно розбити на навчальну (дати з колонки «Діапазон дат» за виключенням дат колонки «прогноз») та тестову вибірку (колонка «прогноз»). Тестова вибірка розглядалась як прогнозні дані. Як видно з таблиці, для Португалії всі дані розділялись на два блоки: блок навчальної вибірки з 16.07.2018 по 2.08.2018 та блок тестової з 3.08.2018 по16.08.2018. У випадку Каліфорнії та Греції тестова вибірка знаходилась всередині блоку наявних даних. Тому навчальна вибірка для цих країн складалась з двох блоків дат. Каліфорнія: 17.07.2018–2.08.2018 + 12.08.2018–13.08.2018. Греція: 16.07.2018 + 27.07.2018–13.08.2018. Як видно з вищевказаного, тестова вибірка для Каліфорнії знаходилась ближче до кінця наявних дат. У випадку Греції прогнозовані дані розташовуються на початку наявних. Необхідно зазначити, що дані з усіх станцій спостереження вимірювались з різною частотою.

У випадку параметрів сонячної активності, дані представляли собою усереднені значення за певний однаковий інтервал часу. Інтервали теж різнились між собою. У випадку 10.7 cm Radio Flux дані вимірювались в 17, 20 та 23 годинах щодня.

Це все означає різний масштаб дискретизації як вхідних так і вихідних даних та наявність пропущених даних в моменти часу де усереднення проводилось за відсутності даних сенсорів.

Слід зазначити, що час вимірювання вхідних даних фіксувався за універсальним часом по Гринвічу. У випаду Каліфорнії: PDT – Pacific Daylight Time, Португалія: WEST – Western European Summer Time = UTC + 1, Греція: Eastern European Summer Time = UTC + 3. Все це треба враховувати при імпорті даних.

Інформація про дискретизацію наведена в таблиці 2.11. Для зручності дані, які отримані з одного ресурсу групувались у відповідні фрейми.

Таблиця 2.11

Зведені дані вхідних та вихідних параметрів

№ фрейму	**Вхідні фактори**	**Дискретизація**
1	Field1: >10MeV, Field2: >30MeV	5 хвилин
2	Field1: 38-53, Field2: 175-315, Field3: 47-68, Field4: 115-195, Field5: 310-580, Field6: 795-1193, Field7: 1060-1900	5 хвилин
3	Field1: Proton Density, Field2: Bulk Speed, Field3: Ion Temperature	1 хвилина
4	Field1: 10.7 cm Radio Flux	3 виміри в день (17, 20 та 23)
5	Field1: T, Field2: H, Field3: P Каліфорнія: Португалія, Греція:	 15 хв – 1 год 30 хв

Як видно з таблиці, масштаб дискретизації вихідних даних коливається в широкому діапазоні значень від 15 хвилин до 1 години. Дискретизація вхідних параметрів 1-2 та 3 фреймів складає відповідно 5 та 1 хвилину. Окремо вимірюється фрейм 4.

Часові ряди даних вхідних та вихідних параметрів представлені на рисунках 2.3-2.12.

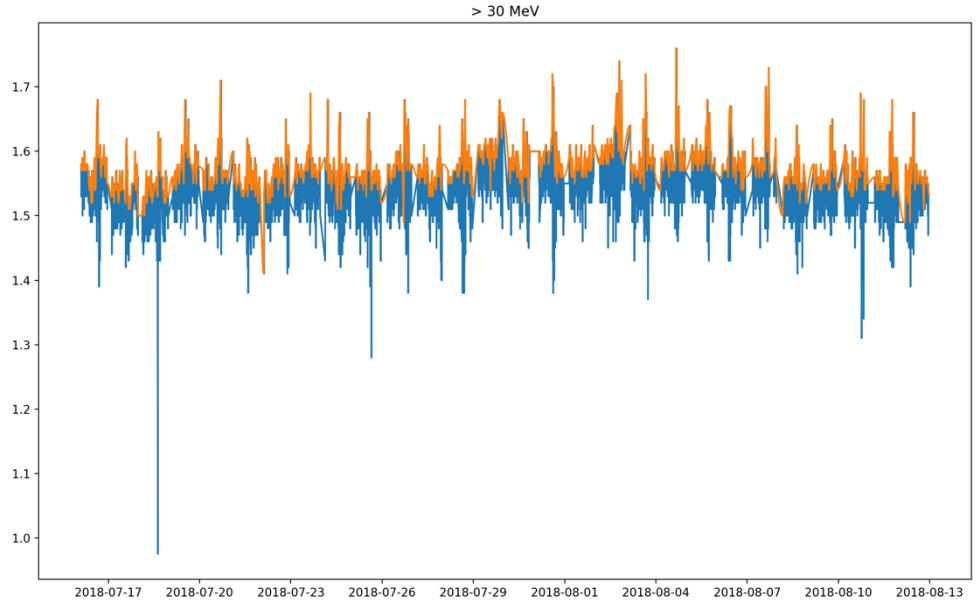


Рис.2.3. Часові ряди вхідних параметрів >10MeV, >30MeV (фрейм 1)

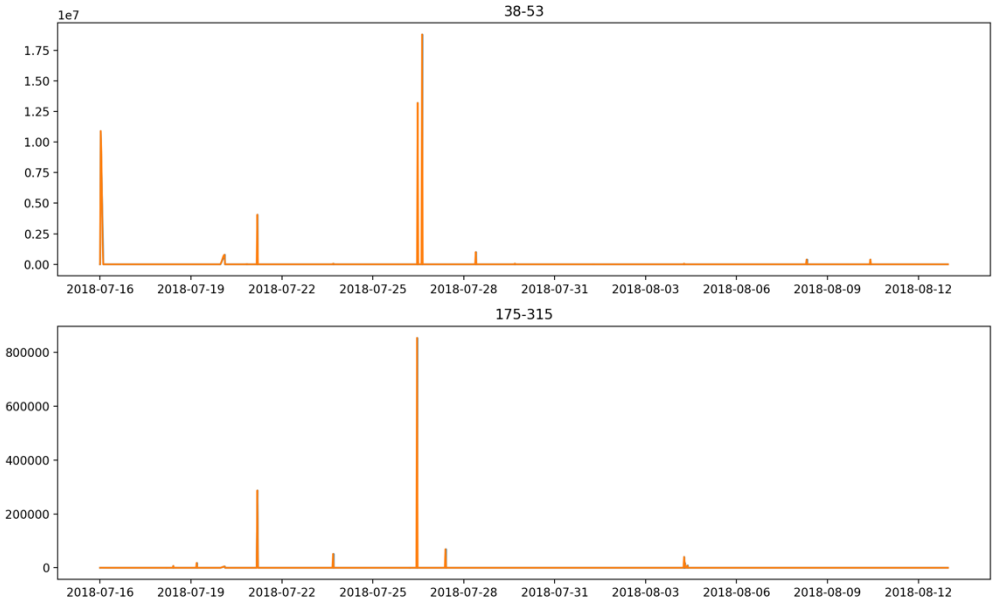


Рис.2.4. Часові ряди вхідних параметрів 38-53,175-315 (фрейм 2)

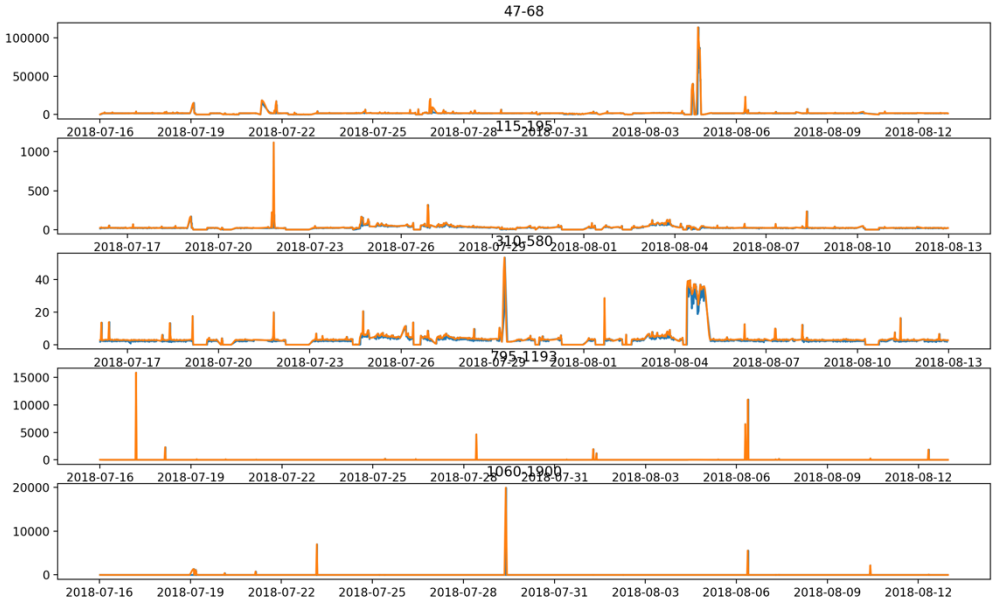


Рис.2.5. Часові ряди вхідних параметрів 47-68, 115-195, 310-580, 795-1193, 1060-1900 (фрейм 2)

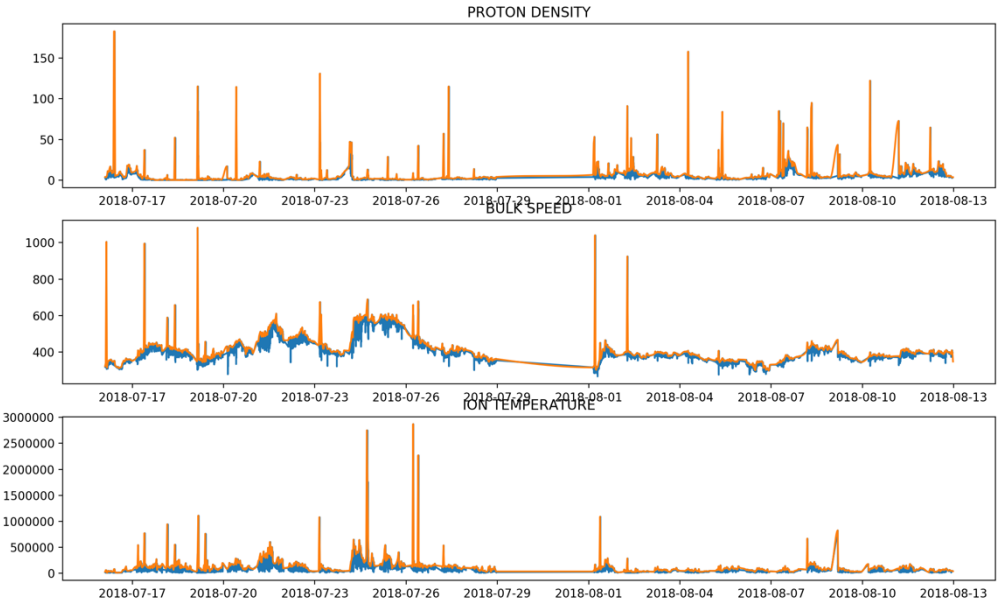


Рис.2.6. Часові ряди вхідних параметрів Proton Density, Bulk Speed, Ion Temperature (фрейм 3)

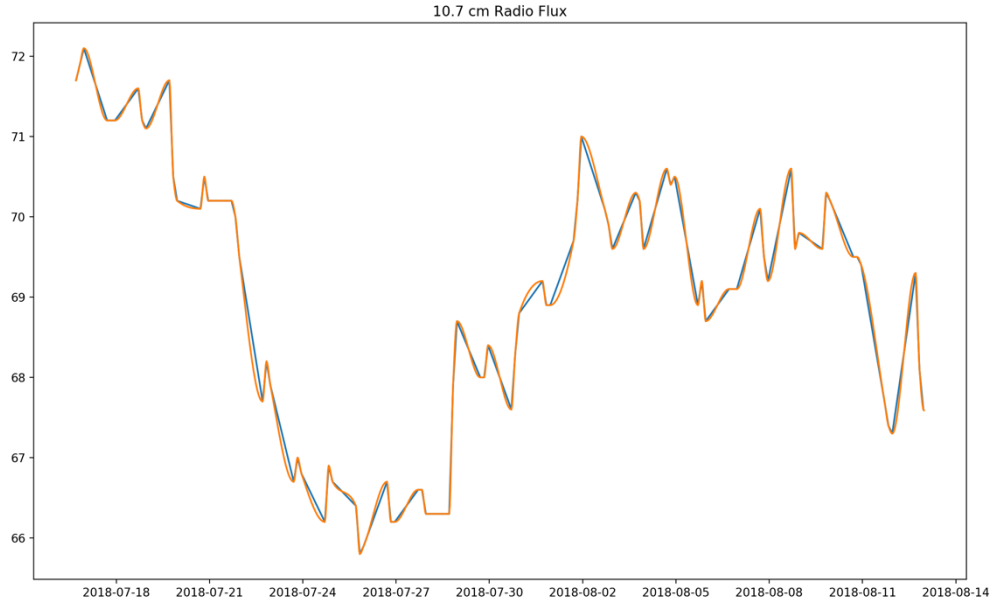


Рис.2.7. Часові ряди вхідних параметрів 10.7 cm Radio Flux (фрейм 4)

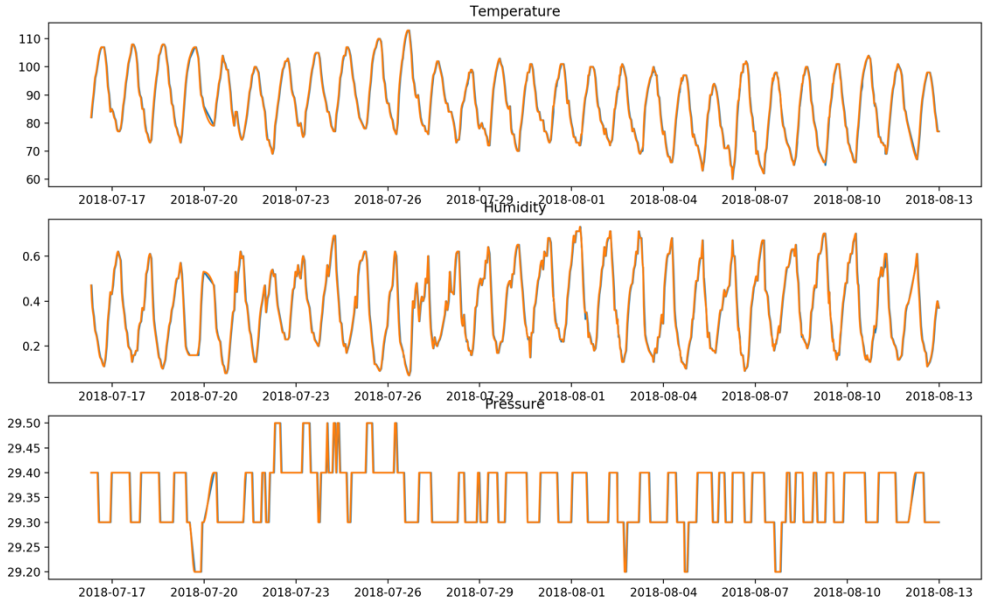


Рис.2.8. Часові ряди вихідних параметрів T, H, P (фрейм 5) Каліфорнія1

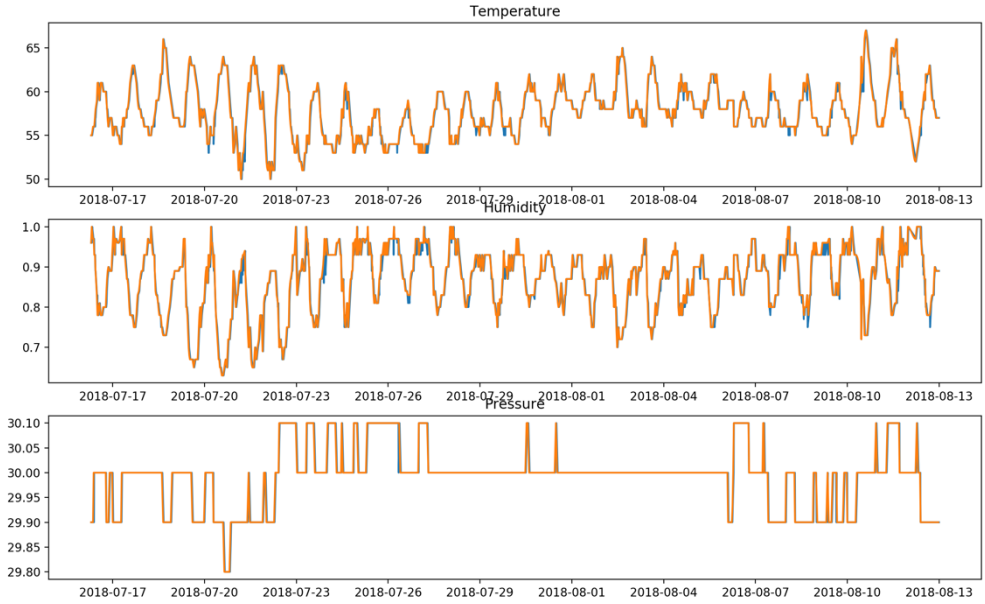


Рис.2.9. Часові ряди вихідних параметрів T, H, P (фрейм 5) Каліфорнія2

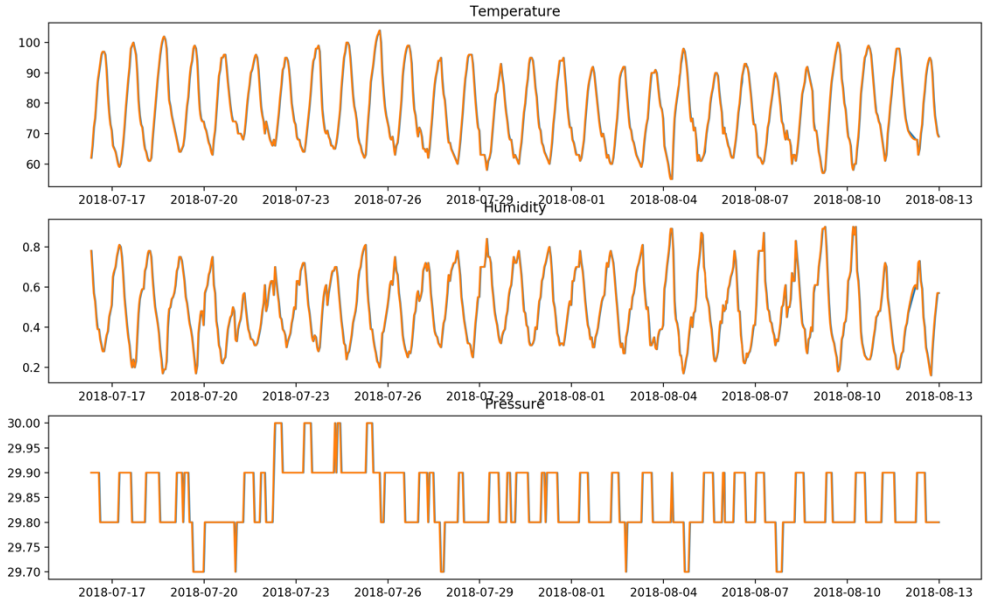


Рис.2.10. Часові ряди вихідних параметрів T, H, P (фрейм 5) Каліфорнія3

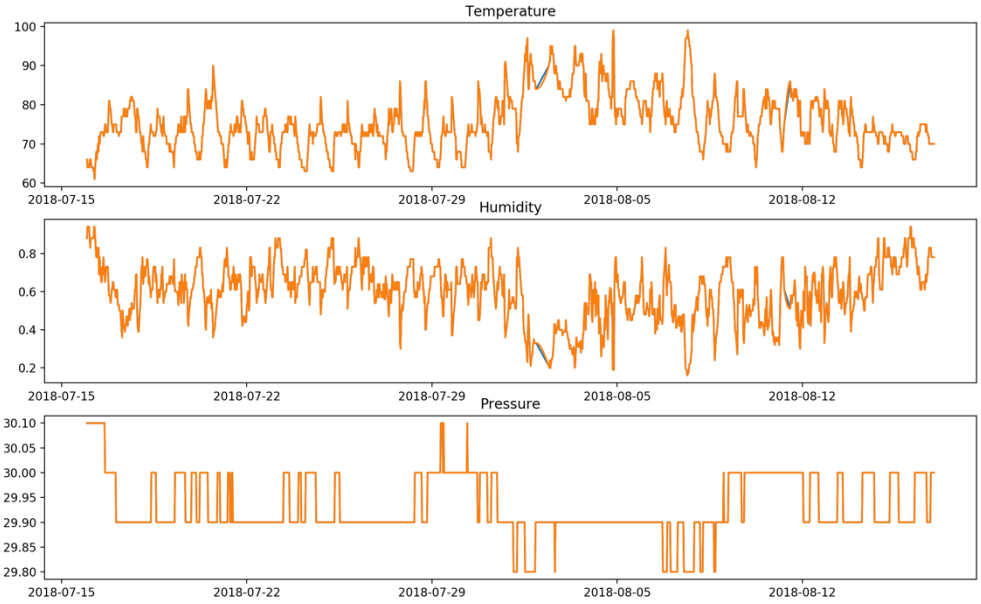


Рис.2.11. Часові ряди вихідних параметрів T, H, P (фрейм 5) Португалія

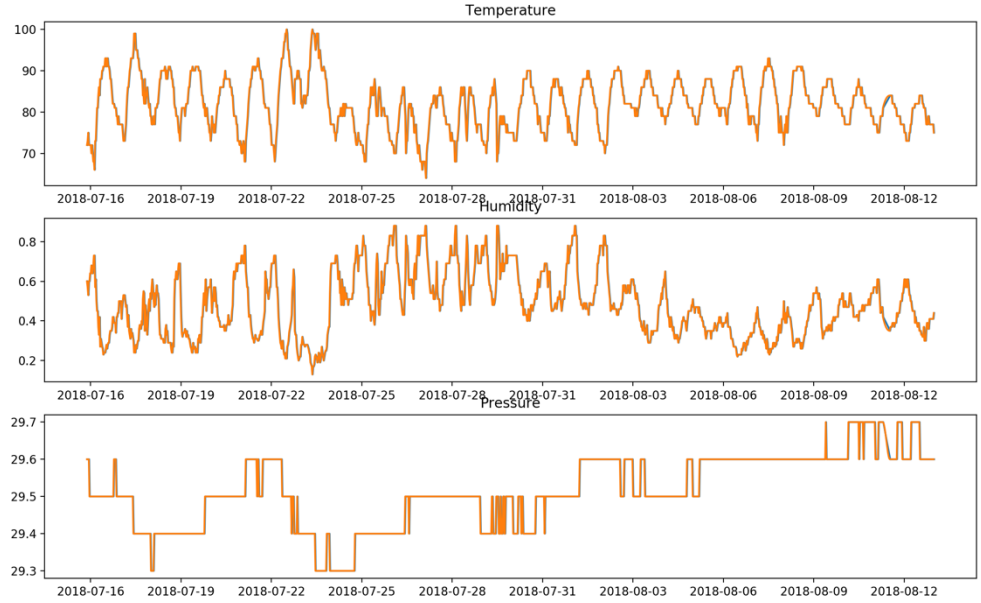


Рис.2.12. Часові ряди вихідних параметрів T, H, P (фрейм 5) Греція

Як видно з рисунків 2.3-2.6 часові ряди мають багато пропущених даних та інколи спостерігаються аномальні коливання в бік малих та великих значень. Крок дискретизації 10.7 cm Radio Flux взагалі є дуже великим та потребує інтерполяції на менший часовий діапазон.

2.1.3.2. Імпорт та інтеграція даних в одну таблицю

Як видно з таблиці 2.11 дані, отримані з різних джерел можуть бути представлені як структури DataFrame. Відмінністю цієї структури від часового ряду є наявність кортежу що складається з декількох полів даних, які виміряні в однаковий момент часу. І відповідно одного спільного індексного поля дата/час:

$$DF_f = (Key = Date\&Time: Data =< Field_{f1}, \dots, Filed_{fn} >) \quad (2.14)$$

де f – номер фрейму з таблиці 1, $Field_{fi}$ – і-поле фрейму f.

Як видно з таблиці 2.11 в цьому випадку є 4 DataFrame вхідних полів та 5 окремих DataFrame для кожної станції.

Наступним кроком є об'єднання всіх DF_f в за індексними полями в одну DataFrame. В результаті було отримано 5 окремих наборів даних відповідно для кожної станції стпостереження:

$$DF^s = DF_1 \cup DF_2 \cup DF_3 \cup DF_4 \cup DF_5^s \quad (2.15)$$

де s індекс станції спостереження

Отримані DataFrame (DF^s) містять множину всіх можливих значень індексних полів зі всіх DF_f, в якості полів даних виступають всі досліджувані вхідні та вихідні поля. Значення полів даних для яких відсутні дані в певний момент часу заповнюються порожніми значеннями. В

результаті кортежі значеннь полів DF^sпредставляють собою розріджені матриці. А індексне поле $Date\&Time$ має нерівномірну в часі дискретизацію.

2.1.3.3. Зведення даних до однакового часового діапазону

Для отримання придатної для навчання навчальної вибірки необхідно трансформувати DF^s шляхом введення однакового кроку часової дискретизації всіх полів. Як видно з таблиці 2.11, вихідні поля мають дискретизацію від 15 хв до 1 години залежно від станції. Дискретизація ж вхідних полів є суттєво меншою. Збільшення кроку дискретизації до однієї години призведе до втрати інформації про флуктуації коливань вхідних полів і суттєвого зменшення навчальної вибірки. Зменшення ж кроку дискретизації до 30 хв вимагатиме інтерполяції даних для станцій в Каліфорнія1 та Каліфорнія3. А отже можливого накопичення помилкових даних. Однак, як видно з рисунків 5 та 7, динаміка зміни показників по цим станціям є достатньо гладкою, що дає змогу провести додаткову дискретизацію без втрати точності. Тому в якості дискретизації було вибрано 30хв діапазон.

При зведенні дискретизації до вищезазначеного діапазону у випадку вхідних полів фреймів 1-3 (таблиця 2.11) дані які попадали в 30 хв діапазон об'єднувались операцією max:

$$DF^s \xrightarrow[\max(\Delta Key=t)]{} DF^{s,t} \qquad (2.16)$$

де t=30 хв.

Це дало змогу з одного боку врахувати спалахи сонячної активності які спостерігались в цих діапазонах (рис 2.3-2.6) відсіяти аномальні від'ємні коливання. В результаті загальна кількість записів зменшилось із 47.000 до 1120 (Каліфорнія та Греція) та 1310 (Португалія).

2.1.3.3. Заповнення пропусків даних

В результаті попередньої трансформації даних вдалось усунути переважну кількість пропущених даних. Однак залишились проблеми пропущених даних на початку та кінці часових рядів, а також інтерполяції для поля 10.7 cm Radio Flux. Для їх усунення для кожного поля була використана сплайнова кубічна інтерполяція [111]. Для зменшення білого шуму було використано ковзне вікно за 4 точками [112]. Результати інтерполяції приведені на рис 2.3-2.12:

$$DF^{s,t} \xrightarrow[spline^3, rolling\ window]{} DF^{s,t,inp} \tag{2.17}$$

де t=30 хв.

Як видно з рисунків 2.3-2.6 *max-cubic-spline* інтерполяція дозволила відсікти мінімуми та підкреслити найбільші спалахи параметрів сонячної активності. Для часового ряду 10.7 cm Radio Flux вдалось визначити проміжні значення цього фактору (рис.2.7.). Як видно з рисунку динаміка зміни цього показника є досить гладкою, тому сплайнова інтерполяція показала непогані результати. Як видно з рисунку 2.8 інтерполяція даних для Каліфорнія 1 та 3 повністю повторює оригінальні часові ряди, що дає право стверджувати про відсутність помилкових даних при інтерполяції.

2.1.3.4. Зменшення кількості вхідних факторів

Для зменшення кількості вхідних параметрів був проведений автокореляційний аналіз між всіма полями $DF^{s,t,inp}$. Як видно з таблиці 2.10, досліджуваний часовий діапазон для станції в Каліфорнії та Греції співпадають. Для Португалії часовий діапазон є дещо більшим. Тому результати автокореляції будуть дещо відрізнятись.

Як показали розрахунки, всі коефіцієнти кореляції є суттєво малими за виключенням >10MeV та >30MeV, що становить 0.93. Крім цього високі від'ємні коефіцієнти кореляції присутні між вихідними полями Температури та Вологістю. Це означає обернений зв'язок між цими полями, а саме при збільшенні температури вологість зменшується.

Отже можна прийти до висновку, що одним з вхідних полів можна знехтувати. Для розрахунків було обрано >30MeV так як він враховує саме високоенергетичні протони. В результаті були отримані DataFrame ($DF^{s,res}$) що шляхом вилучення поля $> 10MeV$ з $DF^{s,t,inp}$:

$$DF^{s,res} = DF^{s,t,inp} \backslash s, DF^{t,inp}_{>10MeV} \quad (2.18)$$

Отримані DataFrame містять 12 вхідних полів та 3 вихідні.

2.1.3.5. Створення нормалізованих навчальних та тестових вибірок

Низькі коефіцієнти кореляції свідчать про наявність нелінійних функціональних залежностей, для встановлення яких необхідно нормалізувати всі вхідні та вихідні данні ($DF^{s,res} \rightarrow \widetilde{DF}^{s,res}$). Це дозволить зменшити помилку заокруглення при навчанні нейронних мереж.

Наступною особливістю даної задачі є те, що вплив вхідних факторів на вихідні може відбуватись з певною затримкою в часі t_L (L – лаг). Окрім того вихідні фактори представляють собою складну систему, яка залежить від інших факторів, які не враховані в цій задачі. Щоб врахувати їх вплив, до вхідних факторів необхідно додати значення вихідних полів за попередній проміжок часу t_L. Як показали наші попередні розрахунки цей час може становити до 4 діб= $24 \cdot 4$ годин. Тобто, враховуючи крок дискретизації 30хв (двічі на годину), це призведе до збільшення вхідних факторів з 15 (12 вхідних +3 вихідні) до $N = 15 \cdot 4 \cdot 24 \cdot 2 = 2880$. Це

унеможливлює розв'язання задачі класичними нейронними мережами та іншими регресійними моделями. Для подолання цієї проблеми можна скористатись або повним перебором всіх можливих моделей зі всіма можливими комбінаціями та кількістю вхідних факторів, або скористатись рекурентними нейронними мережами LSTM, які були розроблені спеціально до такого класу задач [113].

Для використання цього типу нейронної мережі кожен з отриманих $\widetilde{DF}^{s,res}$ необхідно спочатку трансформувати до вигляду:

$$\widetilde{DF}^{s,res}_{LSTM} = (Key = t: Data =< \widetilde{In}_1(t-1), \dots, \widetilde{In}_{12}(t-1), \tilde{T}_1(t-1), .., \tilde{T}_3(t-1), \dots, \widetilde{In}_1(t-t_L), \dots, \widetilde{In}_{12}(t-t_L), \tilde{T}_1(t-t_L), .., \tilde{T}_3(t-t_L), \tilde{T}_1(t), .., \tilde{T}_3(t) >) \quad (2.19)$$

де t – дата і час ключового поля, $\widetilde{In}$ та $\tilde{T}$ – нормалізовані вхідні та вихідні поля відповідно, L=$4 \cdot 24 \cdot 2 = 192$ – максимальний досліджуваний лаг.

Наступним кроком є розділення $\widetilde{DF}^{s,res}_{LSTM}$ на вихідні та вхідні дані. У якості вихідних виступали останні три поля кортежу $Data$, всі інші – вхідні:

$$\widetilde{DF}^{s,res}_{LSTM,tar} = (Key = t: Data^s_{tar} =< \tilde{T}_1(t), .., \tilde{T}_3(t) >) \quad (2.20)$$

$$\widetilde{DF}^{s,res}_{LSTM,in} = (Key = t: Data^s_{in} =< \widetilde{In}_1(t-1), \dots, \widetilde{In}_{12}(t-1), \tilde{T}_1(t-1), .., \tilde{T}_3(t-1), \dots, \widetilde{In}_1(t-t_L), \dots, \widetilde{In}_{12}(t-t_L), \tilde{T}_1(t-t_L), .., \tilde{T}_3(t-t_L) >) \quad (2.21)$$

Далі кортеж значень вхідних полів трансформувався в тривимірну форму вигляду:

$$\widetilde{DF}^{s,3D}_{LSTM,in} = (Key = t: Data^{s,3D}_{in} = \{d_{t,l,f}\}_{t=\overline{1,r_s},l=\overline{1,192},f=\overline{1,15}} >) \quad (2.22)$$

де d – вхідне поле кортежу $Data^{s,}_{in}$, а індекси: t – часу (кількість записів), l – лагу , f – вхідних полів, r_s–кількість записів $\widetilde{DF}^{s,res}_{LSTM}$ відповідної станції s.

Фактично на вхід нейронної мережі необхідно подати двовимірний масив. Кожна колонка масиву представляє собою часовий ряд вхідних факторів та вихідних величин за попередній лаг часу t_L. Вихід складається з 3-х полів (рис.2.13).

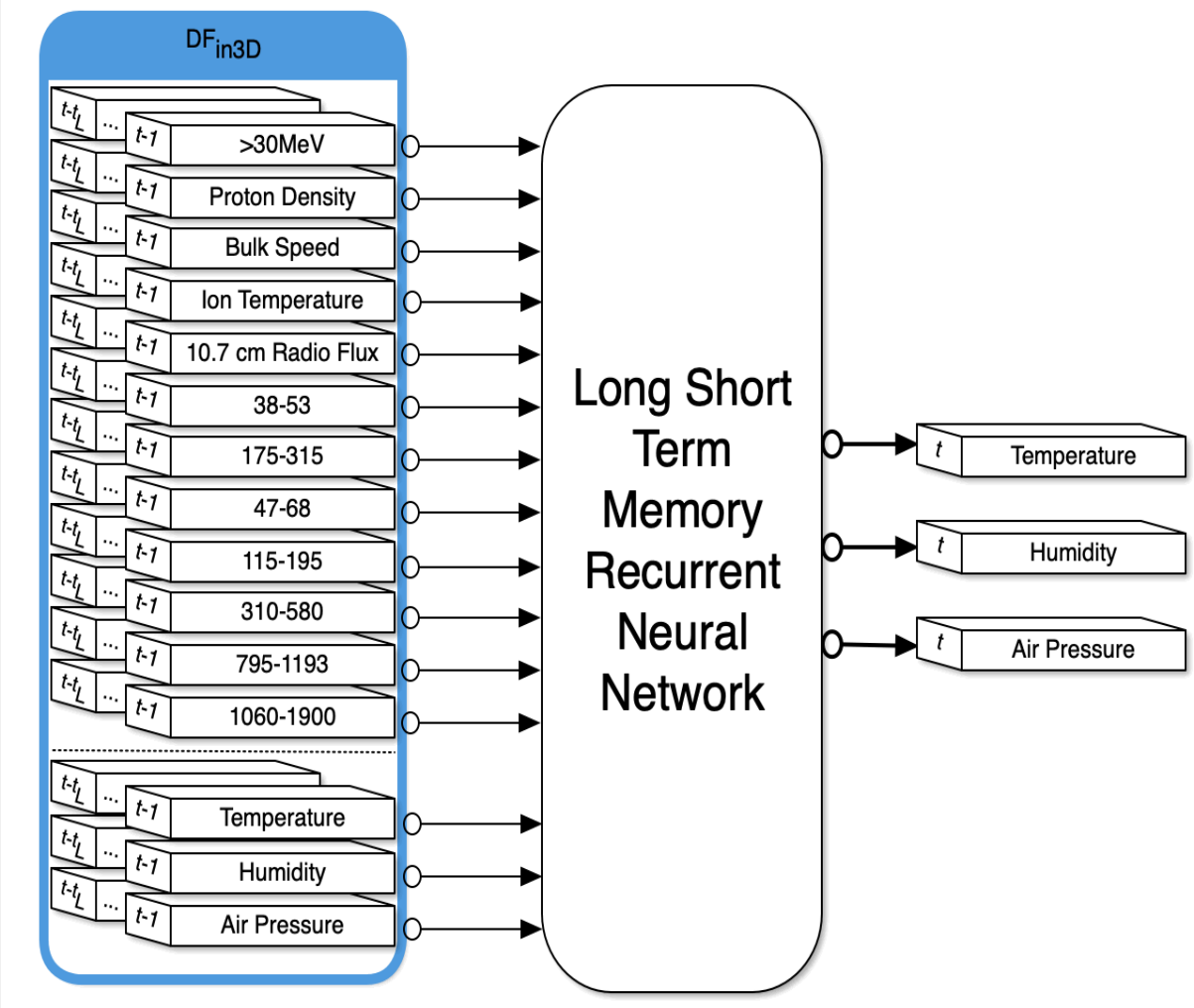


Рис.2.13. Структура входів-виходів Рекурентної нейронної мережі

Для подальшого навчання та тестування нейронної мережі дані кожної навчальної вибірки для окремо взятої станції s ($\widetilde{DF}^{s,3D}_{LSTM,in}$ та $\widetilde{DF}^{s,res}_{LSTM,tar}$) були розділені на навчальні та тестові відповідно до значень дат вказаних в таблиці 2.10.

2.1.3.6. Створення та навчання рекурентних нейронних мереж типу LSTM

Як було зазначено вище, в якості типу рекурентної нейронної мережі була вибрана рекурентна нейронна мережа з довгою короткочасною пам'яттю (LSTM). Ця нейронна мережа дозволяє моделювати поведінку системи, яка залежить від часу. Це реалізується шляхом зворотної передачі вихідного сигналу нейронної мережі в момент часу t-1 назад на вхід одного з мережевих шарів в момент часу t. В загальному структура такої мережі виглядає так:

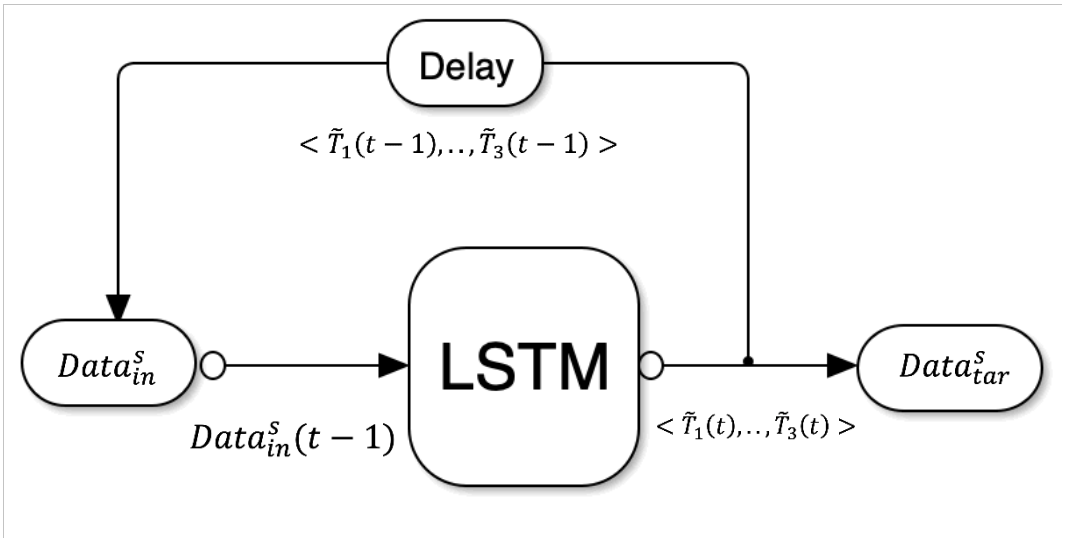


Рис.2.14. Загальна структура рекурентної нейронної мережі LSTM

Як показали тестові розрахунки, найкращі результати отримувались для LSTM наступної конфігурації: кількість входів – 15, виходів – 3, кількість нейронів 50, розмір блоку навчання – 10% (навчання відбувалось поблоково на даних навчальної вибірки), епохи навчання визначались динамічно в процесі навчання з метою уникнення перенавчання, критерій точності – середньоквадратична похибка, метод навчання – Adam [114]. Структура нейронної мережі, отримана за допомогою TensorBoard бібліотеки TensorFlow представлена на рис 2.15.

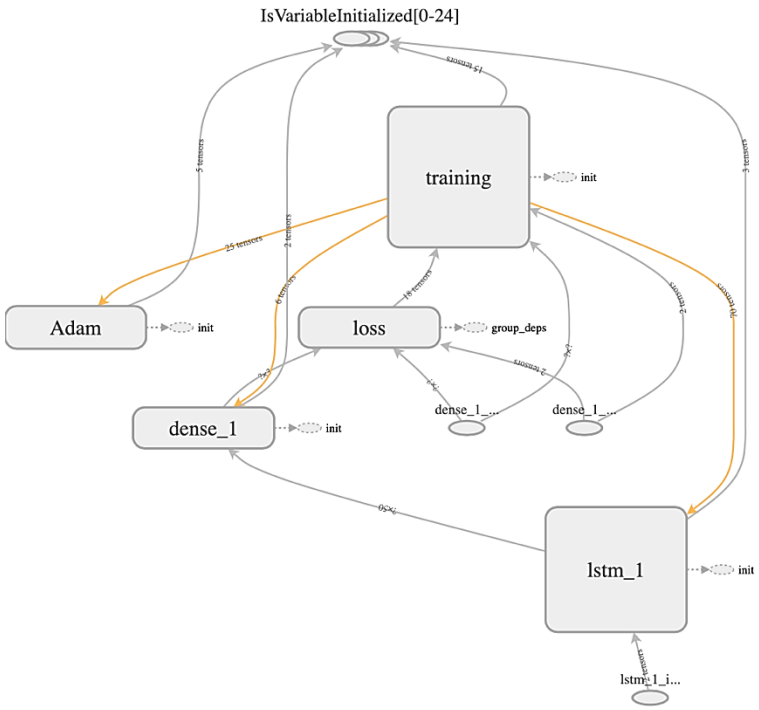


Рис. 2.15. Структура рекурентної нейронної мережі LSTM

2.2. Урагани

2.2.1. Аналіз структури даних

Задача полягала в знаходженні функціональних залежностей між характеристиками сонячної активності та основними характеристиками ураганів. В якості досліджуваних ураганів виступили: IRMA, JOSE та KATIA. Основні характеристики наборів даних наведені в таблиці 2.12. В якості основних досліджуваних критеріїв виступили швидкість вітру та тиску. Як видно з таблиці дані по кожному з ураганів поновлювались кожні 6 годин. Кожен запис представляє собою усереднений показник за зазначений часовий інтервал дискретизації. Динаміка вищезазначених характеристик приведена на рис.2.16.

Таблиця 2.12

Основні характеристики досліджуваних ураганів

	IRMA	JOSE	KATIA
Початок (mm/dd/hh)	08/30/12	09/05/12	09/05/18
Кінець (mm/dd/hh)	09/12/00	09/21/18	09/09/20
Максимум	06/09/2017	09/09/11	09/08/18
Тривалість	13 днів	16 днів	4 дні
Дискретизація	30 хвилин	6 годин	6 годин
Кількість записів	52	66	15

Як видно з рисунку 2.16. кожен з ураганів має яскраво виражений пік на графіку швидкості вітру. Кожному з цих піків, згідно закону Бернуллі, відповідає мінімум тиску в епіцентрі. Також видно, що ці урагани досягли свого максимуму з різницею 1-3 дні. Отже можна припустити, що вони спричинені одними і тими самими глобальними чинниками. До таких чинників можна віднести сонячну активність. Яка, як відомо, може спричиняти глобальні явища із певною часовою (лаговою) затримкою.

2.2.2. Попередня обробка вхідних даних

До характеристик сонячної активності які були протестовані в роботі відносяться потоки протонів та електронів різної енергії, комплексний показник Сонячної активності – Radio Flux 10.7 та характеристики сонячного вітру – швидкість та густина. Основні характеристики набору вхідних даних приведені в таблиці 2.13.

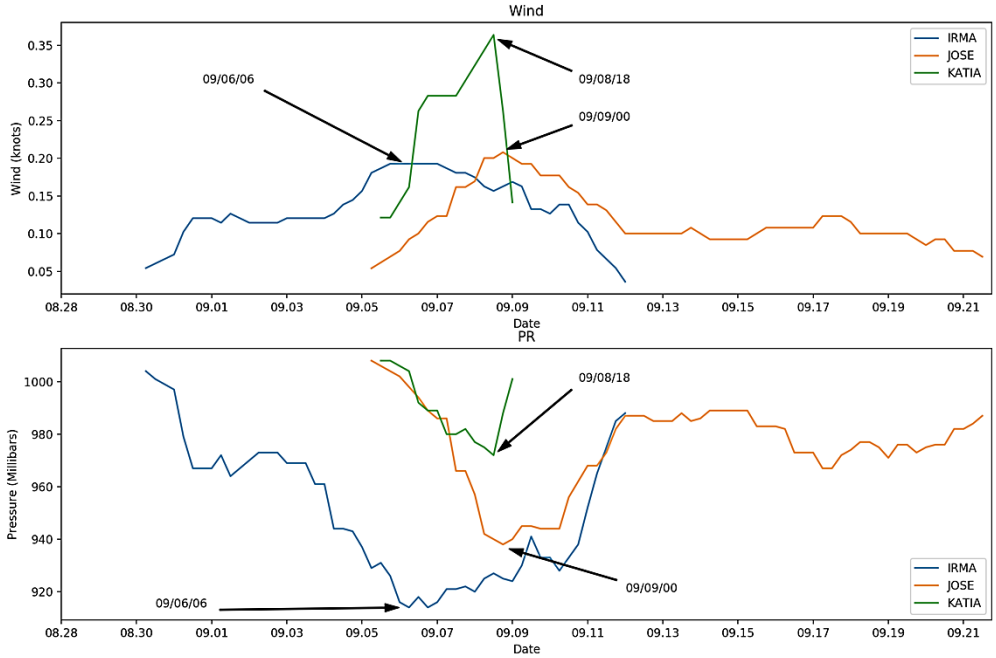


Рис. 2.16. Швидкість вітру та тиск ураганів IRMA, JOSE та KATIA.

Таблиця 2.13

Характеристики набору даних сонячної активності

Характеристики сонячної активності	Одиниці виміру	Початок (mm.dd.hh)	Кінець (mm.dd.hh)	Дискретизація
P > 1, P > 5, P >10, P >30, P >50, P>100	Протонів (> MeV)/ (см^2*с)	28.08.00	22.09.00	5 хв
E>0.8, E>2.0	Електронів (> MeV)/ (см^2*с)	28.08.00	22.09.00	5 хв
Radio Flux 10.7		28.08.00	21.09.00	1 день
Speed	км/с	28.08.00	22.09.00	1 година
Wind	Протонів/см^3	28.08.00	22.09.00	1 година

Як видно з таблиці, діапазон даних вхідних параметрів є більшим за вихідних, що дає змогу врахувати лагові залежності без зменшення розміру часових рядів. Слід зазначити, що дискретизація вимірювання величин в усіх випадках крім Radio Flux 10.7 є більшою за досліджувані вихідні величини (табл.2.12). Для подальшого дослідження дискретизація всіх вхідних даних була зведена до 6 годин шляхом усереднення:

$$\overline{In}(t_i) = \sum_{j=i-b}^{i-1} In(t_j)/b \qquad (2.23)$$

де In – часовий ряд вхідного параметру, b – кількість усереднених даних, j – момент (індекс запису) часу в часовому ряді.

Усереднення проводилось з врахуванням того, що величина на певний момент часу $In(t_j)$ є усередненим значенням за весь попередній період між вимірюванням не включаючи вимірювання в даний момент часу. У випадку часових рядів електронів та протонів величина блоків усереднених даних складала $b = 72$, для Speed та Wind $b = 6$. Це усереднення дало змогу усунути проблему пропущених даних, які інколи зустрічались в часових рядах.

Засобами Python формула (2.23) реалізовувалась за допомогою використання функцій mean та isnan бібліотеки NumPy. На першому кроці вилучався зріз набору вхідних даних DS з поля l із необхідним блоком даних для усереднення (2.24). Наступною командою усереднювались всі непорожні елементи отриманого зрізу (2.25)

```
dt_na=DS[i-1:i+b,l]                                   (2.24)
value= numpy.mean(dt_na[~numpy.isnan(dt_na)])         (2.25)
```

У випадку часового ряду Radio Flux, дискретизація якого є більшою за вихідні поля, дані були інтерпольовані з дискретизацією 6 годин. В якості методу була використана кубічна сплайнова інтерполяція за допомогою полінома Ерміта (PCHIP). В розрахунках була використана функція PchipInterpolator бібліотеки scipy.interpolate. Результати інтерполяції представлені на рис.2.17.

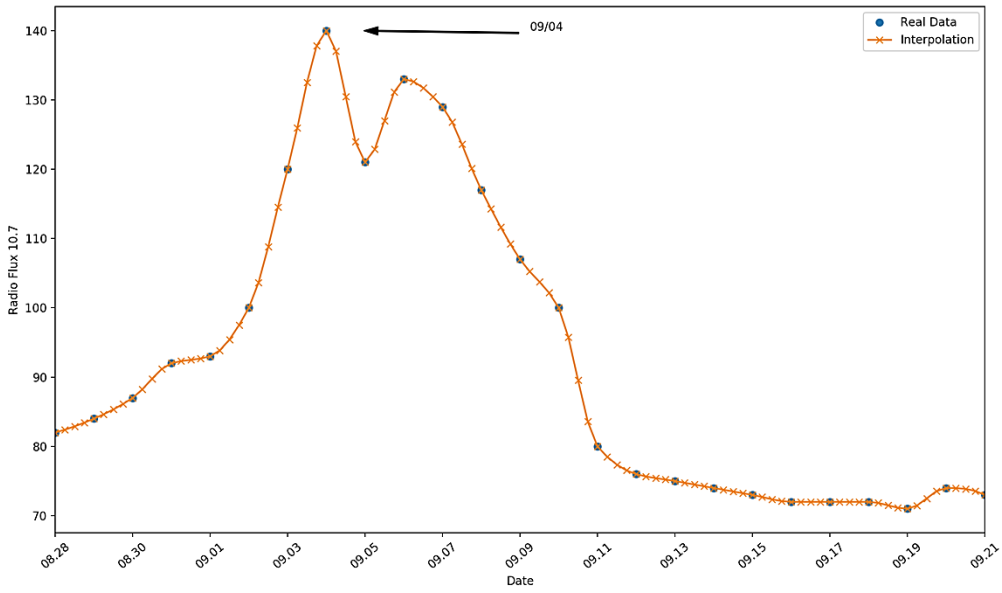


Рис.2.17. Інтерполяція Radio Flux 10.7 на дискретизацію 6 годин

Як видно з рисунку, на графіку відсутні осциляції та присутній яскраво виражений екстремум в точці 04 вересня. Що випереджає піки досліджуваних ураганів відповідно від 2 до 5 днів. Отже в разі встановлення взаємозв'язку між цим фактором та вихідними полями можлива лагова затримка становитиме від 8 до 20 шестигодинних інтервалів.

2.2.3. Кореляційний аналіз

Для встановлення взаємозв'язку між вхідними факторами був проведений автокореляційний аналіз (табл. 2.14.)

Таблиця 2.14

Автокореляційний аналіз вхідних факторів

	P > 1	P > 5	P >10	P >30	P >50	P>100	E>0.8	E>2.0	SPEED	DENSITY	Radio Flux 10.7
P > 1	1.00										
P > 5	0.77	1.00									
P >10	0.66	0.97	1.00								
P >30	0.56	0.91	0.98	1.00							
P >50	0.52	0.87	0.95	0.99	1.00						
P>100	0.45	0.78	0.87	0.94	0.98	1.00					
E>0.8	-0.19	-0.12	-0.05	0.01	0.03	0.06	1.00				
E>2.0	-0.20	-0.17	-0.12	-0.09	-0.07	-0.06	0.81	1.00			
SPEED	0.26	0.11	0.09	0.07	0.07	0.07	0.13	0.02	1.00		
DENSITY	0.27	0.13	0.09	0.07	0.06	0.04	-0.38	-0.19	0.00	1.00	
Radio Flux 10.7	0.12	-0.04	-0.15	-0.20	-0.19	-0.16	-0.07	-0.22	-0.11	-0.10	1.00

Як видно з таблиці, між часовими рядами протонів існує сильний креляційний зв’язок. Така сама ситуація і для потоку електронів. Отже кількість вхідних факторів можна суттєво зменшити. Для вибору найбільш адекватного чинника часові були нормалізовані та зображені на одному графіку (функція normalize бібліотеки sklearn.preprocessing) (рис.2.16).

Як видно з рисунку, всі нормалізовані фактори, які описують потік протонів характеризуються однаковою динамікою. А саме, в усіх 6 часових рядах присутні два яскраво виражені піки. Перший пік припадає на дату 07 вересня, другий на 11 вересня. Тобто перший настав пізніше за екстремум урагану Irma і дещо випереджає урагани Jose і Katia. Другий пік з’являється після того як всі три урагани зменшують свою силу. Отже є малоймовірним припущення, що потік протонів впливає на настання ураганів.

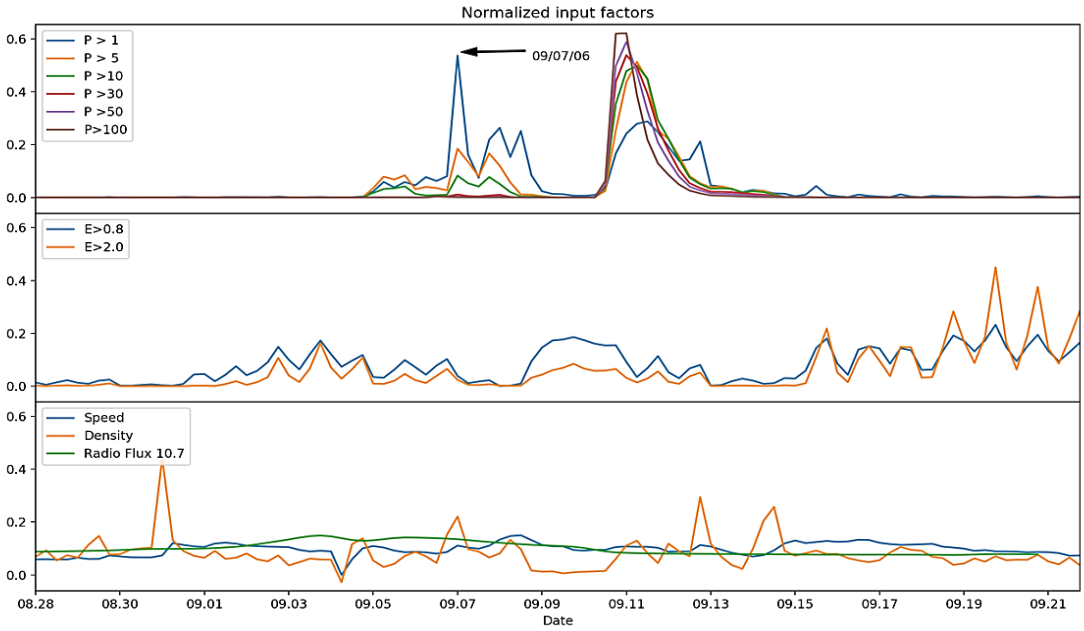


Рис.2.18. Нормалізовані значення вхідних параметрів

Як видно з другого графіку, поведінка потоків електронів різної енергії є досить схожою між собою. На графіках присутні яскраво виражені осциляції, які візуально не спостерігаються на динаміці досліджуваних характеристик ураганів. Поведінка сонячного вітру (третій графік) теж суттєво відрізняється від рис.2.16. Отже є малоймовірною вплив цих факторів на силу вітру та тиск повітря ураганів.

Для підтвердження чи спростування цих висновків був проведений лаговий кореляційний аналіз, що дав змогу встановити кореляцію між окремими часовими рядами вхідних факторів зміщених на певну кількість рядів по вертикалі вниз (лагу) та вихідними факторами. Програмна реалізація лагової трансформації здійснювалась шляхом розрахунку реверсивних зрізів часових рядів target=target[lag:], input=input[:-lag]. Коефіцієнт кореляції визначався як другий елемент матриці кореляції corrcoef бібліотеки NumPy: numpy.corrcoef(input, target)[0][1]. Досліджувався лаг в діапазоні від 0 до 20 (5 днів). Результати аналізу максимальних та мінімальних значень коефіцієнта кореляції в залежності від лагу представлені в таблиці 2.15.

Таблиця 2.15

Зведений кореляційний лаговий аналіз вхідних факторів

	P > 1	P > 5	P >10	P >30	P >50	P>100	E>0.8	E>2.0	Speed	Density	Radio Flux 10.7
	IRMA Wind										
Max	0.21	0.37	0.33	0.20	0.20	0.18	0.73	0.54	0.39	0.07	0.86
Lag	6	6	7	7	7	9	14	13	20	7	6
	IRMA PR										
Min	-0.38	-0.46	-0.43	-0.33	-0.33	-0.22	-0.81	-0.61	-0.51	0.03	-0.91
Lag	7	6	7	7	7	9	14	14	20	7	9
	JOSE Wind										
Max	0.44	0.13	0.12	0.13	0.14	0.16	0.18	0.21	0.45	0.15	0.72
Lag	7	0	0	0	0	0	20	20	3	12	18
	JOSE PR										
Min	-0.37	-0.02	-0.02	-0.07	-0.09	-0.11	-0.33	-0.25	-0.53	-0.24	-0.47
Lag	7	10	0	0	0	0	0	20	4	12	19
	KATIA Wind										
Max	0.55	0.57	0.65	0.68	0.62	0.74	0.68	0.61	0.66	0.51	0.84
Lag	0	9	11	2	12	11	19	19	0	1	17
	KATIA PR										
Min	-0.58	-0.70	-0.68	-0.76	-0.65	-0.77	-0.76	-0.68	-0.71	-0.53	-0.91
Lag	1	9	11	2	2	11	19	19	0	1	17

Як можна бачити з таблиці, найвищий коефіцієнт кореляції R спостерігається для фактору Radio Flux 10.7 для урагану Irma: $R_{wind}=0.86$ (лаг 6), $R_{pr}=-0.91$ – тиск (лаг 9). Далі Katia (лаг 17) $R_{wind}=0.84$ і $R_{pr}=-0.91$. Останній за кореляцією ураган Jose (лаг 18) $R_{wind}=0.72$ і $R_{pr}=-0.47$. Від'ємні коефіцієнти кореляції підтверджують обернений зв'язок між вхідним фактором та вихідним. Це підтверджує попередні висновки стосовно наявності взаємозв'язку між цим фактором та параметрами ураганів.

Як видно з таблиці, інші фактори теж мають достатньо високі коефіцієнти кореляції. Зокрема потік електронів та характеристики сонячного вітру. Це спростовує попереднє твердження про відсутність їх впливу на урагани. Потік протонів має високий коефіцієнт кореляції тільки для урагану Katia. Це може бути пояснено малою кількістю записів в часових рядах даних для цього урагану (табл 2.12.).

Крім того розкид лагів з максимальними (мінімальними) коефіцієнтами кореляції є суттєвим для різних вхідних величин. Отже високі коефіцієнти кореляції та невизначеність з лаговими затримками зумовлює необхідність подальшого дослідження для встановлення функціональних залежностей між зазначеними вхідними та вихідними полями.

Тому в якості тестових полів для подальшого дослідження були вибраний Radio Flux 10.7. А також потік протонів та електронів з максимальною енергією (P>100 та E>2.0) і параметри сонячного вітру Speed та Density.

2.2.4. Паралельні розрахунки для пошуку оптимальних моделей

Для зручної формалізації моделей об'єднаємо вихідні часові ряди у вектор цілей, а вхідні у вектор параметрів:

$$T = (T_1, T_2, T_3, T_4, T_5, T_6) \quad (2.26)$$

$$X = (X_1, X_2, X_3, X_4, X_5) \quad (2.27)$$

де T_i – часові ряди IRMA Wind, IRMA pressure, JOSE Wind, JOSE pressure, KATIA Wind, KATIA pressure відповідно, X_i – часові ряди P>100, E>2.0, Speed, Density, Radio Flux 10.7 відповідно.

Задача полягає в знаходженні для кожного T_i найбільш точної та адекватної функціональної залежності типу:

$$T_i = F_i(X, L_i, \Omega_i) \tag{2.28}$$

де $L_i = \{l_{ij}\}_{j=1,5}$ – вектор оптимальних лагів, Ω_i – параметри моделі.

У якості критеріїв оптимізації розглядались коефіцієнт детермінації R^2 та середньоквадратична помилка. Як показали подальші розрахунки в більшості випадків моделі з максимальним коефіцієнтом детермінації та мінімальною середньоквадратичною похибкою співпадали. У випадку відмінності цих моделей більш адекватною виявились моделі з максимальним значенням коефіцієнта детермінації. Тому в подальшому він використовувався як критерій точності.

Для перевірки точності моделей в ході оптимізації використовувалась крос-валідація по k блокам (K-fold cross-validation). Згідно якої навчальна вибірка розділялась на k однакових за розміром блоків. Кожен з блоків почергово виступає в якості тестової вибірки, а інші $k-1$ блоків в якості навчальної. Результат точності визначається за розрахунком значення коефіцієнту детермінації між складовими вектору цілей T_i та даними моделей на значеннях тестових вибірок $F_i^{cv}(X, L_i, \Omega_i)$. Для цього була використана функція cross_val_predict бібліотеки sklearn.model_selection. В ході розрахунків величина тестової вибірки вибиралась в розмірі 10% від загального розміру навчальної вибірки, тобто $k = 10$. В загальному задача оптимізації виглядає наступним чином:

$$R_i^2(T_i, F_i^{cv}(X, L_i, \Omega_i)) \xrightarrow{yields} max \tag{2.29}$$

Змінні рішення: $L_i \in Tasks, \Omega_i$

$$l_{ij} < 22;$$

$$\max_{j=1,5}\{l_{ij}\} < (lag - 2)_{Lag=\overline{0-22}}$$

де Ω_i – параметри моделі, які визначаються шляхом підгонки вихідних даних моделей до вектору цілей, метод підгонки залежить від типу моделі (лініцна, нейронна мережа, тощо); оптимізація відбувалась шляхом повного перебору всіх можливих комбінацій вектору лагів L_i для кожної компоненти з $X_i = \{x_{ij}\}_{j=1,5}$ від 0 до 22. Величина максимально лагу вибиралась з попереднього аналізу таблиці 2.15, де максимальний лаг становив 20. Тому було прийнято рішення перевірити на 2 лаги більше. В такому випадку множина комбінацій лагів визначається як Декартовий добуток векторів тестових лагів для кожного вхідного параметру і становить 23^5=6 436 343:

$$Tasks(22) = \prod_{j=1,5} L(22) \tag{2.30}$$

де $L(22) = \{0,1, \dots ,22\}$

Реалізація Декартового добутку засобами Python проводилась наступним чином:

```
lag_list=[list(range(lag+1))]*len(X)          (2.31)
task_lag=list(itertools.product(*lag_list))   (2.32)
```

де lag_list – список, елементами якого є списки лагів L, які необхідно протестувати. На основі цього списку формується список задач task_lag за допомогою функції product бібліотеки itertools. Для розгортання списку lag_list в список аргументів функції використовується оператор “*”. Кожен елемент task_lag представляє кортеж $\left(l_{ij}\right)_{j=1,5}$ довжиною 5 елементів, що містить значення лагів отриманих шляхом Декартового добутку списків списку lag_list.

Враховуючи, що для крос-валідації необхідно оптимізувати 10 моделей + одна додаткова на повній множині значень навчальної вибірки, а також що таку оптимізацію треба провести для кожного елементу вектору цілей яких

є 6, загальна кількість моделей, які потребують оптимізації складе: 23^5*11*6 = 424 798 638! Така величезна кількість задач вимагає оптимального вибору як виду типу моделі (2.29) так і алгоритмів оптимізації.

Для зменшення кількості тестованих лагів був запропонований алгоритм пошуку оптимальної моделі:

1. Визначається максимальна кількість лагів $lag = 2$
2. Формується множина задач (2.30):
    a. Для першого проходу $Tasks(lag)$
    b. Для наступних проходів, з метою уникнення повторів задач, розраховується різниця множин задач між попереднім кроком $Tasks(lag)' = Tasks(lag) \backslash Tasks(lag - 1)$
3. Знаходиться оптимальна модель згідно (2.29)
4. Якщо максимальне значення лагу для будь-якої компоненти оптимальної моделі не перевищує $lag - 2$ вважається, що оптимальне значення знайдене і алгоритм завершується
5. Якщо $lag = 22$ алгоритм завершується і вважається, що оптимального значення не має
6. Збільшення $lag += 1$ та перехід до 1 кроку.

Наявність множини незалежних задач, що використовують одну і ту саму область пам’яті дозволяють використати паралельні розрахунки шляхом формування пулу для мультипроцесного розрахунку задач (2.29). Для цього використані функції Pool бібліотеки multiprocessing. З пулу почергово вибиралась задача з конкретним кортежем лагів та відправлялась на розрахунок до вільного ядра процесора “воркера (worker)”. Для формування списку результатів, отриманих на різних ядрах процесора, використовувалась функція dict бібліотеки multiprocessing.Manager. Після розрахунку всіх задач знаходилась функція з максимальним коефіцієнтом детермінації та перевірялась умова 4 та 5 алгоритму.

У якості тестових були вибрані лінійні моделі. Це дозволило суттєво скоротити час розрахунку та визначити оптимальні лаги для кожного з вхідних параметрів для всіх векторів цілей. Для цього були використані функції LinearRegression та fit з бібліотеки sklearn.

2.2.5. Уточнення моделей за допомогою штучних нейронних мереж

В результаті попереднього аналізу отримуються шість лінійних моделей $T_i = F_i\left(X, L_i, \Omega_i^{Lin}\right)$ (2.28). Тобто відомими є параметри лінійних моделей Ω_i та вектор лагів вхідних параметрів L_i. Визначені лаги були основою для уточнення моделей за допомогою нейронних мереж. Як видно з таблиці 2.12 максимальний розмір навчальних вибірок становить 66, а мінімальна 15 записів. Кожна нейрона мережа має мати 5 входів. Такий невеликий розмір навчальної вибірки ставить певні обмеження на розмір нейронних мереж та можливість їх адекватного навчання. В якості нейронних мереж були вибрані багатошарові перцептрони (MLP) зворотного поширення. Методом навчання слугував квазі-Ньютоновський метод оптимізації. В якості активаційної була вибрана логістична функція. Нейроні мережі були створені за допомогою функції MLPRegressor бібліотеки sklearn.neural_network. Навчання та крос-валідація проводилось універсальними функціями fit та cross_val_predict тієї ж бібліотеки.

Як показали попередні розрахунки найкращі результати спостерігались для одношарових нейронних мереж з кількістю нейронів прихованому прошарку рівному 7. Зменшення кількості нейронів прихованого прошарку погіршувало здатність мережі до навчання. А збільшення – навпаки призводило до перенавчання. Тобто результати навчання суттєво покращились, а на результатах тестування методом крос-валідації спостерігались суттєві коливання. Однак навіть в найкращих випадках у вихідних результатах спостерігались 1-2 аномальні флуктуації. Які зникали при повторному навчанні в одному місці часового ряду і з’являлись в

іншому. Для вилучення цих флуктуацій був використаний метод Делфі для експертних оцінок. Він полягав в наступному:

1. Для кожної моделі $T_i = F_i\left(X, L_i, \Omega_i^{ANN}\right)$ (2.28) створювались та навчались декілька нейронних мереж. В нашому випадку оптимальна їх кількість становила 9. Їх збільшення не призводило до покращення результату.
2. Для кожної з мереж розраховувались прогнозні значення на тестових наборах даних методом крос-валідації. В результаті отримувалась матриця типу:

$$Res_i = \begin{bmatrix} f_{i1}^1 & \cdots & f_{im}^1 \\ \vdots & \ddots & \vdots \\ f_{i1}^9 & \cdots & f_{im}^9 \end{bmatrix} \quad (2.33)$$

   де m – розмір навчальної вибірки для конкретного вектору цілей, верхній індекс – порядковий номер нейронної мережі
3. Кожна колонка сортувалась, після чого з неї вилучались 10% записів з мінімальними та максимальними значеннями.
4. Для залишених значень по кожній з колонок визначалась медіана, що і слугувала результатом

Засобами Python третій крок алгоритму реалізовується однією командою:

Res =numpy.sort(Res, axis=0)[int(len(Res)*0.1):- int(len(Res)*0.1), :] (2.34)

де Res – матриця (2.33) формату NumPy, функція sort сортує значення по зазначеній параметром axis осі, вираз в квадратних дужках реалізує зріз даних по горизонтальній осі матриці.

Формування результуючого часового ряду реалізовувалось за допомогою генератора списку на транспонованій матриці Res (функція transpose бібліотеки nunpy.matrix):

Res=nunpy.matrix.transpose(Res) (2.35)

Res= [statistics.median_grouped(i) for i in Res] (2.36)

де функція median_grouped бібліотеки statistics розраховує медіану часових рядів з перцентилем 50%.

У результаті цього етапу, на відміну від лінійних моделей, для кожного з векторів цілей отримувався список з 9 навчених нейронних мереж: $T_i = \left\{F_i^n\left(X, L_i, \Omega_{i,n}^{ANN}\right)\right\}_{n=\overline{1-9}}$. Враховуючи, що для кожної крос-валідації будувались та навчались 10 нейронних мереж + 1 мережа на повній навчальній вибірці (необхідно для подальшого аналізу чутливості), загальна кількість нейронних мереж склала: 6*9*11=594. Для зменшення комп'ютерного часу теж були задіяні паралельні розрахунки, що використовувались і на рівні крос-валідації.

2.2.6. Прогнозування з використанням рекурентних нейронних мереж

Одним із недоліків вищезазначених підходів є те, що вони не враховують кумулятивний ефект поведінки вхідних полів протягом досліджуваного лага. Тобто вони враховують лише одне значення вхідних факторів, зміщене на певний часовий інтервал. Однак може виникнути ситуація, коли поведінка лише одного фактора протягом певного періоду часу може призвести до кризової події. Як зазначалося вище, для цього не можна використовувати лінійні моделі та нейронні мережі зворотного поширення, оскільки кількість вхідних параметрів значно збільшиться. Рішенням цієї ситуації є використання рекурентних нейронних мереж [115]. Реалізовано з'єднання зворотного зв'язку, в яких вихідний сигнал подається на вхідний рівень як додаткові входи, щоб врахувати часову поведінку полів у системі. Таким чином, вихідний сигнал залежить від попереднього стану системи. Ітеративно прокручуючи сигнал, можна розглянути поведінку системи за певну кількість відстаючих кроків.

Крім того, слід зазначити, що вихідні фактори залежать не лише від згаданих вище 5 вхідних факторів. Це складна нелінійна система, яка залежить від багатьох факторів, не врахованих у завданні. До вхідних факторів необхідно додати значення вихідного фактора в попередні моменти часу, щоб врахувати їх комплексний вплив. Для цього набір даних потрібно перетворити в тривимірну форму наступним чином:

$$X_L^i = (X_1(t-1), \dots, X_5(t-1), T_i(t-1), X_1(t-t_L), \dots, X_5(t-t_L), T_i(t-t_L) >) \quad (2.37)$$

$$X_{3D,L}^i = \left(X_1(t-l)_{l=\overline{1,L}}, \dots, X_5(t-l)_{l=\overline{1,L}}, T_i(t-l)_{l=\overline{1,L}}\right) \quad (2.38)$$

де t– індекс рядка, L– максимальне значення лагу, i – цільовий індекс (2.27).

Бібліотека Pandas була використана для реалізації цієї трансформації. Усі матричні структури перетворено на тип даних DataFrame з полем індексу – Дата й час. Потім метод Shift використовувався для зсуву лагів полів введення та виведення. 3D трансформація DataFrame була реалізована методом Reshape.

Поля виведення залишаються без змін. Отже, ми бачимо, що кількість полів введення збільшилася на 1 за рахунок додавання поля виводу. Кожен вхід являє собою масив значень поля для попереднього часу відставання L . Така трансформація DataSet дозволяє усунути розмірні проблеми та врахувати поведінку вхідних параметрів за попередній період часу. З іншого боку, необхідно враховувати попередні значення цільового фактора. Це означає, що кількість рядків у DataSet буде зменшена на кількість досліджуваних лагів L . Це, з обмеженим розміром DataSet , значно зменшує розмір досліджуваного лага L. Тому врахувати lag =22 для рекурентних нейронних мереж LSTM в даному випадку неможливо. Для дослідження обрано інтервал часу L = 4 години.

Рекурентна нейронна мережа с довго короткочасна пам'ять (LSTM) було обрано як модель дослідження. Ця нейронна мережа дозволяє моделювати поведінку системи, яка залежить від затримки часу. Це усвідомлює зворотний передача вихідного сигналу нейронної мережі в момент часу *t-1* повернутися до входу одного з мережевих рівнів. Цей складний вхід використовується для обчислення виходу за час t.

LSTM це тип рекурентна нейронна мережа, що дозволяє запам'ятовувати значення протягом тривалого або короткого періоду часу. Ця мережа не має використовувати ан функції активації в межах його повторюваних компонентів. Таким чином, збережене значення ні зникнути ітеративно з часом.

Бібліотека Keras і TensorFlow (бібліотека Google) були використані для побудови, підгонки та прогнозування нейронної мережі. LSTM представляє собою нейронну мережу, тому кінцеве значення було розраховано аналогічно класичній нейронній мережі методом Delphi.

2.2.7. Встановлення взаємозв'язку між піками

Як було встановлено вище, піки на прогнозних значеннях моделей (2.26) та (2.28) приблизно співпадають. Тому наступним етапом перевірки було встановлення взаємозв'язку між піками на графіку Bulb Speed та графіку швидкості урагану. Для цього спочатку необхідно визначити час настання всіх піків. Для цього часові ряди трьох ураганів та сонячного вітру бінаризуються:

$$\acute{bs}_i \to bs_i^b$$

$$v_i \to v_i^b$$

де $bs_i^b = 0$ або $v_i^b = 0$ – відповідає відсутність піку, 1 – пік.

Для усунення з розрахунку випадкових незначних флуктуацій, мінімальною шириною піку було вибрано 20 лагів (5 годин). Тобто, якщо на інтервалі 5 годин було декілька коливань – вибирався один максимальний екстремум. Крім того, в разі наявності залежності між цими явищами, кількість піків на графіку сонячного вітру та швидкості ураганів має приблизно співпадати. Саме при цих значеннях ширини кількість піків складала: сонячний вітер – 13, IRMA – 10, Jose – 12. Та 1 для урагану Katia.

Результати локалізації встановлених піків представлено на рисунку 2.19. Як видно з рисунків, ураган Katia має лише один пік, зумовлений малою кількістю даних. Для інших ураганів кількість піків є близькою до кількості піків на графіку сонячного вітру.

Для встановлення взаємозалежності між піками, навчальна вибірка була трансформована аналогічно (2.28):

$$v^b(t) = F\left(b\acute{s}^b(t), .., b\acute{s}^b(t-l)\right) \tag{2.39}$$

У результаті такої трансформації загальна кількість записів складала 539.

Так як вихідне значення представляє собою бінарне поле, то задача (2.39) представляє собою класичну задачу класифікації. Для перевірки гіпотези в якості навчальної вибірки було обрано дані по урагану Irma, а в якості тестових – Jose та Katia. Потім навпаки Jose – навчальна, Irma + Katia – тестова. Основна ідея була перевірити, чи навчаючи нейронну мережу по даним одного урагану можна передбачити час настання піків інших ураганів. На сьогодні існує величезний набір різних методів класифікації. В роботі було протестовано 8 різних класифікаторів.

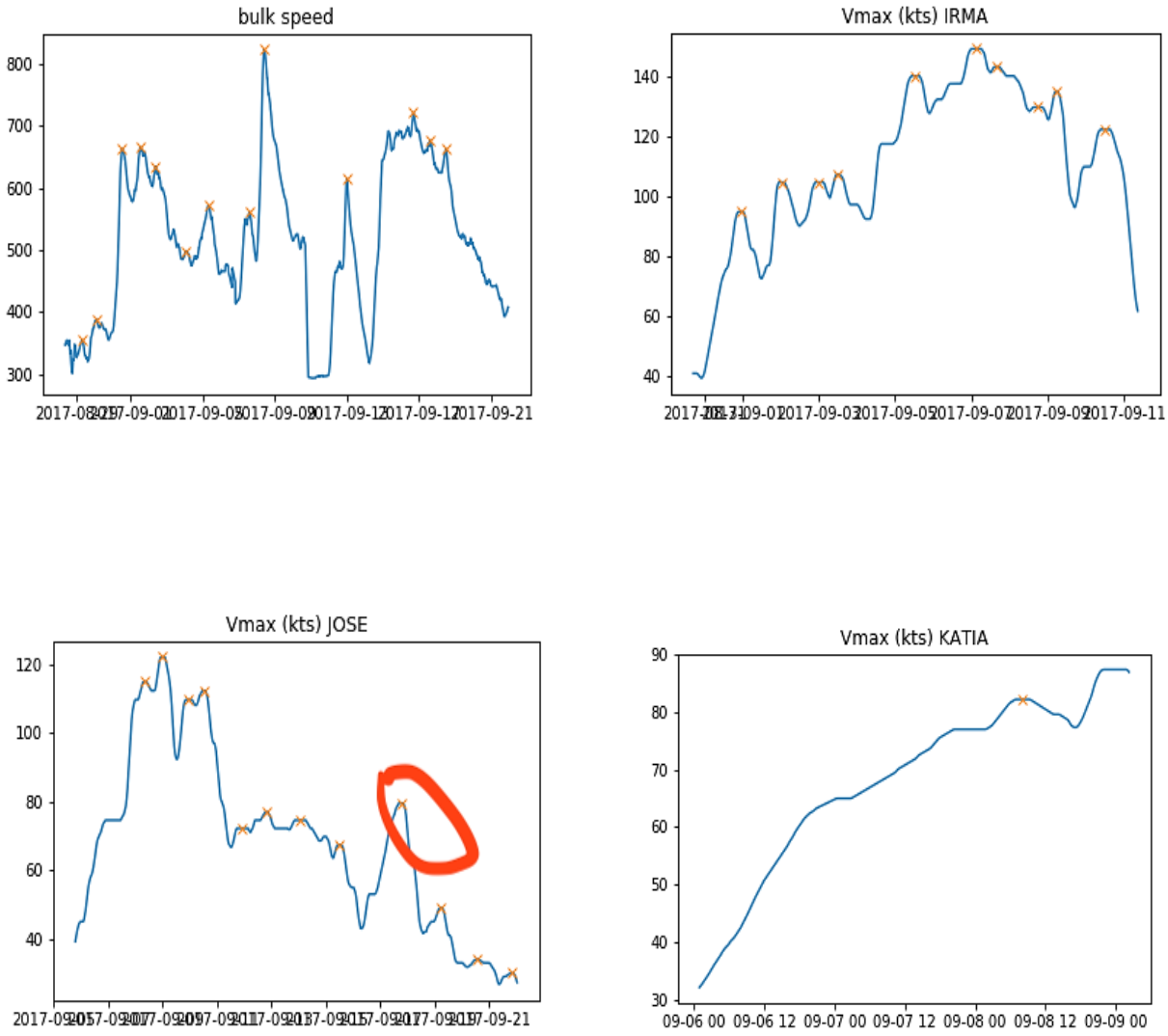


Рис. 2.19. Встановлені піки на графіках сонячного вітру та ураганів

Як показали тестові розрахунки: логістична регресія, дерево рішень, CN2 rule, Naive Bayes – виявились абсолютно непридатними для цих розрахунків. На противагу ним такі методи як: Random Forest, SVM, kNN та Back propagation Neural Network дали абсолютно однакові результати. У таблиці 2.16 наведено основні характеристики зазначених методів. У таблиці розміри та характеристики навчальних та тестових наборів.

На рисунку 2.20 представлена Orange схема розрахунків

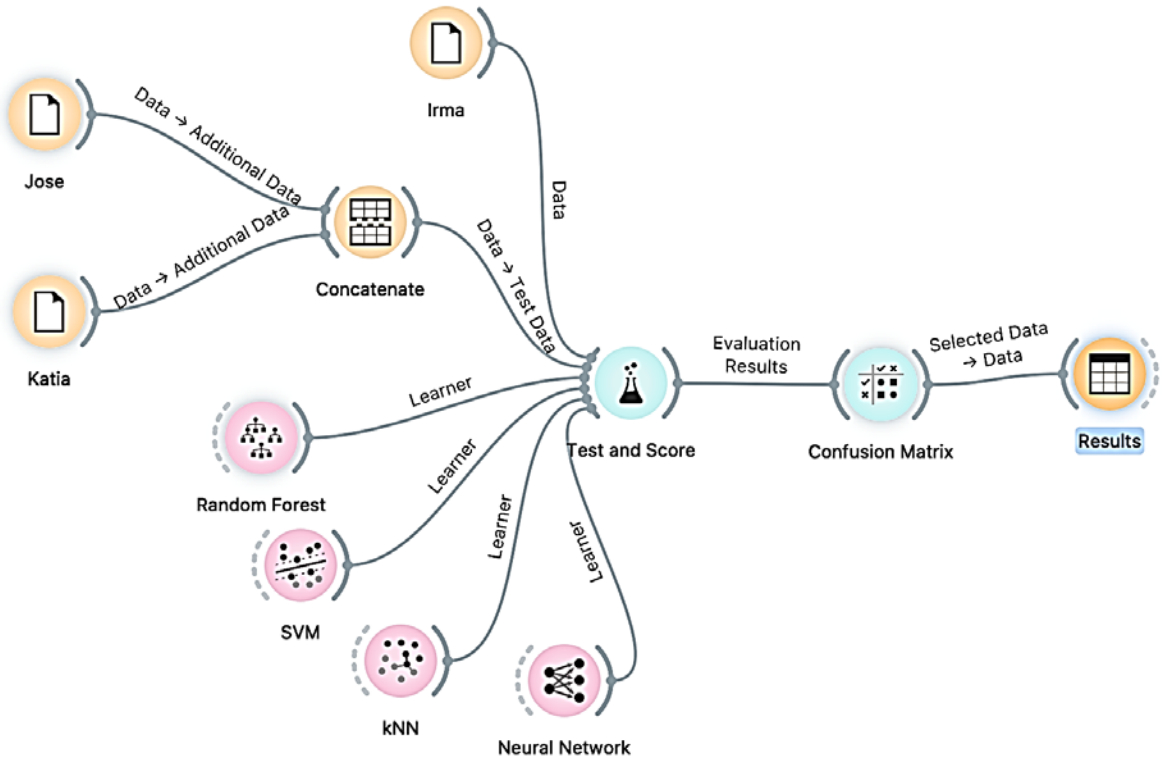


Рис. 2.20. Схема класифікаційного аналізу за допомогою множини класифікаторів

2.3. Паводки

2.3.1. Аналіз структури даних

Це розділ присвячений аналізу залежностей між опадами та повенями у Великій Британії спричиненими потоками частинок від Сонця на основі 20 повеней у період з жовтня 2001 р. по грудень 2019 р. З використанням машинного навчання і класифікаційного прогностичного моделювання, були встановлені приховані залежності між цими явищами та розроблена прогнозна модель.

Таблиця.2.16

Основні параметри класифікаційних моделей

Модель	Параметри
Random Forest	Number of trees: 10. Min split: 5
SVM	Cost: 1.00 ε: 0.1 Kernel: RBF Tolerance: 0.0001 Iteration limit: 100
kNN	Neighbors: 5 Metric: Euclidean Weight: uniform
NN	Hidden layers Structure: (100, 100) Activation: Relu Solver: Adam Max iteration: 200

Щоб перевірити можливий зв'язок між повінню, спричиненою опадами, та сонячною активністю, ми використали кілька наборів даних і джерел даних. Для аналізу були використані дані з 20 незалежних блоків даних для різних повеней (r). Кожен блок даних складався з окремих наборів даних:

- Flood (F):

  $DS_F^r =< Date, precipitations,$

  $days\ from\ the\ beginning\ of\ the\ flood\ >$

- INTEGRAL PROTON FLUX (IPF, p/cs2-sec-ster) :

  $DS_{IPF}^r =< Date, (IPF > 10\ \text{MeV}), (\text{IPF} > 30\ \text{MeV}) >$

- DIFFERENTIAL ELECTRON AND PROTON FLUX: (DF, p/cs2-sec-ster). Ці блоки містили різні характеристики сонячної енергії для різних періодів під час різних повеней. Виміряні діапазони для диференціального потоку електронів становили 38-53 кеВ і 175-315 кеВ для всіх досліджених повеней, тоді як виміряні діапазони для диференціального потоку протонів змінювалися залежно від періоду, в якому відбулася повінь. Диференціальні потоки протонів були виміряні в наступних діапазонах: 47-65 кеВ, 47-68 кеВ, 65-112 кеВ, 112-187 кеВ, 115-195 кеВ, 310-580 кеВ, 761-1220 кеВ, 795-1193 кеВ, 1060-1900 кеВ і 1060-1910 кеВ, але єдиним загальним діапазоном для всіх повеней був 310-580 кеВ. Ми використовували лише наступні ознаки, які були загальними для всіх повеней: $DS^r_{DF} =< Date,\ 38\ \text{keV} \leq \text{DF} \leq 53\ \text{keV}, 175\ \text{keV} \leq \text{DF} \leq 315\ \text{keV}, 310\ \text{keV} \leq \text{DF} \leq 580\ \text{keV} >$. (табл 2.17)
- SOLAR WIND (SW):

  $$DS^r_{SW} =< Date, Proton\ Density\ \left(\frac{\text{particles}}{\text{cc}}\right), Bulk\ Speed\ \left(\frac{\text{km}}{\text{s}}\right), Ion\ Temperature(\text{degrees K}) >$$

- 10.7 CM RADIO FLUX (RF, solar flux units):

  $DS^r_{RF} =< Date, Radio\ Flux\ >$

Дані вибирались в діапазоні 10 днів до і тиждень після паводку.

Таблиця 2.17

Поля DIFFERENTIAL FLUX (DS^{r}_{IPF}) для різних річок

River	DIFFERENTIAL FLUX						
2001_0645_GBR	38-53	175-315	47-65	112-187	310-580	761-1220	060-1910
2002_0463_GBR_1	38-53	175-315	65-112	112-187	310-580	761-1220	060-1910
2002_0463_GBR_2	38-53	175-315	65-112	112-187	310-580	761-1220	060-1910
2002_0488_GBR	38-53	175-315	65-112	112-187	310-580	761-1220	060-1910
2002_0774_GBR	38-53	175-315	47-68	115-195	310-580	761-1220	060-1910
2004_0423_GBR	38-53	175-315	47-68	115-195	310-580	761-1220	060-1910
2007_0201_GBR	38-53	175-315	47-68	115-195	310-580	761-1220	060-1910
2007_0247_GBR	38-53	175-315	47-68	115-195	310-580	761-1220	060-1910
2007_0278_GBR	38-53	175-315	47-68	115-195	310-580	761-1220	060-1910
2008_0055_GBR	38-53	175-315	47-68	115-195	310-580	761-1220	060-1910
2008_0381_GBR	38-53	175-315	47-68	115-195	310-580	761-1220	060-1910
2009_0497_GBR	38-53	175-315	47-68	115-195	310-580	761-1220	060-1910
2012_0446_GBR	38-53	175-315	47-68	115-195	310-580	795-1193	1060-1900
2012_0488_GBR	38-53	175-315	47-68	115-195	310-580	795-1193	1060-1900
2012_0548_GBR	38-53	175-315	47-68	115-195	310-580	795-1193	1060-1900
2012_0549_GBR	38-53	175-315	47-68	115-195	310-580	795-1193	1060-1900
2012_0552_GBR	38-53	175-315	47-68	115-195	310-580	761-1220	060-1910
2013_0572_GBR	38-53	175-315	47-68	115-195	310-580	761-1220	060-1910
2015_0561_GBR	38-53	175-315	47-68	115-195	310-580	761-1220	060-1910
2017-0490-GBR	38-53	175-315	47-68	115-195	310-580	761-1220	060-1910
2019_0568_GBR	38-53	175-315	47-68	115-195	310-580	761-1220	060-1910

2.3.2. Часова трансформація вхідних даних

Слід зазначити, що дані сонячної активності та дані по паводкам фіксувались з різним часовим інтервалом (табл.2.18.)

Таблиця 2.18

Часові інтервали вхідних та вихідного наборів даних

Набори даних	Часовий інтервал
DS^r_{IPF}, DS^r_{IPF}	5 хв
DS^r_{SW}	1 хв
DS^r_{RF}	1 або 3 рази в день
DS^r_F	1 раз в день

Для подальшого аналізу блоки даних для кожної річки були згруповані в окремі набори даних (DS^r), що згруповані до максимального інтервалу в 1 день.

$$DS^r = DS^r_{IPF} \cup DS^r_{IPF} \cup DS^r_{SW} \cup DS^r_{RF} \cup DS^r_F \qquad (2.40)$$

Так як згідно гіпотези ми досліджували вплив спалахів сонячної активності на повені, то в якості групування використовувались як максимальне значення max() так і відносний спалах протягом дня: $\frac{\max(X_i) - \min(X_i)}{\min(X_i)}$

2.3.3. Кореляційний аналіз

Так, як вхідні набори даних для різних річок були різними, спочатку була проведена спроба знаходження незалежних функціональних залежностей для кожної річки окремо. Для цього спочатку був проведений кореляційний аналіз між вхідними факторами та опадами з врахуванням часових (лагових) затримок. Для врахування лагової затримки кожен часовий ряд вхідного параметру зсувався по вертикалі вниз на необхідну кількість днів (лагів). Записи, в яких при цьому з'являлись пропущені дані, вилучались.

Результати приведені в таблиці 2.19. Як видно з таблиці, для всіх повеней не існує рівномірних лінійних залежностей між факторами, навіть з урахуванням лагів. Тобто, якщо в якійсь події повені існує висока кореляція для одного з факторів (наприклад, 2001_0645, R = 0,87, Lag = 2), але вона повністю відсутня для інших подій повені. Це свідчить про випадковість цієї залежності.

Таблиця 2.19

Максимальні значення коефіцієнтів кореляції між вхідними факторами та опадами для лага 0-3.

	DS^r_{IPF}		DS^r_{DF} (electron)		DS^r_{DF} (proton)										DS^r_{SW}			DS^r_F
flood events	**IPF > 10 MeV**	**IPF > 30 MeV**	**38 keV ≤ DF ≤ 53 keV**	**175 keV ≤ DF ≤ 31 keV**	**47 keV ≤ DF ≤ 65 keV**	**47 keV ≤ DF ≤ 68 keV**	**65 keV ≤ DF ≤ 112 keV**	**112 keV ≤ DF ≤ 187 keV**	**115 keV ≤ DF ≤ 195 keV**	**310 keV ≤ DF ≤ 580 keV**	**761 keV ≤ DF ≤ 1220**	**795 keV ≤ DF ≤ 1193 keV**	**1060 keV ≤ DF ≤ 1910 keV**	**1060 keV ≤ DF ≤ 1900 keV**	**BULK SPEED**	**ION TEMPERATURE**	**PROTON DENSITY**	**10.7 cm Radio Flux**
2001_0645	0.79	0.87	0.69	0.58	0.64	-		0.76	-	0.73	0.77	-	0.75	-	0.48	0.51	0.38	0.64
2002_0463	0.03	0.88	0.96	0.97	-	-	0.93	0.94	-	0.94	0.94	-	0.94	-	0.77	0.90	0.63	0.81
2002_0488	0.31	0.41	0.03	-0.03	-	-	0.59	0.33	-	0.04	0.07	-	0.07	-	0.24	0.37	0.43	0.39
2002_0774	0.05	0.04	0.10	0.06	-	0.30	-	-	0.38	0.36	0.25	-	0.27	-	0.24	0.25	0.53	0.05
2004_0423	0.27	0.27	0.82	0.65	-	-0.15	-	-	0.76	0.85	0.85		0.85		-0.15	0.19	0.06	0.57
2007_0201	-0.22	-0.21	0.05	0.68	-	0.16	-	-	0.11	-0.15	-0.26	-	-0.14	-	0.68	0.72	0.06	-0.39
2007_0247	0.05	0.06	-0.01	0.09	-	0.01	-	-	0.04	0.17	0.55	-	0.37	-	0.38	0.28	0.29	-0.22
2007_0278	0.63	0.73	0.06	0.74	-	-0.02	-		0.08	-0.07	-0.06	-	-0.06	-	0.01	0.83	0.77	-0.05
2008_0055	-0.10	-0.10	0.21	0.07	-	0.69	-	-	0.71	0.49	0.15	-	0.17	-	0.29	0.21	0.08	0.37
2008_0381	0.21	0.76	0.65	0.49	-	0.18	-	-	0.41	0.75	0.69		0.61		0.48	0.72	0.88	-0.02
2009_0497	0.33	0.33	0.04	-0.02	-	0.38	-	-	0.39	0.29	0.19	-	0.21	-	0.28	0.30	0.11	0.68
2012_0446	0.12	0.12	-0.08	-0.02	-	0.86		-	0.87	0.87	-	0.09	-	0.08	0.64	0.40	0.12	0.45
2012_0488	-0.26	-0.26	0.36	-0.21	-	0.39	-		0.08	-0.27	-	0.63	-	0.50	0.76	0.34	-0.27	-0.15
2012_0548	0.31	0.31	0.26	0.34	-	0.29	-	-	0.27	0.31	-	0.19	-	0.19	0.38	0.51	0.21	0.55
2012_0549	0.32	0.28	0.01	0.17	-	0.29	-	-	0.35	0.48	-	0.26	-	0.26	0.45	0.74	0.47	-0.07
2012_0552	0.22	0.14	0.35	0.16	-	-0.09	-	-	-0.10	-0.07	0.05	-	0.17	-	0.21	0.37	0.20	0.03
2013_0572	0.36	0.38	0.12	0.29	-	0.06		-	0.29	0.16	0.18	-	0.61	-	-0.14	-0.09	0.47	-0.05
2015_0561	-0.14	-0.20	-0.18	-0.05	-	0.24	-		0.13	0.09	0.84	-	0.84	-	0.29	0.17	0.52	0.71
2017-0490	0.51	0.53	0.06	0.17	-	0.90	-	-	0.36	-0.12	0.54	-	-0.12	-	0.65	-0.03	0.17	0.37
2019_0568	0.32	0.32	0.22	0.03	-	0.23	-	-	0.09	0.06	-0.07		-0.07		0.96	0.71	0.65	0.19

Модель прогнозу настання повені для кожної річки можна формалізувати так:

$$\text{Precipitations}_r = F\left(X_{r1}, \dots, X_{rm}, X_{1,t-1}, \dots, X_{m,t-1}, \dots, X_{1,t-n}, \dots, X_{m,t-n}\right), \quad (2.41)$$

де r – індекс річки, m – кількість вхідних параметрів, n – максимальний лаг.

Слід зазначити, що врахування лагу призводить як до збільшення вхідних параметрів (якщо модель враховує значення певного фактора за декілька днів, а не просто зміщені на лаг), так і до зменшення записів. Оскільки кількість записів для кожної повені коливається від 11 до 38, навіть лаг 2 призводить до того, що кількість вхідних параметрів перевищує кількість записів, що унеможливлює використання як нелінійного, так і лінійного методів. Отже, цю задачу можна вирішити, об'єднавши всі дані в один набір даних:

$$DS = \bigcup_{r=1}^{20} DS^r \quad (2.42)$$

Для подальшого аналізу були залишені лише ті характеристики сонячної активності, які присутні у всіх наборах даних DS^r, а саме: $IPF > 10$ MeV (X_1), $IPF > 30$ MeV (X_2), $38 \text{ keV} \leq \text{DF} \leq 53$ keV (X_3), $175 \text{ keV} \leq \text{DF} \leq 315$ keV (X_4), $310 \text{ keV} \leq \text{DF} \leq 580$ keV (X_5), PROTON DENSITY (X_6), BULK SPEED (X_7), ION TEMPERATURE (X_8), 10.7 cm Radio Flux (X_9).

2.3.4. Проблема дисперсії

Об'єднання наборів даних спричинює появу іншої проблеми – різна дисперсія вихідних даних, адже дані по опадам були отримані для різних річок. Тому подавати на вихід таке поле як опади не є коректним, адже різні річки по різному реагують на кількість опадів і кількість опадів також

залежить від географічного розташування річки. Тому було запропоновано в якості вхідних параметрів використовувати не абсолютні значення піків, а їх час настання (положення) на графіках сонячної активності (рис.2.21). В якості вихідного поля слугувала дата початку паводку. Дні паводку позначались як True. Дні без паводку – False. Фактично задача тоді зводиться до задачі бінарної класифікації де на вхід подаються значення True/False по кожному полю з врахуванням лагу. А на виході теж бінарне поле. Для отримання фінального набору даних для кожної річки розраховувались положення піків для полів сонячної активності, а також фіксувався початок та закінчення паводку. Положення піків визначалось програмно з подальшою ручною верифікацією. Після бінаризації проводилась лагова трансформація цих бінарних наборів даних. Для цього кожне вхідне поле дублювалось та проводився зсув по вертикалі на необхідну кількість лагів. Записи, в яких при цьому з'являлись пропуски – вилучались. Фрагмент результуючого набору даних представлено в таблиці 2.20.

Таблиця 2.20

Фрагмент результуючого набору даних для лагу від 0 до 9

> 10 MeV(t-0)	> 30 MeV(t-0)	38-53(t-0)	175-315(t-0)	310-580(t-0)	PROTON DENSITY(t-0)	...	310-580(t-9)	PROTON DENSITY(t-9)	BULK SPEED(t-9)	ION TEMPERATURE(t-9)	10.7 cm Radio Flux(t-9)	days from the beginning of the flood
False	True	False	True	True	False	...	False	False	False	False	False	False
True	False	True	False	False	True	...	True	False	False	False	False	True
False	False	False	False	False	False	...	False	True	False	False	False	True
False	False	False	False	False	False	...	False	False	False	True	True	True
False	False	False	False	False	False	...	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...
False	False	False	False	True	False	...	False	False	False	False	False	False
True	True	False	False	False	False	...	False	True	False	True	False	True
False	False	False	False	False	False	...	True	False	False	False	False	True
False	False	False	False	True	False	...	False	False	False	False	False	False
False	False	False	False	False	False	...	False	False	True	True	False	True

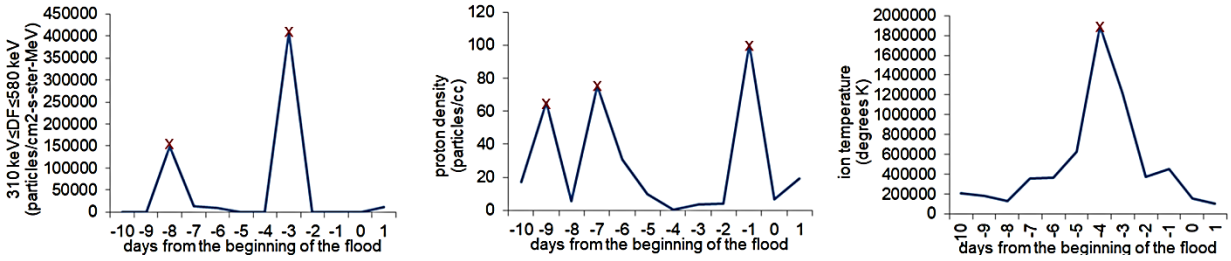


Рис.2.21. Приклад встановлення піків.

2.3.4. Класифікація та прогноз паводків

2.3.4.1. Метрики оцінювання

Для оцінки якості навчання моделі існують чотири різні метрики, а саме: 'accuracy', 'precision', 'recall', 'f1'. В нашому випадку є важливим передбачення саме паводку. Ситуація, коли модель помилково передбачає паводок, а на справді його не було є не важливим в нашому випадку. Помилка ж моделі, коли згідно прогнозу паводку не має, а він насправді є – є критичною. Для оцінки саме таких ситуацій служить метрика 'recall'. Саме вона оцінює точність позитивних прогнозів. Інші ж метрики враховують прогноз як настання паводку так і його відсутність. Тому ці метрики апріорі будуть мати вищі значення точності, але вони не є адекватними в нашому випадку.

$$recall = tp / (tp + fn), \tag{2.43}$$

де tp — кількість істинних позитивних результатів, а fn — кількість помилкових негативних результатів.

2.3.4.2. Вибір моделей

На сьогоднішній день існує величезна кількість класифікаційних моделей та не існує однозначного правила їх вибору. Можна зазначити, що переважна кількість класифікаційних моделей не дозволяє описати правила класифікації, чи побудувати дерево рішень. Тому в роботі розглядались два підходи:

1. Побудова прогнозу на основі одного класифікатора – дерева рішень
2. Побудова прогнозу на основі ансамблю моделей.

В першому випадку можна наглядно зрозуміти та обґрунтувати прийняте рішення по класифікації. В другому можна досягнути значно більшої точності прийняття рішення.

Точність моделі визначалась шляхом кроссвалідації, згідно якого навчальний набір ділився випадковим чином на 3 частини. Кожна з цих частин по черзі виступала як тестова. Тобто класифікатор навчався тричі на 3-х різних наборах даних. Для кожного випадку розраховувалась точність тестового та навчального наборів та усереднювалась. Аналіз величин цих метрик дав змогу оцінити точність, адекватність та наявність перенавчання.

2.3.4.3. Дерево рішення

Для визначення правил класифікації та візуалізації результатів було використано метод дерева рішень [116]. Це непараметричний метод навчання з учителем і один із широко використовуваних алгоритмів класифікації. Алгоритм дерева рішень будує гілки дерева за ієрархічним підходом. Кожна гілка використовує правило «що-якщо» та ділить набір даних на підмножини на основі найважливіших характеристик. Основна ідея дерева рішень полягає в тому, щоб визначити характеристики, які містять найбільше інформації про цільову функцію, а потім розділити набір даних разом із значеннями цих характеристик, щоб значення цільових характеристик у результуючих вузлах були максимально чистими. як можна. Правила вивчаються послідовно, використовуючи навчальні дані одне за одним. Кожного разу, коли вивчається правило, кортежі, які покривають правила, видаляються. Мета полягає в тому, щоб створити модель, яка передбачає значення цільової змінної шляхом вивчення простих правил прийняття рішень, отриманих на основі характеристик даних. Дерево можна розглядати як кусково-постійне наближення. Основною перевагою вибору цього методу є простота розуміння та можливість візуалізації результату, тоді як

недоліком є складність роботи з відсутніми даними та те, що він може створювати складні дерева, які можуть бути неефективно класифіковані. Індекс Джині був обраний як критерій для вимірювання порогу розщеплення [117]. Це показник нерівності розподілу деякого значення чисел, який розраховує ймовірність конкретної ознаки, яка класифікується неправильно при виборі. випадковим чином. Стратегія, яка використовується для вибору розподілу в кожному вузлі, полягає в пошуку найкращого розподілу.

2.3.4.4. Ансамбль моделей

Ансамблеві методи поєднують передбачення з кількох моделей, щоб отримати кращу ефективність прогнозування, ніж можна було б отримати за допомогою будь-якого із складових алгоритмів навчання окремо. Існує три різні способи побудови модельних ансамблів, бегінг, стекінг та бустінг [118, 119]. У цьому дослідженні було використано 25 різних алгоритмів машинного навчання (таблиця 2.21) з різними параметрами та 3 ансамблі. Також ми протестували ансамблі моделей, заснованих на бустингу (Ada Boost Classifier і Gradient Boosting Classifier) і бегінгу (Bagging Classifier). Після цього ми об’єднали їх усіх в одну остаточну модель ансамблю шляхом жорсткого голосування:

Таблиця 2.21

Перелік класифікаторів та ансамблів, які використовувалися в розрахунках

№	*Класифікатори*
1.	DecisionTreeClassifier()
2.	LogisticRegression(random_state=1)
3.	QuadraticDiscriminantAnalysis()
4.	GaussianNB()
5.	RandomForestClassifier(max_depth=5, max_features=1, n_estimators=10)

Продовження таблиці 2.5

№	***Класифікатори***
6.	SVC(decision_function_shape='ovo')
7.	SGDClassifier()
8.	MLPClassifier(alpha=1e-05, hidden_layer_sizes=(20, 10), random_state=1,solver='lbfgs')
9.	ExtraTreesClassifier(random_state=0)
10	KNeighborsClassifier(n_neighbors=3)
11	OutputCodeClassifier(estimator=RandomForestClassifier(random_state=0),random_state=0)
12	OneVsOneClassifier(estimator=LinearSVC(random_state=0))
13	OneVsRestClassifier(estimator=SVC())
14	RidgeClassifier()
15	PassiveAggressiveClassifier(random_state=0)
16	GaussianProcessClassifier(kernel=1**2 * RBF(length_scale=1), random_state=0)
17	BernoulliNB()
18	LabelPropagation()
19	LabelSpreading()
20	LinearDiscriminantAnalysis()
21	LinearSVC(random_state=0, tol=1e-05)
22	MultinomialNB()
23	NearestCentroid()
24	Perceptron()
25	SVC(gamma='auto')
	Ансамблі
26	AdaBoostClassifier (n_estimators =100, random_state =0)
27	GradientBoostingClassifier (Learning_rate =1,0, max_depth =1, random_state =0)

№	*Класифікатори*
28	BaggingClassifier (base_estimator =SVC(), random_state =0)
29	VotingClassifier (…,voting='hard')

2.4. Висновки до розділу 2

В результаті дослідження був сформований комплекс передових математичних методів, включаючи сплайн-інтерполяцію, кореляційний аналіз, лінійні моделі, адаптивну нейронечітку систему інференційних фунуцій (ANFIS) та нейронні мережі зворотнього поширення помилки мережі (ANN) та довготривалої короткочасної пам'яті (LSTM) що дозволило детально проаналізувати та підтвердити зв'язок між сонячною активністю та виникненням природних катастроф, таких як лісові пожежі, урагани та повені.

В результаті проведених досліджень:

- Здійснено аналіз впливу сонячної активності на природні катастрофи, зокрема лісові пожежі, урагани та повені, за допомогою кореляційного та лагового аналізу, використовуючи сплайн-інтерполяцію для заповнення прогалин у даних, що на відміну від інших досліджень дозволило виявити вплив затримок сонячної активності на атмосферні явища. Зокрема було встановлено, що вони становили 4, 5 та 10 днів при прогнозуванні лісових пожеж, ураганів та паводків відповідно.
- Запропоновано ансамблі математичних методів, включаючи лінійні моделі, ANFIS, ANN та LSTM, для прогнозування кризових явищ на основі даних про сонячну активність, що дозволило підвищити точність прогнозів порівняно з традиційними методами.
- Розроблено метод прогнозування паводків в залежності від піків

сонячної активності, що на відміну від класичних методів прогнозування дало змогу робити прогнози на основі невеликої кількості географічно розподілених вхідних даних.

– Розроблено методи консолідації розрізнених даних сонячної активності та даних про природні катастрофи в набір даних, що дає змогу використовувати для подальшого аналізу та прогнозування методами штучного інтелекту.

Основні наукові результати розділу опубліковані в працях [120 – 124].

РОЗДІЛ 3.
МОДЕЛЮВАННЯ ПРИРОДНИХ КАТАСТРОФ

3.1. Лісові пожежі

3.1.1 Лісові пожежі в Португалії

3.1.1.1. Результати моделювання

Порівняння моделей, що містять фактори *I1* та *I2,* дозволяє зробити висновок, що фактор *I1* дозволяє краще описати вихідні поля. Це вірно для моделей Linear і ANFIS.

Як показали розрахунки, врахування фактора, що описує диференціальний потік електронів, дозволило незначне збільшення коефіцієнтів кореляції для моделей Linear та ANFIS. Слід зазначити, що вплив факторів *E1* і *E2* приблизно однаковий. Тому для подальших розрахунків було обрано коефіцієнт *E2* .

Отже, на наступному етапі ми досліджували найточніші моделі:

$$T(H, P) = \mathrm{F}\,(I1_{\mathrm{l}}, W1_{\mathrm{l}}, W2_{\mathrm{l}}, W3_{\mathrm{l}}, E2_{\mathrm{l}}), \qquad (3.1)$$

де l – відставання.

Як чітко видно з таблиці 2.5, існує велика кількість моделей, більш точних, ніж (2.1) (стовпці 2-3 і 4-5). Як показує останній стовпець таблиці моделей, моделі (2.1) займають далеко не перші місця серед точних моделей. Тому класичним підходом до визначення точних моделей є екв. (2.1), описане в етапі аналізу затримки, не підходить для цього класу завдань. На рисунку 3.1 представлено розподіл коефіцієнтів кореляції для всіх моделей (3.1). Як видно з рисунка 3.1, існує багато моделей, які можуть робити прогнози полів виробництва з високим рівнем точності. Це може бути основою для створення багатомодельної експертної системи прогнозування кризових подій.

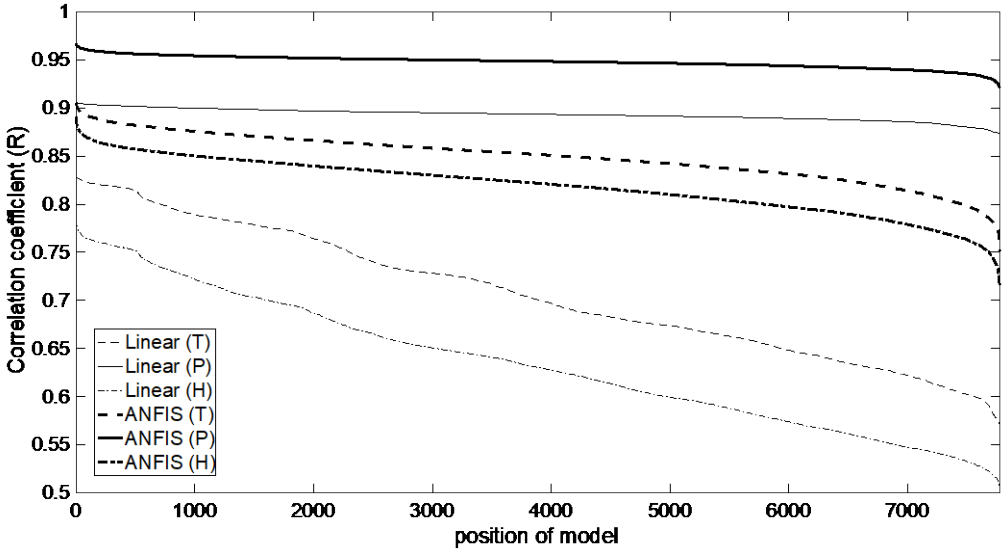


Рис. 3.1. Розподіл коефіцієнтів кореляції всіх можливих моделей (3.1)

3.1.1.2. Аналіз точності

Для підтвердження цього висновку була перевірена адекватність трьох найбільш точних моделей:

- Лінійний з вищим коефіцієнтом кореляції,
- ANFIS з більш високим коефіцієнтом кореляції,
- Модель із вищим загальним ($R_{Linear} + R_{ANFIS}$) коефіцієнтом кореляції.

Інформація про поля введення цих моделей представлена в таблиці 3.1.

Як видно з таблиці 3.1., найкраща лінійна модель для температури:

$$T_1 = \mathrm{F}(I1_1, W1_5, W2_0, W3_0, E2_5). \quad (3.2)$$

Найкраща модель ANFIS:

$$T_2 = \mathrm{F}(I1_0, W1_5, W2_5, W3_3, E2_5). \quad (3.3)$$

Модель з вищим сумарним коефіцієнтом кореляції:

$$T_3 = \mathrm{F}(I1_5, W1_5, W2_2, W3_3, E2_5). \quad (3.4)$$

Як бачимо, лаги полів введення цих моделей різні.

Таблиця 3.1

Лаги (рівн. 2.1) найкращих моделей для прогнозування *T, H, P*

Модель	Найкраще для:	$I1_{л}$	$W1_{л}$	$W2_{л}$	$W3_{л}$	$E2_{л}$	*P*
		температура					
T_1	Лінійний	1	5	0	0	5	0,8287
T_2	ANFIS	0	5	5	3	5	0,9051
T_3	Лінійний+ANFIS	5	5	2	3	5	1,7215
		Вологість					
H_1	Лінійний	1	5	0	0	5	0,7831
H_2	ANFIS	5	4	4	2	5	0,8910
H_3	Лінійний+ANFIS	4	5	5	3	4	1,6392
		Тиск					
$П_1$	Лінійний	4	3	3	3	4	0,9061
P_2	ANFIS	5	0	5	3	1	0,9673
P_3	Лінійний+ANFIS	5	0	5	3	0	1,8684

Крім того, лагі полів введення цих моделей іноді значно менші, ніж у (2.1). З іншого боку, це підтверджує, що мультимодельний підхід може робити прогнозування на різні періоди часу від 0 до 5 годин.

За даними для лагів з табл. 3.1. побудовано дев’ять лінійних та дев’ять моделей ANFIS:

$$T_1 = 195 - 88{,}8 \cdot I_1 + 0{,}58 \cdot W1_5 + 0{,}015 \cdot W2_0 - 1{,}96 \cdot 10^{-5} \cdot W3_0 + 4{,}43 \cdot 10^{-5} \cdot E2_5 ,$$

$$T_2 = 193 - 87{,}15 \cdot I_0 + 0{,}55 \cdot W1_5 + 0{,}072 \cdot W2_5 - 9{,}47 \cdot 10^{-6} \cdot W3_3 + 2{,}59 \cdot 10^{-5} \cdot E2_5 ,$$

$$T_3 = 220 - 98{,}7 \cdot I_5 + 0{,}56 \cdot W1_5 - 6{,}29 \cdot 10^{-4} \cdot W2_2 - 4{,}68 \cdot 10^{-6} \cdot W3_3 + 2{,}97 \cdot 10^{-5} \cdot E2_5 ,$$

$$H_1 = -4{,}53 + 2{,}76 \cdot I_1 - 0{,}02 \cdot W1_5 - 8{,}79 \cdot 10^{-4} \cdot W2_0 + 8{,}80 \cdot 10^{-7} \cdot W3_0 - 2{,}12 \cdot 10^{-6} \cdot E2_5 ,$$

$$H_2 = -6{,}37 + 3{,}41 \cdot I_5 - 0{,}014 \cdot W1_4 + 4{,}08 \cdot 10^{-4} \cdot W2_4 - 1{,}29 \cdot 10^{-7} \cdot W3_2 - 1{,}75 \cdot 10^{-6} \cdot E2_5 ,$$

$$H_3 = -6{,}43 + 3{,}50 \cdot I_4 - 0{,}019 \cdot W1_5 + 1{,}34 \cdot 10^{-4} \cdot W2_5 + 1{,}46 \cdot 10^{-7} \cdot W3_3 - 1{,}48 \cdot 10^{-6} \cdot E2_4 ,$$

$$P_1 = 888 + 64{,}47 \cdot I_4 + 0{,}02 \cdot W1_3 + 0{,}01 \cdot W2_3 + 7{,}77 \cdot 10^{-6} \cdot W3_3 - 1{,}39 \cdot 10^{-5} \cdot E2_4 ,$$

$$P_2 = 886 + 65{,}05 \cdot I_5 - 0{,}002 \cdot W1_0 - 0{,}01 \cdot W2_5 + 6{,}08 \cdot 10^{-6} \cdot W3_3 - 8{,}60 \cdot 10^{-6} \cdot E2_1 ,$$

$$P_3 = 886 + 65{,}07 \cdot I_5 - 0{,}001 \cdot W1_0 - 0{,}01 \cdot W2_5 + 6{,}13 \cdot 10^{-6} \cdot W3_3 - 1{,}57 \cdot 10^{-5} \cdot E2_0 .$$

Після навчання для кожної моделі ANFIS ми отримали набір функцій належності змінних, правил, методів фазифікації та дефазифікації тощо [125]. Атрибути отриманих функцій приналежності вхідних факторів представлені в таблиці 3.2. Як зазначалося вище, кожен вхідний фактор складається з 2 функцій приналежності.

Таблиця 3.2

Параметри функцій належності (4) Моделі ANFIS

	[σc]				
Модель	***I1*** л	***W1*** l	***W2*** l	***W3*** l	***E2*** л
	температура				
T 1	[0,05 1,97] [0,03 2,10]				
T 2	[0,02 1,95] [0,04 2,09]	[11,0 1,0] [11,0 27,0]	[111 362] [111624]	[16706130599] [167061423999]	[63953-99999] [6395350600]
T 3	[0,041,95] [0,05 2,09]				
	Вологість				
H 1	[0,06 1,98] [0,04 2,11]				
H 2	[0,04 1,96] [0,05 2,09]	[11,0 1,0] [11,0 27,0]	[111 362] [111624]	[16706130599] [167061423999]	[63953-99999] [6395350600]
H 3	[0,04 1,96] [0,053 2,097]				

Продовження таблиці 3.2

	Тиск				
П 1	[0,03 1,96] [0,03 2,11]				
P 2	[0,03 1,97] [0,04 2,10]	[11,0 1,0] [11,0 27,0]	[111 362] [111624]	[16706130599] [167061423999]	[63953-99999] [6395350600]
P 3	[0,04 1,96] [0,04 2,10]				

Як видно з табл. 3.2., під час навчання змінювалися лише параметри функцій належності I1. Це підтверджує, що це поле є найбільш значущим у цих моделях. Функції приналежності Гаусса для моделей ANFIS T1-T3 представлені на рисунку 3.2. Як показано на рисунку, зміна цих функцій є значною.

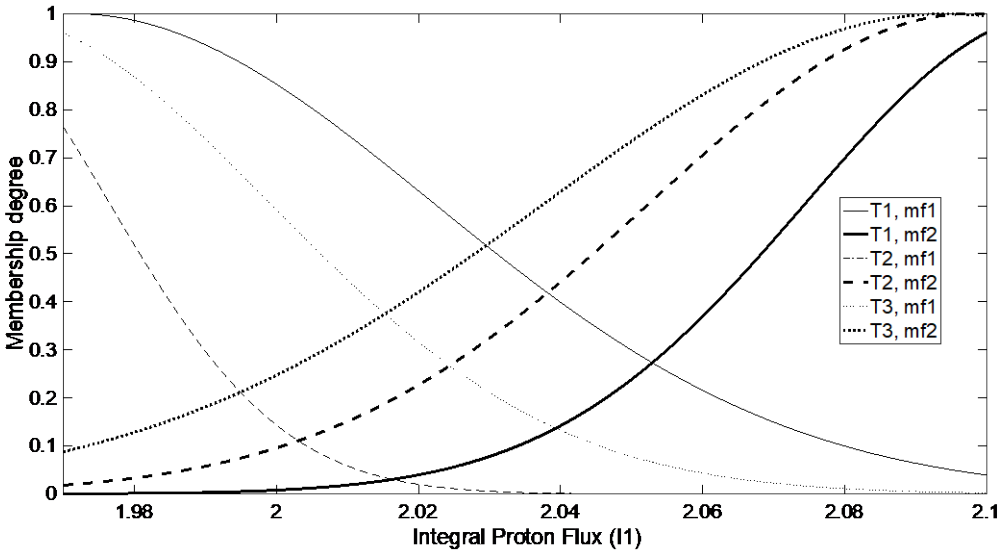


Рис. 3.2. Зміна функцій належності для моделей ANFIS T_1 -T_3

Для перевірки точності цих моделей результати модельного прогнозування порівнювали з реальними даними та розраховували коефіцієнт кореляції (табл. 3.3.).

Як видно з таблиці 3.3., всі моделі мають високий R . Усі моделі ANFIS мають вищий коефіцієнт кореляції, ніж лінійні. Це підтверджує, що моделі ANFIS більш точні та враховують нелінійні ефекти. Для візуального

порівняння результатів, прогнозовані значення, отримані моделями, у порівнянні з фактичними даними представлені на рисунку 3.3.

Таблиця 3.3

Коефіцієнти кореляції Пірсона моделей з таблиці (6)

Модель	Лінійна	ANFIS
T_1	0,8287	0,8714
T_2	0,7697	0,9051
T_3	0,8204	0,9012
H_1	0,7831	0,8374
H_2	0,7076	0,8910
H_3	0,7629	0,8763
$П_1$	0,9061	0,9581
P_2	0,9010	0,9673
P_3	0,9032	0,9652

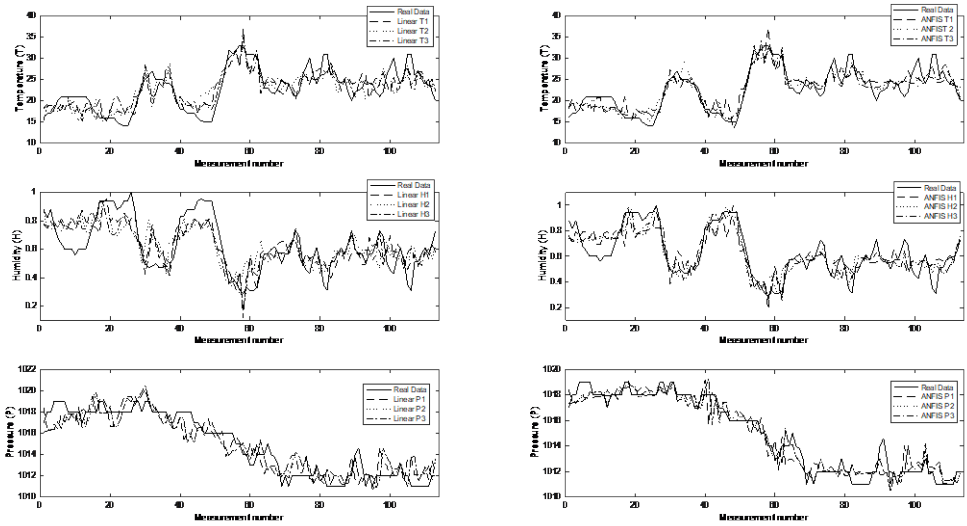


Рис.3.3. Порівняння результатів, прогнозованих значень, отриманих моделями Linear і ANFIS з таблиці 6, у порівнянні з фактичними даними

3.1.1.3. Аналіз чутливості

Як видно з рисунків, моделі ANFIS краще описують і прогнозують коливання амплітуди. Це добре видно на моделях Humidity. Незважаючи на високий коефіцієнт моделей ANFIS, лінійні моделі також точно описують досліджувані вихідні поля. Тому для вибору правильного типу моделі потрібен відповідний аналіз і аналіз чутливості.

Таблиця 3.4

Аналіз чутливості моделей з Таблиці 3.1.

Модель	Найкраще для:	***I1*** $_{л}$	***W1*** $_{l}$	***W2*** $_{l}$	***W3*** $_{l}$	***E2*** $_{л}$
		температура				
T 1	Лінійний	-82%	2%	3%	-1%	0%
	ANFIS	-39%	0%	3%	0%	0%
T2	Лінійний	-80%	1%	2%	-1%	0%
	ANFIS	-30%	3%	31%	-1%	0%
T 3	Лінійний	-91%	1%	0%	0%	0%
	ANFIS	-57%	1%	11%	1%	0%
		Вологість				
H 1	Лінійний	96%	-3%	-7%	3%	0%
	ANFIS	72%	-1%	-15%	1%	0%
H 2	Лінійний	116%	-2%	3%	0%	0%
	ANFIS	91%	1%	-16%	-1%	0%
H 3	Лінійний	121%	-2%	1%	0%	0%
	ANFIS	58%	-2%	-18%	2%	0%
		Тиск				
П 1	Лінійний	1,29%	0,00%	-0,05%	0,01%	0,00%
	ANFIS	0,36%	0,00%	-0,07%	0,01%	0,00%
P 2	Лінійний	1,30%	0,00%	-0,04%	0,01%	0,00%
	ANFIS	0,39%	0,00%	-0,05%	0,02%	0,00%
P 3	Лінійний	1,30%	0,00%	-0,04%	0,01%	0,00%
	ANFIS	0,43%	-0,01%	-0,07%	0,01%	0,00%

Для цього для кожної моделі був зроблений такий розрахунок:

1. Для кожного рядка навчальної множини кожне значення вхідного параметра токарної обробки змінено на 10%,

2. Розраховано відносну зміну вихідного поля при зміні окремого вхідного поля,

3. Середні дані по всіх записах.

Результати цих розрахунків представлені в таблиці 3.4.

Як видно з таблиці, -82% для лінійного T_1 означає, що в середньому, якщо коефіцієнт *I1* збільшиться на 10%, температура знизиться на 82% через 1 годину. Аналогічно, те саме збільшення *I1* призведе до зниження температури через 5 годин на 57% відповідно до моделі ANFIS. Як видно, найбільш значущими факторами є *I1* і *W2* . Ці результати також підтвердили, що електрони не впливають на досліджувані вихідні поля. Результати показують, що збільшення *I1* призведе до зниження температури та підвищення вологості. Навпаки, збільшення *W2* призведе до підвищення температури та зниження вологості. Чітко видно, що вхідні фактори мають слабкий вплив на тиск, незважаючи на найвищий коефіцієнт кореляції моделей.

3.1.2. Лісові пожежі в США

3.1.2.1. Результати моделювання

Отже в результаті навчання для кожного з шести лагів було отримано по 4 моделі на основі нейронних мереж (всього 24) та по 1 (всього 6) на основі ANFIS для великих і малих пожеж відповідно (разом: 48 – нейронні мережі, 12 – ANFIS). Як відомо, результат навчання нейромережі залежить від конфігурації, методу навчання, стохастичних параметрів. Результат навчання ANFIS характеризується більшою стійкістю при навчанні. Тому для аналізу були взяті усереднені дані для 4-х нейронних мереж. Як було показано в попередній роботі [126] такий підхід дозволяє відсіяти випадкові флуктуації у функціонуванні нейронних мереж, а отже досягнути кращих результатів. Для перевірки точності моделей був проведений кореляційний аналіз між реальними значеннями кількості пожеж $\tilde{F}^{small(large)}$ і

прогнозованими за допомогою моделей $M_L^{small(large)}$ для кожного лагу окремо. Це дало змогу встановити інтервал часу між початком лісових пожеж та сонячною активністю (рис. 3.4).

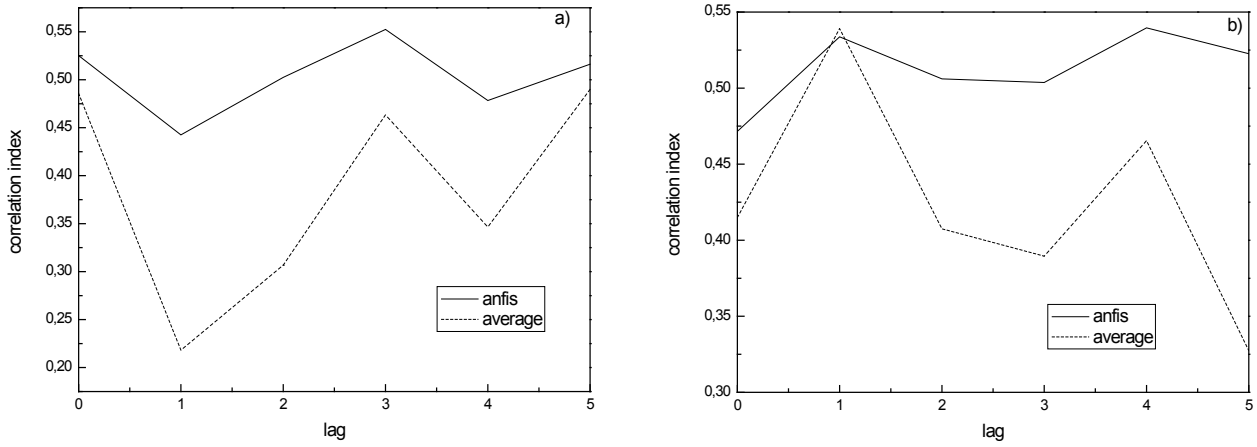


Рис. 3.4. Залежність коефіцієнта кореляції $\tilde{F}^{small(large)}$ (a) та $M_L^{small(large)}$ (b) від лагу L.

Як видно з рисунку, коефіцієнти кореляції для моделі на основ ANFIS є більшими за нейронні мережі. Крім того видно, що на графіках спостерігаються схожі тенденції, зокрема у випадку великих пожеж є наявними незначні піки для lag =1, 4. Отже можна дійти до висновку, що існує затримка в 1 або 4 доби від початку сонячної активності і настанням великих лісових пожеж, що спричинені нею. Аналогічна ситуація спостерігається для невеликих пожеж. Однак максимальна кореляція спостерігається при lag = 0, 3. Як видно з графіку ANFIS, різниця між коефіцієнтами кореляції є незначною. Натомість нейромережі демонструють більшу «чутливість» до лагу, не зважаючи на те, що абсолютні значення коефіцієнта кореляції є меншими. Для перевірки отриманого висновку про залежність від часового лагу був проведений порівняльний аналіз збігів числа малих і великих лісових пожеж між реальними даними та моделями (рис. 3.5). Також були проаналізовані помилкові піки та відмінність амплітуди піків.

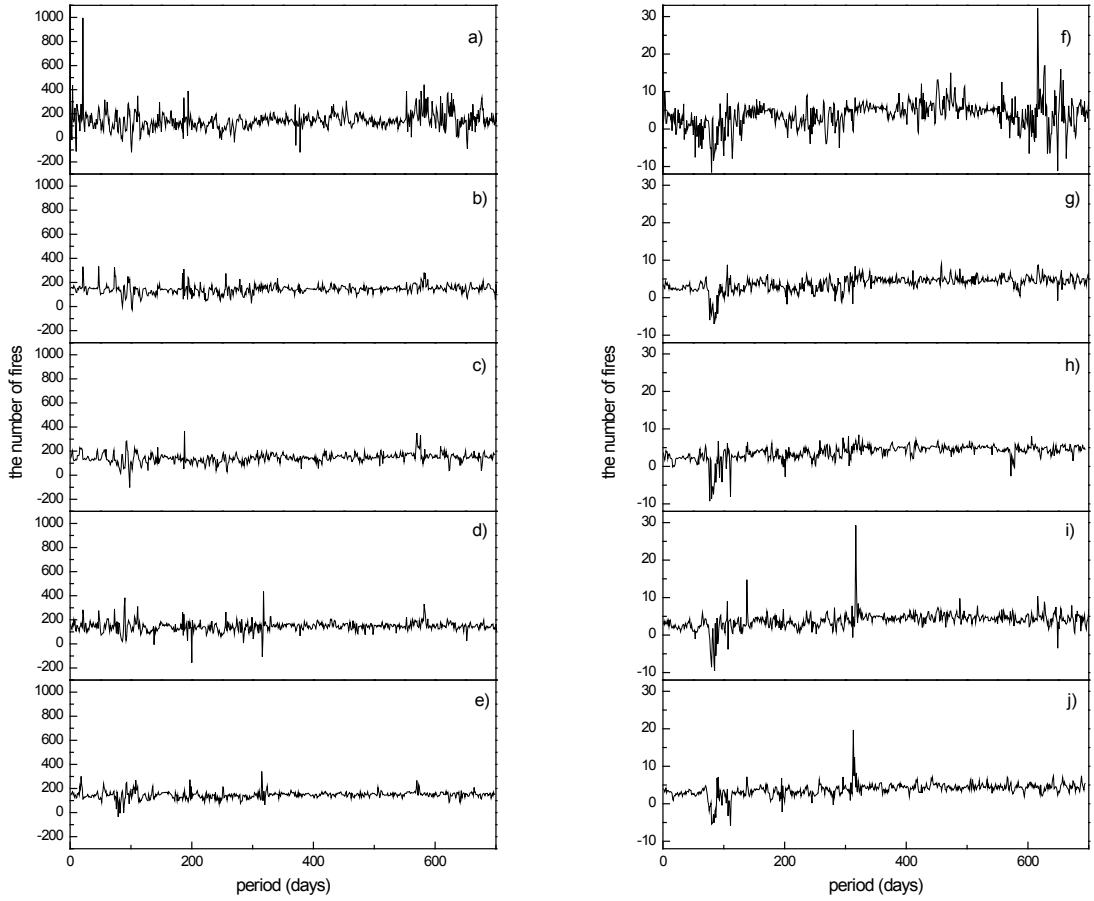


Рис. 3.5. Порівняння результатів моделювання з реальними даними. Малі пожежі: a) реальні дані, b) ANFIS (lag = 0), c) ANFIS (lag = 3), d) нейронні мережі (lag = 0), e) нейронні мережі (lag = 3); великі пожежі: f) реальні дані, g) ANFIS (lag = 1), h) ANFIS (lag = 4), i) нейронні мережі (lag = 1), j) нейронні мережі (lag = 4).

Як можна побачити з рисунку, всі моделі дають можливість пояснити основний вплив сонячної активності на малі та великі лісові пожежі. Модельні піки за положенням та амплітудою є близькими до реальних. Це вказує на достовірність цих моделей. Однак для точного аналізу необхідно кількісно підрахувати вищезазначені показники.

3.1.2.2. Аналіз точності

Для перевірки точності результатів, був поведений порівняльний аналіз між числом, положенням, амплітудою реальних спалахів пожеж (піки на рис.3.5(*a, f*)) і спалахами пожеж, що прогнозуються моделями (піки на рис.3.5.(*b , c , d, e , g, h, j, i*)). Розглянемо два випадки:

- порівняльний аналіз прогнозованих спалахів пожеж з точністю в межах 1 доби (тобто прогноз вважається правильним, якщо пожежа настає в той самий день, що і згідно пронозу);

- порівняльний аналіз прогнозованих спалахів пожеж з точністю в межах ± 1 доби (тобто, якщо модель передбачає спалах пожежі наприклад у середу, а реальна пожежа сталась в проміжку з вівторка до четверга – вважається, що прогноз точний).

Результати розрахунків наведені в таблиці 3.5 та 3.6.

Таблиця 3.5

Аналіз точності пожеж, проведених методом ANFIS

Лаг	**Реальна кількість пожеж**	**Прогнозовані спалахи**	**Пояснені моделлю спалахи пожеж з**		**Середня різниця в амплітуді**	**Помилкові піки з точністю 1 доба**		**Не пояснені моделлю спалахи**		**Пояснені моделлю спалахи**		**Середня різниця в**	**і Помилкові піки з точністю ±1**	
1.	2.	3.	4.		5.	6.		7.		8.		9.	10.	
Малі пожежі														
0	207	189	73	35%	-4,6%	116	61%	48	23%	169	82%	-4,4%	20	11%
1	206	187	59	29%	-7,7%	128	68%	53	26%	170	83%	-3,4%	17	9%
2	204	197	78	38%	-5,1%	119	60%	44	22%	178	87%	-3,4%	19	10%
3	202	185	78	39%	-5,1%	107	58%	41	20%	170	84%	-2,1%	15	8%
4	202	180	65	32%	-1,2%	115	64%	42	21%	162	80%	-2,9%	18	10%
5	201	182	76	38%	6,7%	106	58%	38	19%	159	79%	-4,1%	23	13%
Великі пожежі														
0	229	191	71	31%	11,6%	120	63%	55	24%	186	81%	-6,2%	5	3%
1	229	210	82	36%	2,1%	128	61%	60	26%	210	93%	2,3%	0	0%
2	226	194	75	33%	-1,9%	119	61%	52	23%	194	86%	12,8%	0	0%
3	225	189	66	29%	-2,4%	123	65%	58	26%	188	88%	13,7%	1	1%
4	223	193	69	31%	33,1%	124	64%	56	25%	177	79%	3,2%	16	8%
5	222	197	71	32%	13,3%	126	64%	56	25%	189	85%	22,3%	8	4%

Таблиця 3.6

Аналіз точності пожеж, проведених методом нейронних мереж

Лаг	Реальна кількість пожеж	Прогнозовані спалахи пожеж	Пояснені моделлю спалахи пожеж з точністю 1 доба		Середня різниця в амплітуді	Помилкові піки з точністю 1 доба		Не пояснені моделлю спалахи пожеж		Пояснені моделлю спалахи пожеж з точністю ±1 доба		Середня різниця в амплітуді	Помилкові піки з точністю ±1 доба	
1.	2.	3.	4.		5.	6.		7.		8.		9.	10.	
Малі пожежі														
0	207	204	88	43%	-7,4%	116	57%	39	19%	185	89%	0,6%	19	9%
1	206	187	70	34%	-7,4%	117	63%	54	26%	164	80%	6,6%	23	12%
2	204	196	75	37%	-3,4%	121	62%	40	20%	171	84%	-1,4%	25	13%
3	202	207	85	42%	-5,3%	122	59%	41	20%	185	92%	1,5%	22	11%
4	202	197	74	37%	-8,4%	123	62%	47	23%	175	87%	3,3%	22	11%
5	201	204	75	37%	7,7%	129	63%	49	24%	182	91%	-1,0%	22	11%
Великі пожежі														
0	229	176	57	25%	39,03%	119	68%	59	26%	175	76%	-7,63%	1	1%
1	229	201	82	36%	-2,92%	119	59%	54	24%	197	86%	11,56%	4	2%
2	226	198	68	30%	0,27%	130	66%	63	28%	196	87%	8,75%	2	1%
3	225	179	73	32%	22,74%	106	59%	48	21%	169	75%	6,52%	10	6%
4	223	193	68	30%	-9,68%	125	65%	60	27%	180	81%	2,75%	13	7%
5	222	186	54	24%	7,87%	132	71%	63	28%	178	80%	-2,58%	8	4%

Як видно з таблиць, розроблені моделі характеризуються високою точністю прогнозу в наближенні однієї доби (стовпець 4). Найбільша точність прогнозу невеликих пожеж спостерігається при lag=0 та 3 (також високою є точність для lag=5). У випадку великих пожеж найбільш точними виявились моделі з lag = 1. Моделі на основі ANFIS показують також високу точність для лагу 4 та 5. ANFIS моделі можуть передбачити до 39 % малих пожеж і 36% великих пожеж з точністю прогнозу в одну добу. Моделі на основі нейронних мереж показують більшу точність для прогнозу малих пожеж – 43%. У випадку великих пожеж точність залишається такою самою. Отже, ці результати підтверджують попередні висновки кореляційного аналізу в розрізі залежності від часової затримки. Тобто затримка між спалахом на сонці та спалахом малих пожеж становить 0 або 3 доби, що свідчить про наявність декількох механізмів, що призводять до спалаху лісових пожеж. Стосовно

великих пожеж – затримка становить 1 добу (підтверджено 3-ма моделями) та 4-5 діб (підтверджено 2-ма моделями).

Якщо ж розглядати точність в межах ±1 доби, то результати пронозу стають надзвичайно оптимістичними (стовпець 8): за допомогою ANFIS можна отримати прогноз до 87 % малих пожеж (lag = 2) і 93 % великих пожеж (lag = 1). Нейронні мережі показали знову більшу точність для малих пожеж: до 89% (lag = 0) та 92% (lag = 3). У випадку великих пожеж проноз дещо гірший: 86-87% (lag=1 та 2). Отже в цій точності прогнозу лише менш, ніж 21% (100% – колонка 8) спалахів лісових пожеж не залежить від активності сонця.

Слід зазначити, що якщо розглядати точність прогнозу в межах 1 доби то в середньому до 57-65% прогнозованих спалахів пожеж (для встановлених «точних» лагів) виявляються помилковими (колонка 6). Ці помилкові прогнози присутні як для великих, так і для малих пожеж. Тобто згідно прогнозу має настати пожежа, а насправді пожежі не сталось. Однак, більш важливою інформацією є скільки реальних пожеж розроблені моделі не в змозі передбачити. Щоб перевірити це, було підраховано кількість випадків, коли на графіку реальних пожеж спостерігалися піки, а на модельних графіках значення було нижче за середнє (колонка 7). Як показали розрахунки, тільки 19-26% реальних спалахів малих пожеж не можуть бути передбачені розробленими моделями. Для великих пожеж, це число складає приблизно 23-27%.

Однак, якщо точність передбачення складає ±1 добу, то кількість помилкових піків є меншою за 13 % для всіх розрахунків (10 стовпець). Також відсутні спалахи реальних пожеж, що неможливо передбачити.

Цікавою є інформація про прогнозовану амплітуду піків в порівнянні з реальними піками на рис.3.3. Тобто як співвідноситься кількість прогнозованих спалахів пожеж в конкретний день з реальною кількістю пожеж, зареєстрованими в цей самий день. Як показано в таблиці (5

колонка), у випадку невеликих пожеж амплітуда, зазвичай, є меншою в середньому на 5% , ніж фактичне число спалахів для ANFIS моделей. Нейронні мережі показують дещо гірший результат – -7%. Якщо прогноз зроблено в наближенні ±1 доба, то похибка інтенсивності є меншою: -4 – -2 % для ANFIS та в межах -1% – 3% для нейронних мереж (9 стовпець).

Для великих пожеж спостерігається інша ситуація. У випадку «точних» лагів (lag=1), спостерігається найменша похибка по амплітуді: від -2 % до 2 % (5 колонка). Для lag = 4 та 5 спостерігається дуже сильніше відхилення по амплітуді аж до 33% для ANFIS та -9% – 8% для нейронних мереж. У разі точності пронозу в наближенні ±1 доба, похибка амплітуди складе від -7 % до +22 %. Однак для «точних» лагів ця помилка складає: 2–3% для ANFIS та 3-11% для нейронних мереж.

Незважаючи на точність прогнозування, як по часу так і по амплітуді, ці моделі не дозволяють передбачати географічне положення джерел пожеж. Причина полягає у відсутності геопросторової інформації в навчальній вибірці. Цей недолік може бути усунутий, якщо долучити цю інформацію до бази даних.

3.1.2.3. Аналіз чутливості

Для визначення ступеня залежності кількості спалахів пожеж від зміни вхідних параметрів, був проведений аналіз чутливості. Оскільки результати моделювання методами ANFIS та нейронних мереж показали схожі результати, аналіз чутливості був проведений саме для ANFIS моделей для «точних» лагів. Для цього, значення всіх вхідних факторів усереднюються (табл. 3.1) і досліджується залежність кількості спалахів пожеж, що прогнозує модель, від послідовних змін кожного фактора. Результати цього аналізу представлені на рисунку 3.6.

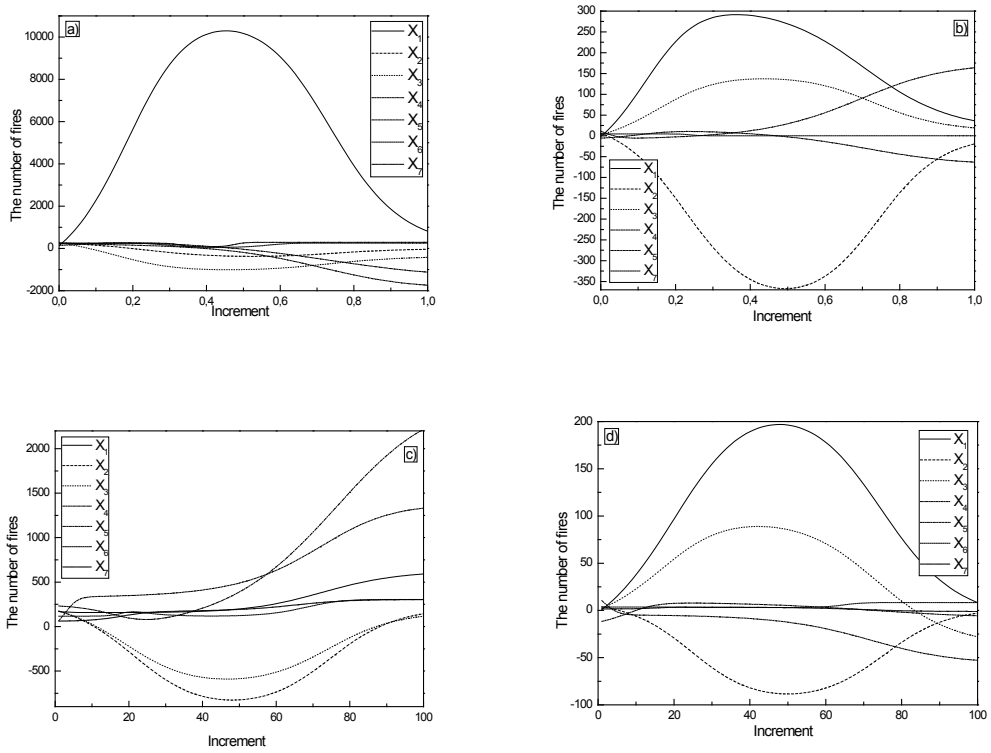


Рис 3.6. Чутливість малих (a – lag = 0, c – lag = 3) і великих (b – lag = 1, d – lag = 4) лісових пожеж від зміни X_i.

Як видно з рисунків, залежність кількості спалахів лісових пожеж від вхідних параметрів є нелінійною. Зокрема, невеликі пожежі є найбільш чутливими до X_1 (lag=0). Залежність від останнього параметру має квадратичну форму. Коли активність X_1 збільшується від середнього значення 0,008 до 0,5, прогнозована кількість пожеж стрімко зростає. Збільшення цього фактора від 0,5 до 1 призводить до їх зменшення. Це можна пояснити тим, що такого великого збільшення зазначеного параметру без зміни інших факторів ніколи раніше не спостерігалось. Тому діапазон 0,5–1 є неможливим у природі і висновки моделі по ньому можуть не братись до уваги. Параметри $X_2 - X_7$ практично не впливають на кількість невеликих пожеж. Зовсім інша ситуація спостерігається для lag = 3 (рис.3.6.c). Найбільш чутливим є параметр X_5: в діапазоні від 0 до 0,1. Від 0,1 до 0,5 цей фактор не впливає на пожежі. Після 0,5, збільшення цього

фактора знову призводить до різкого збільшення пожеж. Однак, після 0,6 найбільш впливовим параметром стає X_4.

Інша ситуація спостерігається для великих лісових пожеж. Залежності для lag = 1, 4 є схожими. Як видно з рис 3.6. (b, d), найбільш вагомими параметрами є X_1 та X_3, залежність кількості великих пожеж від їх почергової зміни є аналогічною для X_1 у випадку невеликих пожеж. Залежність від X_5 має експоненційну форму. Як видно з рис.3.6.b, кількість великих пожеж стрімко зростає, коли X_5 стає більшою за 0,5 (це характерно лише для lag = 1).

Така цікава поведінка потребує подальших експериментальних та теоретичних досліджень для підтвердження чи спростування отриманих висновків.

3.1.3. Лісові пожеж в США, Португалії та Греції

3.1.3.1. Результати моделювання

На основі моделі представлені в розділі 2 були проведено навчання серії гібридних нейронних мереж LSTM. На рис. 3.7 представлена динаміка зміни середньоквадратичної помилки тестового та навчального набору даних протягом навчання однієї з нейронних мереж. Як видно на графіку спостерігаються декілька піків збільшення помилки, це пояснюється переходом до наступного блоку навчання. Для визначення необхідної кількості епох та кількості нейронів досліджувалась динаміка зміни середньо-квадратичної похибки протягом періоду навчання. Навчання припинялось, коли графік помилки тестового набору починав рівномірно зростати протягом 10 епох навчання. Як видно з графіка, нейронна мережа доволі швидко адаптувалась до навчального набору і тривалий час до тестового. Що свідчить про необхідність глибокого навчання. Окрім того помилка тестового та навчального наборів суттєвою різняться між собою. Це свідчить про наявність достатньо складних функціональних залежностей.

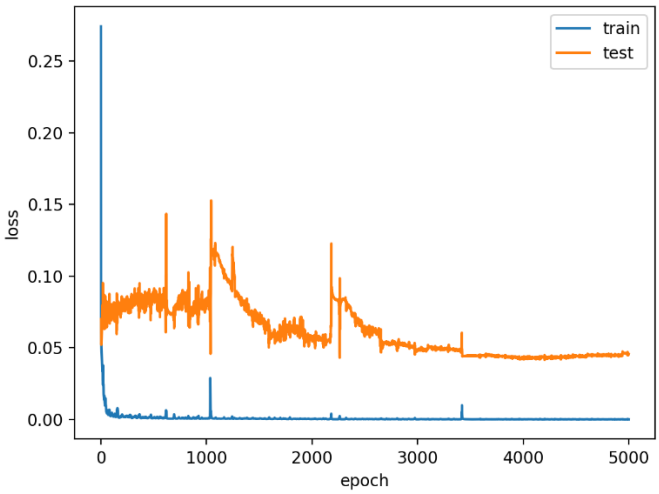


Рис. 3.7. Динаміка зміни середньо-квадратичної похибки протягом навчання для навчальної та тестової вибірок

На рисунках 3.8-3.12 приведені результати прогнозів даних 9 нейронних мереж. Як видно з рисунку, всі нейронні мережі ідеально добре підігнані до навчальних наборів даних. Однак на тестових наборах спостерігаються сильні флуктуації. Зокрема найгірше узгодження отримано у випадку Греції та Каліфорнії2. Найкращі результати прогнозування спостерігаються у випадку станцій Каліфорнія1 та Каліфорнія2. Також з графіків видно, що у випадку Каліфорнії2, Греції та Португалії відкритим питанням залишається проблема стійкості, а отже точності та адекватності різних моделей LSTM.

3.1.3.2. Аналіз точності ансамблю моделей LSTM

Для вирішення цієї проблеми були використані ансамблі LSTM моделей на основі Gradient Boosting (GB) for regression. GB створює адитивну модель поетапно; це дозволяє оптимізувати довільні диференційовані функції втрат. На кожному етапі дерево регресії підбирається на від'ємному градієнті заданої функції втрат (рис. 3.13). [127]. В якості критерія оптимізації була вибрана середньоквадратична помилка, кількість етапів посилення для виконання дорівнювала 100, швидкість навчання = 0.1, максимальна глибина індивідуальних оцінок регресії = 1.

У результаті навчання ансамблю регресійних моделей були отримані наступні результати (рис. 3.14-3.18).

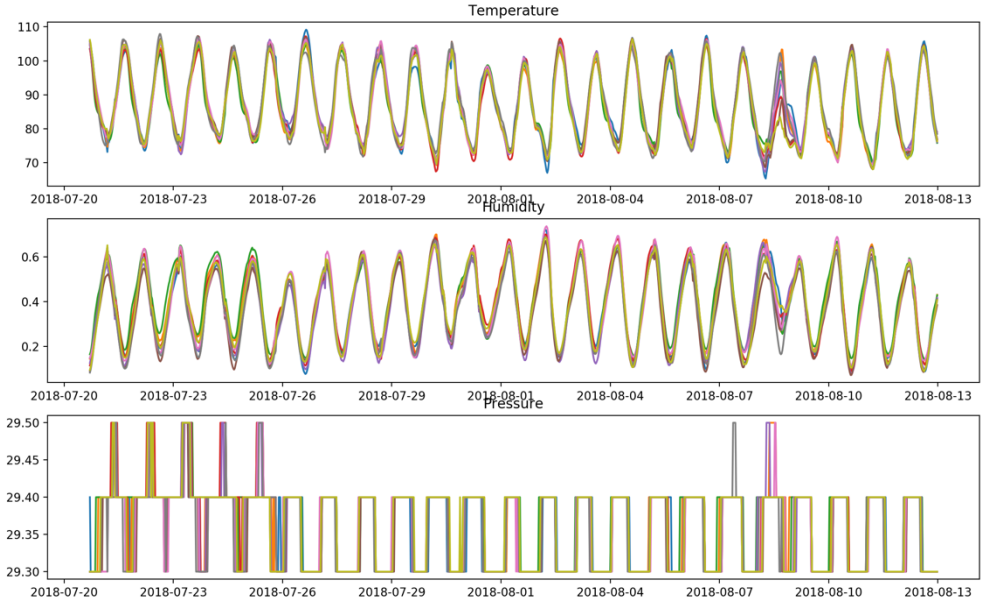


Рис.3.8. Результати прогнозування 9 LSTM рекурентних нейронних мереж в порівнянні з реальними даними для Каліфорнія1

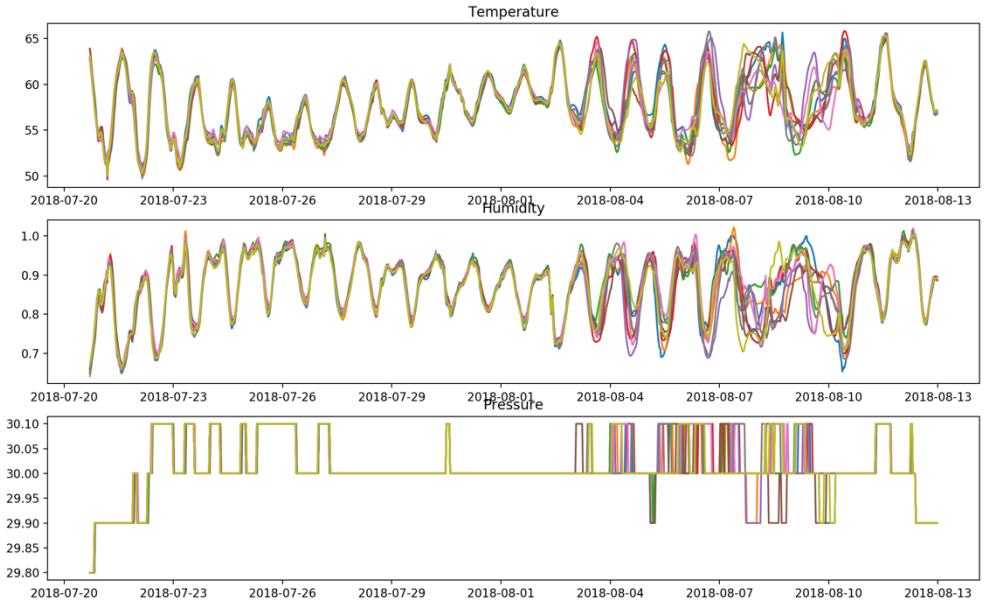


Рис.3.9. Результати прогнозування 9 LSTM рекурентних нейронних мереж в порівнянні з реальними даними для Каліфорнія2

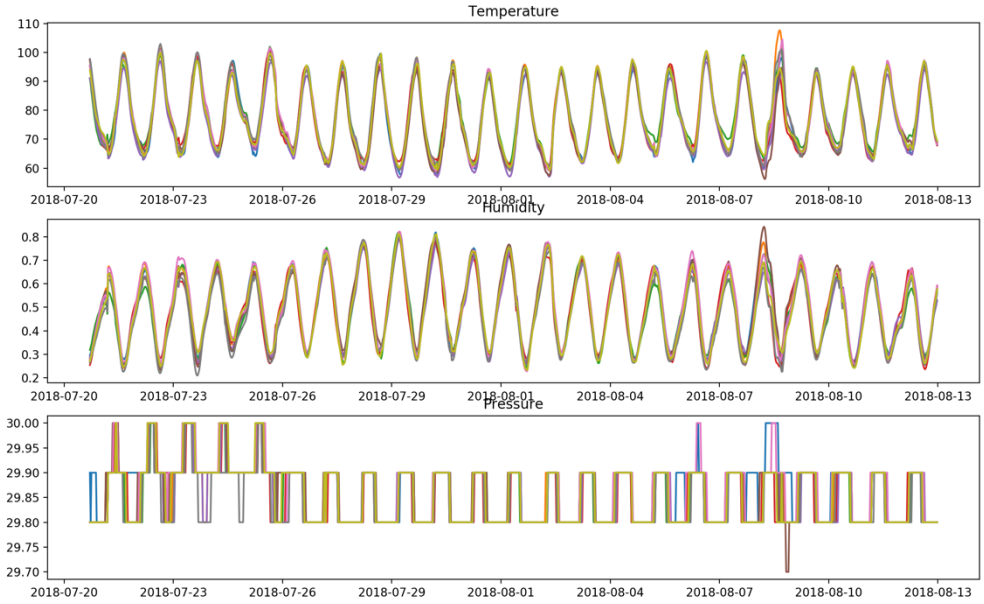


Рис.3.10. Результати прогнозування 9 LSTM рекурентних нейронних мереж в порівнянні з реальними даними для Каліфорнія3

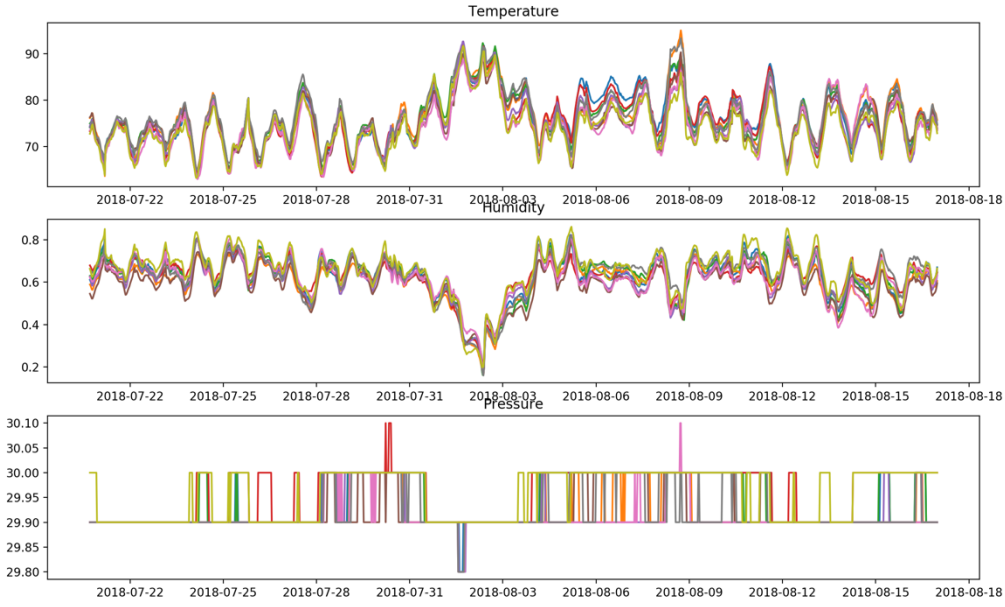


Рис.3.11. Результати прогнозування 9 LSTM рекурентних нейронних мереж в порівнянні з реальними даними для Португалії

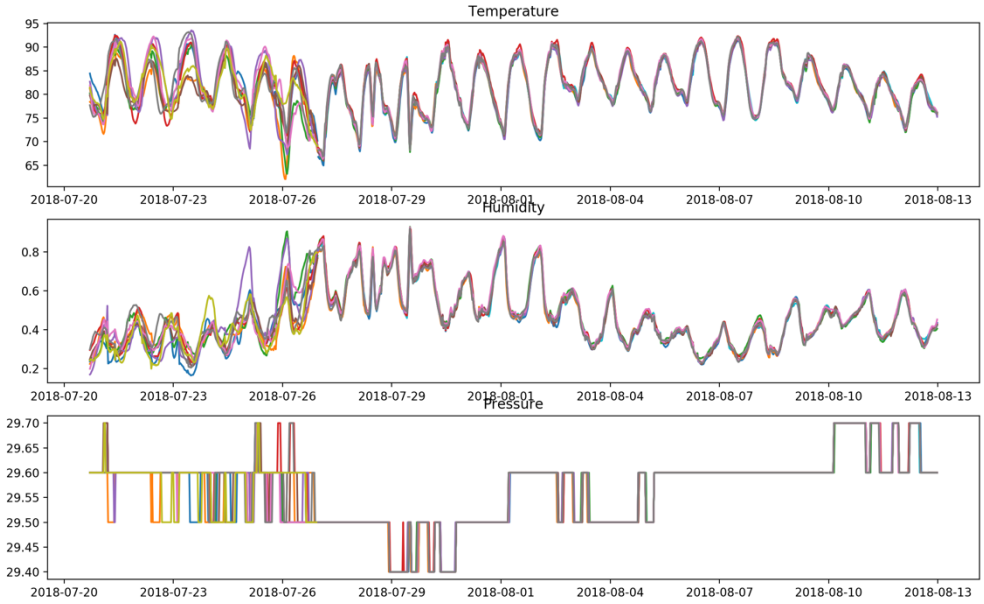


Рис.3.12. Результати прогнозування 9 LSTM рекурентних нейронних мереж в порівнянні з реальними даними для Греції

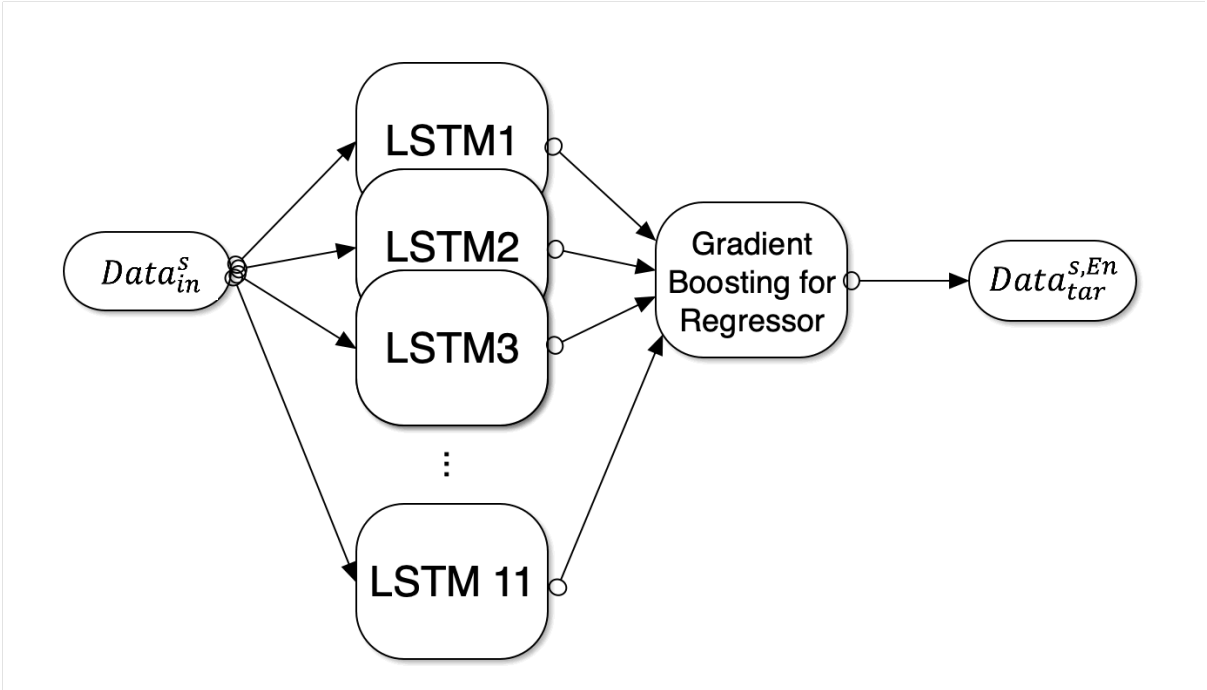


Рис.3.13. Ансамбль LSTM моделей

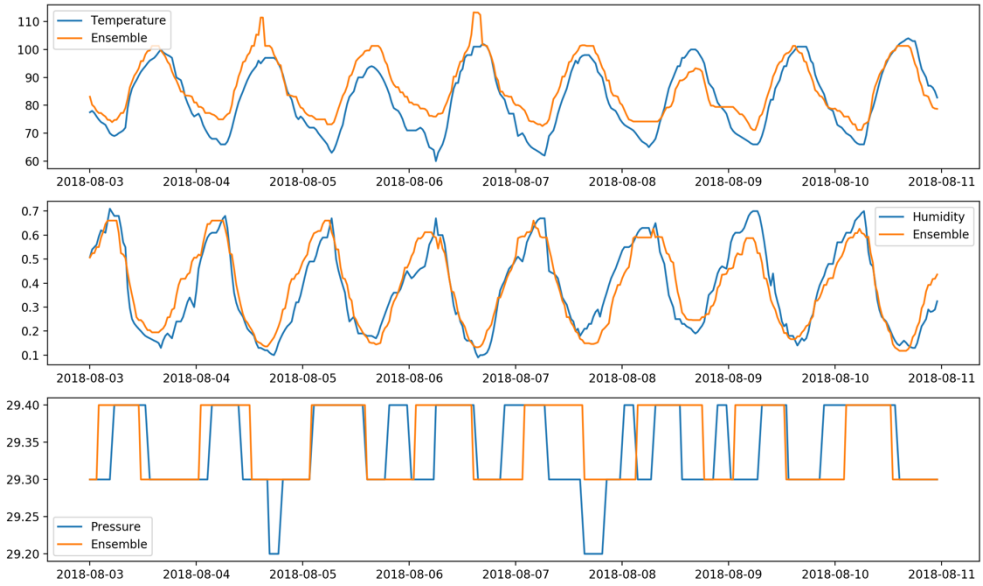


Рис 3.14. Результат прогнозування ансамблем LSTM тестових даних для Каліфорнія1

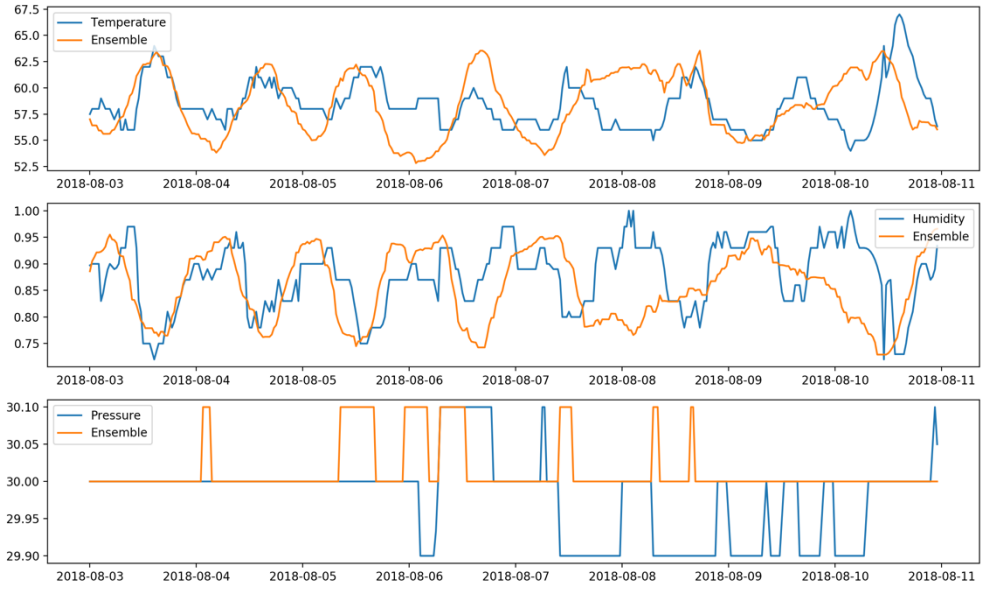


Рис 3.15. Результат прогнозування ансамблем LSTM тестових даних для Каліфорнія2

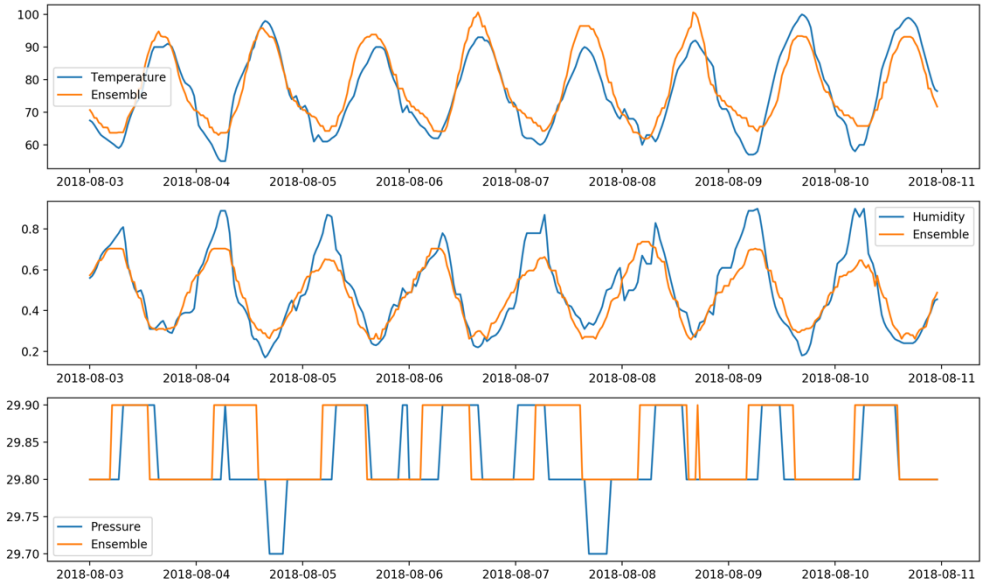


Рис 3.16. Результат прогнозування ансамблем LSTM тестових даних для Каліфорнія3

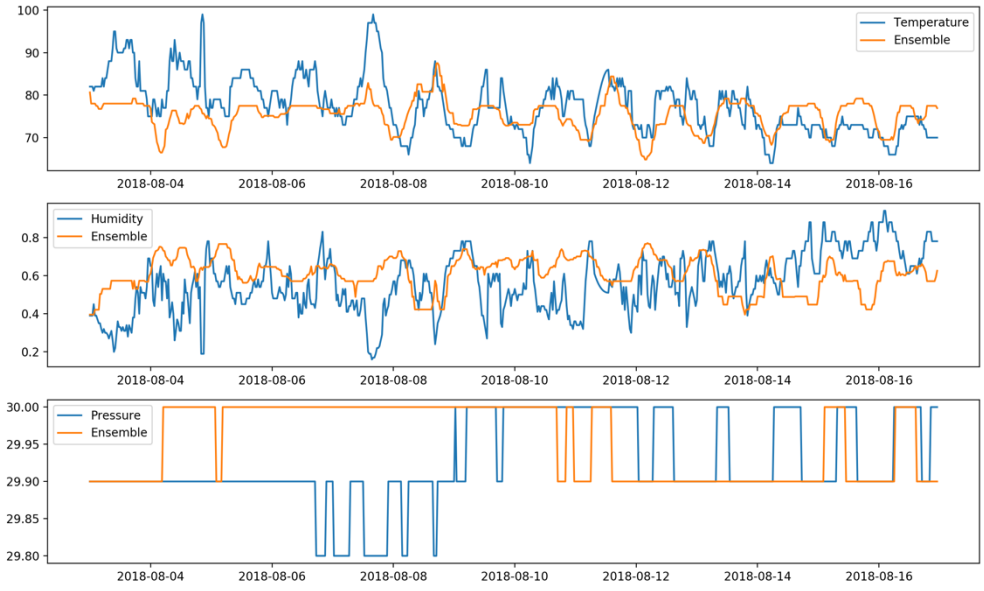


Рис 3.17. Результат прогнозування ансамблем LSTM тестових даних для Португалія

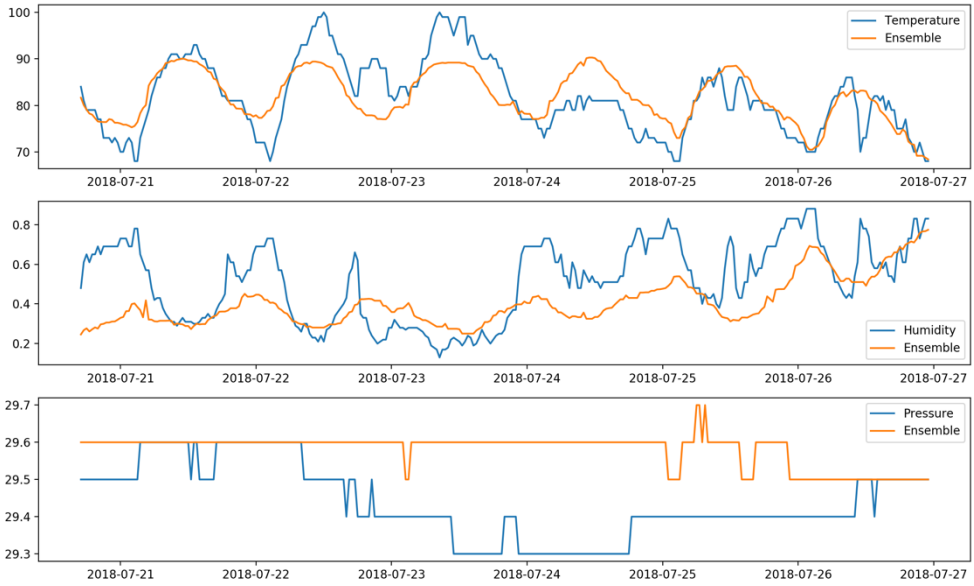


Рис 3.18. Результат прогнозування ансамблем LSTM тестових даних для Греція

Як видно з рисунків результати прогнозування ансамблю моделей є найкращими у випадку станцій Каліфорнія1 та 2. У випадку прогнозування тиску всі 5 ансамблів LSTM моделей показали поганий результат. Це пов’язано із дискретними значеннями вимірювання тиску. Тому для таких даних зручніше використовувати інші типи моделей. І це питання потребує подальшого дослідження. Для кількісної оцінки точності ансамблю моделей зручно провести кореляційний аналіз між рядами реальних даних та LSTM мереж як для кожної так і для ансамблю моделей. Слід зазначити, що кореляційний аналіз для дискретних даних не є коректним. Тому в таблиці 3.7. представлені лише дані для Температури та Вологості.

Таблиця 3.7

Коефіцієнти кореляції отриманих моделей

	LSTM1	LSTM2	LSTM3	LSTM4	LSTM5	LSTM6	LSTM7	LSTM8	LSTM9	Ensemble
Каліфорнія1										
T	0.90	0.93	0.90	0.89	0.93	0.86	0.91	0.95	0.86	**0.91**
H	0.87	0.93	0.92	0.89	0.93	0.89	0.92	0.91	0.89	**0.92**
Каліфорнія2										
T	0.29	0.37	0.37	0.29	0.31	0.17	0.40	0.36	0.24	**0.29**
H	0.27	0.38	0.49	0.28	0.27	0.24	0.36	0.39	0.16	**0.33**
Каліфорнія3										
T	0.93	0.90	0.91	0.92	0.94	0.92	0.92	0.93	0.92	**0.93**
H	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.91	**0.91**
Португалія										
T	0.44	0.35	0.42	0.48	0.57	0.40	0.39	0.42	0.45	**0.44**
H	0.04	0.00	0.06	0.19	0.10	0.09	0.07	0.17	-0.01	**0.04**
Греція										
T	0.71	0.62	0.70	0.70	0.81	0.58	0.70	0.59	0.67	**0.75**
H	0.63	0.49	0.55	0.59	0.69	0.63	0.50	0.56	0.52	**0.60**

Як видно з таблиці, найбільш точними моделями виявились Каліфорнія 1 та 3. Далі за точністю іде Греція. Прогнозні дані для Португалії та Каліфорнія2 абсолютно не корелюють із реальними даними. Це може свідчити, що для цих станцій отримані моделі є непридатними. І для побудови прогнозів потрібно враховувати зовсім інші фактори.

3.1.3.3. Аналіз чутливості

Отримані моделі дають змогу порахувати чутливість вихідних полів від впливу вхідних. Для цього для кожного запису j в $\widetilde{DF}_{LSTM,in}^{s,3D}$ розраховується значення ансамблю моделей $Data_{tar}^{s,En} = \{< \tilde{T}_{1,j}(t), .., \tilde{T}_{3,j}(t) >\}_{j=\overline{1,M}}$, де M – кількість записів DataFrame. Наступним кроком всі значення часових рядів факторів *i=1-15* в $\widetilde{DF}_{LSTM,in}^{s,3D}$ почергово збільшуються на 10%. Та розраховуються вихідні значення: $Data_{tar,i}^{s,En} = \{< \tilde{T}_{1,j}(t), .., \tilde{T}_{3,j}(t) >\}^{i}{}_{j=\overline{1,M};i=\overline{1,15}}$. Наступним кроком розраховується матриця відносних змін для кожного запису:

$$Se = \{se_j^i\}_{i=\overline{1,15},j=\overline{1,M}} = \{\frac{Data_{tar,i}^{s,En} - Data_{tar}^{s,En}}{Data_{tar}^{s,En}}\}_{i=\overline{1,15}} \quad (3.5)$$

де i – індекс вхідного параметру, j – індекс запису.

Після чого знаходиться вектор усередненого вплив кожного фактора по всім записам $Data_{tar}^{s,En}$:

Таблиця 3.8

Аналіз чутливості температури, вологості та атмосферного тиску від факторів сонячної активності (%) для Каліфорнія1

Input factor	**> 30 MeV**	**38-53**	**175-315**	**47-68**	**115-195**	**310-580**	**795-1193**
Temperature	0.59%	0.00%	0.00%	-0.01%	0.01%	-0.12%	0.01%
Humini dy	3.25%	0.00%	0.00%	0.11%	0.09%	0.31%	-0.02%
Input factor	**1060-1900**	**PROTON DENSITY**	**BULK SPEED**	**ION TEMPERATURE**	**10.7 cm Radio Flux**	**Temperature**	**Humidity**
Temperature	0.00%	-0.06%	0.09%	0.00%	0.25%	0.75%	-0.03%
Humini dy	-0.01%	-0.34%	-0.23%	0.03%	-0.96%	2.55%	-1.10%

Таблиця 3.9

Аналіз чутливості температури, вологості та атмосферного тиску від факторів сонячної активності (%) для Каліфорнія2

Input factor	> 30 MeV	38-53	175-315	47-68	115-195	310-580	795-1193
Temperature	-0.06%	0.00%	0.00%	0.03%	0.03%	-0.02%	0.00%
Humidity	0.07%	0.00%	0.00%	-0.07%	-0.02%	-0.13%	-0.07%
Input factor	1060-1900	PROTON DENSITY	BULK SPEED	ION TEMPERATURE	10.7 cm Radio Flux	Temperature	Humidity
Temperature	0.00%	-0.04%	-0.12%	0.00%	-0.03%	0.27%	0.33%
Humidity	0.00%	0.23%	0.18%	0.00%	-0.46%	-0.21%	0.10%

Таблиця 3.10

Аналіз чутливості температури, вологості та атмосферного тиску від факторів сонячної активності (%) для Каліфорнія3

Input factor	> 30 MeV	38-53	175-315	47-68	115-195	310-580	795-1193
Temperature	0.38%	0.00%	0.00%	0.00%	0.08%	-0.09%	0.00%
Humidity	1.65%	0.00%	0.01%	0.08%	0.01%	0.22%	-0.04%
Input factor	1060-1900	PROTON DENSITY	BULK SPEED	ION TEMPERATURE	10.7 cm Radio Flux	Temperature	Humidity
Temperature	0.00%	-0.02%	0.06%	0.00%	0.48%	0.34%	-0.01%
Humidity	0.01%	-0.07%	0.14%	0.05%	-1.26%	2.01%	0.85%

Таблиця 3.11

Аналіз чутливості температури, вологості та атмосферного тиску від факторів сонячної активності (%) для Португалія

Input factor	> 30 MeV	38-53	175-315	47-68	115-195	310-580	795-1193
Temperature	0.35%	0.00%	0.00%	0.02%	0.04%	0.08%	-0.01%
Humidy	1.57%	0.00%	0.00%	0.03%	0.02%	-0.53%	0.00%
Input factor	1060-1900	PROTON DENSITY	BULK SPEED	ION TEMPERATURE	10.7 cm Radio Flux	Temperature	Humidity
Temperature	0.00%	-0.01%	-0.10%	-0.03%	-0.14%	0.33%	0.06%
Humidy	-0.01%	0.00%	0.32%	0.10%	0.71%	0.26%	1.62%

Таблиця 3.12

Аналіз чутливості температури, вологості та атмосферного тиску від факторів сонячної активності (%) для Греція

Input factor	> 30 MeV	38-53	175-315	47-68	115-195	310-580	795-1193
Temperature	0.30%	0.00%	0.00%	0.01%	0.02%	-0.01%	0.00%
Humidy	0.04%	0.00%	0.01%	-0.03%	-0.08%	0.10%	0.06%
Input factor	1060-1900	PROTON DENSITY	BULK SPEED	ION TEMPERATURE	10.7 cm Radio Flux	Temperature	Humidity
Temperature	0.00%	-0.05%	-0.23%	0.00%	0.42%	0.01%	0.37%
Humidy	0.07%	0.01%	1.06%	0.15%	-3.89%	0.02%	1.31%

$$\overline{Se} = \sum_{j=1}^{M} se_j^i \quad (3.6)$$

Результати такого аналізу приведені в таблиці 3.8-12.

Як видно з таблиць для Каліфорнія 1 та 3 сонячна активність найбільше впливає на вологість. Так збільшення фактору > 30 MeV на 10% призведе до зростання вологості на 3.25% і 1.65% на станціях Каліфорнія 1 та 3 відповідно. Крім того підвищення температури на 10% призведе до зростання вологості на 2.55% та 2.02% відповідно. Температура є менш чутливою до змін факторів. Так при зростанні > 30 MeV на 10% температура в Каліфорнії1 зросте лише на 0.59% і на 0.38% в Каліфорнії 3. Також значення температури залежить від попередніх своїх значень. Так зростання останніх на 10% вплине на зростання в Каліфорнії 1 на 0.75% і 0.34%. Також видно Що вологість залежить від зміни температури, а зворотного впливу не спостерігається

Аналіз чутливості для тиску не проводився, так як даний тип моделей виявився не адекватним до цих даних.

3.2. Урагани

3.2.1. Результати парадельних розрахунків

Найкращий спосіб описати, наскільки змодельовані дані відповідають реальним даним, — це нанести їх на єдиний графік для кожного урагану, який був предметом цього дослідження. Результати розрахунків для лінійних моделей і штучних нейронних мереж представлені на рисунку 3.19.

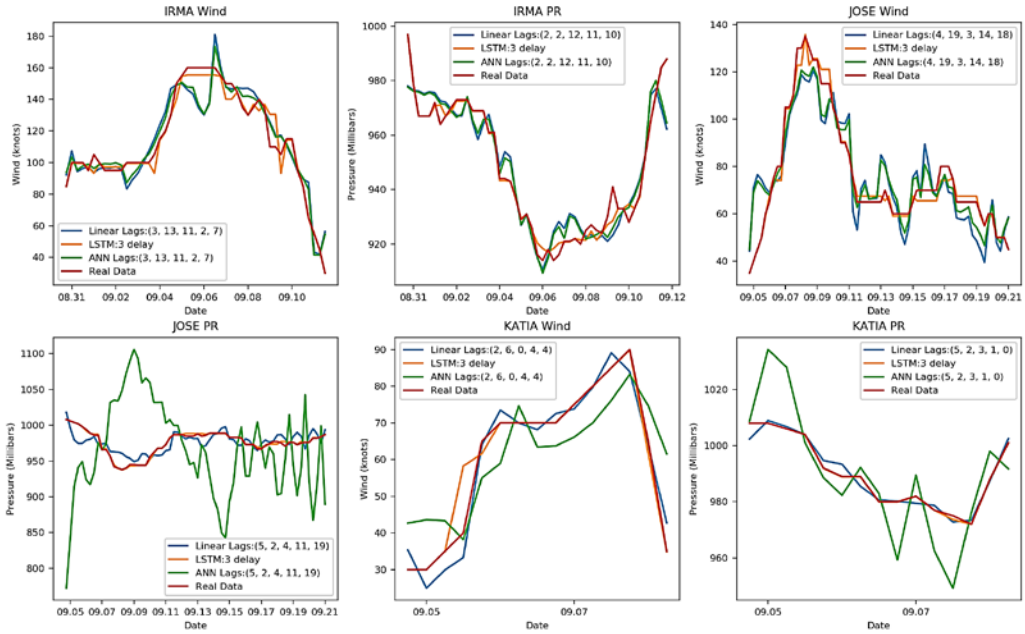


Рис. 3.19. Результати прогнозування ураганів за допомогою лінійних моделей і штучних нейронних мереж для: (а) швидкості вітру урагану IRMA, (б) тиску урагану IRMA, (в) швидкості вітру урагану JOSE, (г) тиску урагану JOSE , (е) Швидкість вітру урагану KATIA, (f) Тиск урагану KATIA

3.2.2. Аналіз точності

Як видно з рисунку, у більшості випадків моделі LSTM показують найкращий результат прогнозу для всіх шести досліджуваних цільових векторів. Як було підписано раніше, лише 4-годинний лаг використовувався в розрахунках у LSTM. Це означає, що поведінка вхідних факторів у попередні моменти часу відіграє ключову роль у прогнозуванні ураганів. Результати для штучних нейронних мереж і лінійних моделей подібні для ураганів IRMA та JOSE (тобто швидкість вітру). Як видно з малюнків 4d–4f, нейронні мережі показують значно гірші результати, ніж LSTM та лінійні моделі. У випадку з ураганом KATIA дивним є також відставання оптимальної моделі для поля RadioFlux, яке дорівнює 4 і нулю. Це абсолютно не узгоджується з попереднім аналізом. Це можна пояснити

малим розміром набору даних (15 записів) і, відповідно, неможливістю адекватного навчання як лінійних, так і нейронних мереж. Щодо ураганів IRMA та JOSE, отримані лаги добре узгоджуються з попереднім аналізом для поля RadioFlux , яке є найвпливовішим (як було показано вище). Кількісне порівняння точності результатів наведено в табл. 3.13.

Таблиця 3.13

Лаги та коефіцієнти кореляції отриманих моделей

Ураган	Параметр	Модель		Номери тестових моделей	Лаги	R^2 Повний набір даних	R^2 Перехресна перевірка
		Рівняння	Тип				
IRMA	Швидкість вітру	$F_1(X, L_1, \Omega_1^{Lin})$	Лінійний	1 048 576	L_1 =(3, 13, 11, 2, 7)	0,89	0,85
		$\{F_1(X, L_1, \Omega_1^{ANN})\}$	ANN	99		0,89	0,75
		$\{F_1(X_{3D,L}, L_{LSTM}, \Omega_1^{LSTM})\}$	LSTM	99	$L_{LSTM} = \{l_i = \overline{1,4}\}_{i=\overline{1,6}}$	0,98	0,88
	Тиск	$F_2(X, L_2, \Omega_2^{Lin})$	Лінійний	759,375	L_2 =(2, 2, 12, 11, 10)	0,90	0,88
		$\{F_2(X, L_2, \Omega_2^{ANN})\}$	ANN	99		0,90	0,87
		$\{F_2(X_{3D,L}, L_{LSTM}, \Omega_1^{LSTM})\}$	LSTM	99	$L_{LSTM} = \{l_i = \overline{1,4}\}_{i=\overline{1,6}}$	0,99	0,93
JOSE	Швидкість вітру	$F_3(X, L_4, \Omega_4^{Lin})$	Лінійний	5,153,632	L_3 =(4, 19, 3, 14, 18)	0,86	0,77
		$\{F_3(X, L_4, \Omega_4^{ANN})\}$	ANN	99		0,86	0,74
		$\{F_3(X_{3D,L}, L_{LSTM}, \Omega_1^{LSTM})\}$	LSTM	99	$L_{LSTM} = \{l_i = \overline{1,4}\}_{i=\overline{1,6}}$	0,98	0,61
	Тиск	$F_4(X, L_4, \Omega_4^{Lin})$	Лінійний	5,153,632	L_4 =(5, 2, 4, 11, 19)	0,69	0,56
		$\{F_4(X, L_4, \Omega_4^{ANN})\}$	ANN	99		0,58	0,70
		$\{F_4(X_{3D,L}, L_{LSTM}, \Omega_1^{LSTM})\}$	LSTM	99	$L_{LSTM} = \{l_i = \overline{1,4}\}_{i=\overline{1,6}}$	0,98	0,45
KATIA	Швидкість вітру	$F_5(X, L_5, \Omega_5^{Lin})$	Лінійний	100 000	L_5 =(2, 6, 0, 4, 4)	0,98	0,96
		$\{F_5(X, L_5, \Omega_5^{ANN})\}$	ANN	99		0,72	0,34
		$\{F_5(X_{3D,L}, L_{LSTM}, \Omega_1^{LSTM})\}$	LSTM	99	$L_{LSTM} = \{l_i = \overline{1,4}\}_{i=\overline{1,6}}$	0,95	0,48
	Тиск	$F_6(X, L_6, \Omega_6^{Lin})$	Лінійний	59 049	L_6 =(5, 2, 3, 1, 0)	0,98	0,96
		$\{F_6(X, L_6, \Omega_6^{ANN})\}$	ANN	99		0,65	0,53
		$\{F_5(X_{3D,L}, L_{LSTM}, \Omega_1^{LSTM})\}$	LSTM	99	$L_{LSTM} = \{l_i = \overline{1,4}\}_{i=\overline{1,6}}$	0,99	0,38
Всього			Лінійний	12,274,264			
			ANN	594			
			LSTM	594			

Як видно з таблиці, найбільший коефіцієнт кореляції спостерігається для моделей LSTM у всіх прогнозах. Тестування перехресної перевірки підтвердило, що ці моделі є точними та адекватними у випадку урагану IRMA. Невеликий коефіцієнт перехресної перевірки для JOSE можна пояснити випадковими факторами, які не були враховані. Погані результати тесту Каті пояснюються занадто малими даними для LSTM.

Для лінійних моделей і моделей ШНМ таблиця 5 показує, що найвищі коефіцієнти кореляції отримані для цільових векторів, таких як швидкість вітру урагану IRMA, тиск урагану IRMA та швидкість вітру урагану JOSE. Коефіцієнти детермінації для лінійних моделей і нейронних мереж збігаються. Результати перехресної перевірки трохи нижчі, але вони також мають високі значення. Це також підтверджує адекватність цих моделей. З таблиці також видно, що тиск урагану JOSE має низькі значення коефіцієнтів кореляції як для лінійних моделей, так і для нейронних мереж. Тому точність цієї моделі низька. Стосовно урагану KATIA слід зазначити, що, як і на графіках, результати є точними для лінійних моделей і низькими для нейронних мереж, що може бути викликано малим обсягом навчальної вибірки.

Під час розрахунків отримано наступні оптимальні лінійні моделі:

$$F_1\left(X, L_1, \Omega_1^{Lin}\right) = -16.44 - 1.09 \cdot x(3)_1 + 2.88 \cdot 10^{-04} \cdot x(13)_2 - 0.05 \cdot x(11)_3 + 0.85 \cdot x(2)_4 + 1.40 \cdot x(7)_5, \quad (3.7)$$

$$F_2\left(X, L_2, \Omega_2^{Lin}\right) = 1067.52 + 0.55 \cdot x(2)_1 - 5.42 \cdot 10^{-04} \cdot x(2)_2 + 0.02 \cdot x(12)_3 + 0.63 \cdot x(11)_4 - 1.17 \cdot x(10)_5, \quad (3.8)$$

$$F_3\left(X, L_3, \Omega_3^{Lin}\right) = -80.15 - 0.71 \cdot x(4)_1 + 4.93 \cdot 10^{-04} \cdot x(19)_2 + 0.12 \cdot x(3)_3 + 1.62 \cdot x(14)_4 + 0.84 \cdot x(18)_5, \quad (3.9)$$

$$F_4\left(X, L_2, \Omega_2^{Lin}\right) = 1073.42 + 0.54 \cdot x(5)_1 - 2.83 \cdot 10^{-04} \cdot x(2)_2 - 0.08 \cdot x(4)_3 - 1.27 \cdot x(11)_4 - 0.52 \cdot x(19)_5, \quad (3.10)$$

$$F_5\left(X, L_5, \Omega_5^{Lin}\right) = -413.61 - 94.62 \cdot x(2)_1 - 8.08 \cdot 10^{-04} \cdot x(6)_2 + 0.17 \cdot x(0)_3 - 1.88 \cdot x(4)_4 + 3.14 \cdot x(4)_5, \quad (3.11)$$

$$F_6\left(X, L_6, \Omega_6^{Lin}\right) = 783.42 - 26.24 \cdot x(5)_1 + 1.42 \cdot 10^{-04} \cdot x(2)_2 + 0.12 \cdot x(3)_3 - 2.30 \cdot x(1)_4 + 1.19 \cdot x(0)_5. \quad (3.12)$$

де в квадратних дужках вхідних параметрів вказано значення лага.

Результатом навчання нейронних мереж є 108 нейронних мереж, параметри яких змінюються під час навчання, тому виводити таку кількість динамічних матриць вагових коефіцієнтів нейронів недоцільно.

Як видно з табл. 3.14, загальна кількість перевірених лінійних моделей за рахунок використання запропонованого алгоритму зменшилася з 424 798 638 до 12 274 264 · 11 = 135 016 904, тобто загальна кількість моделей зменшена в 3 рази і становить 32% від попередній показник. Враховуючи, що час розрахунку становив 4,5 години на Mac Book Pro (2015) (табл. 3.14), це заощадило приблизно 3 години комп'ютерного часу.

Таблиця 3.14

Інструменти в експериментальних середовищах

Пункт	Інструмент
Операційна система	macOS Sierra 10.13.2
комп'ютер	MacBook Pro (Retina, 15 дюймів, середина 2015 р.)
Процесор	2,5 ГГц Intel Core i7
Оперативна пам'ять	16 ГБ 1600 МГц DDR3
Мова програмування середовища	Python 3.5.1 на Darwin

Використання нейронних мереж в рамках такого алгоритму займає на порядки більше часу і вимагає, відповідно, залучення комп'ютерного кластера.

3.2.3. Аналіз чутливості

Щоб перевірити адекватність моделей, було проведено аналіз чутливості моделі до зміни факторів для всіх моделей у таблиці 5 [128]. Аналіз був наступним. Для кожного кортежу *r* вектора вхідних параметрів $X^r = \{x_j^r\}_{\substack{j=\overline{1-5} \\ r=\overline{1-N}}}$ (для моделей LSTM x_j^r – це вектор значень протягом лагу *L*) , який складається з *N* записів, значення вхідних параметрів було збільшено на 10% і зміна відповідної моделі $F_{i=\overline{1-6}}$або набору моделі розраховували методом Delphi (у випадку нейронних мереж). Потім усі отримані значення усереднювали. Отримане значення означає середню зміну швидкості вітру або тиску конкретного урагану при збільшенні вхідного параметра на 10%.

Для реалізації цього створено діагональну матрицю варіаційних коефіцієнтів розмірністю, що дорівнює числу вхідних параметрів, у нашому випадку п’яти:

$$V = \begin{bmatrix} 0.1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0.1 \end{bmatrix}_{5\times5}. \tag{3.13}$$

У Python це можна реалізувати шляхом виконання команд *zeros* , які *fill_diagonal* бібліотек *NumPy* :

V = numpy.zeros ((5, 5), float)

numpy.fill _diagonal (V, 0,1).

Кожен кортеж вектора вхідних параметрів дублюється по вертикалі в кількості, яка дорівнює довжині кортежу (тобто кількості вхідних параметрів) за допомогою функції *repmat* ($X^r, 5, 1$) бібліотеки *NumPy.matlib* :

$$A^r = \begin{bmatrix} x_1^r & \cdots & x_5^r \\ \vdots & \ddots & \vdots \\ x_1^r & \cdots & x_5^r \end{bmatrix}. \tag{3.14}$$

Матриця тестових значень розраховується як елементний добуток матриць:

$$T^r = (V+1)\cdot A^r = \begin{bmatrix} 1.1\cdot x_1^r & \cdots & 1.0\cdot x_5^r \\ \vdots & \ddots & \vdots \\ 1.0\cdot x_1^r & \cdots & 1.1\cdot x_5^r \end{bmatrix}. \quad (3.15)$$

Розраховується вектор значень:

$$S_i^r = F_i\left(T^r, L_i, \Omega_i^{Lin(ANN)}\right) = \begin{bmatrix} f_{i,x_1}^r \\ \vdots \\ f_{i,x_5}^r \end{bmatrix}. \quad (3.16)$$

Масив отриманих змін функцій F_i формується шляхом оцінки значень S_i^r для всіх кортежів вектора X:

$$S_i = \begin{bmatrix} (S_i^1)^T \\ \vdots \\ (S_i^N)^T \end{bmatrix}. \quad (3.17)$$

Потім розраховується вектор прогнозованих моделлю значень і дублюється по горизонталі на кількість полів введення:

$$M_i = \{m_i^r\}_{r=\overline{1-N}} = F_i\left(X, L_i, \Omega_i^{Lin(ANN)}\right), \quad (3.18)$$

$$Mx_i = \begin{bmatrix} m_i^1 & \cdots & m_i^1 \\ \vdots & \ddots & \vdots \\ m_i^N & \cdots & m_i^N \end{bmatrix}_{N\times 5}. \quad (3.19)$$

Останнім кроком є побудова матриці відносних змін шляхом обчислення елементарної різниці та ділення матриць Mx_iі S_i. Потім проводиться усереднення по стовпцях:

$$D = (S_i - Mx_i)/Mx_i, \quad (3.20)$$

$$Sens = \overline{D}_{col}. \quad (3.21)$$

Результати розрахунків наведені в табл. 3.15.

Як видно з таблиці, ми отримали абсолютно різні результати між моделями LSTM і лінійні з ANN. Це можна легко пояснити різним підходом до врахування лагової поведінки вхідних факторів. Крім того, для цих моделей використовувався різний часовий лаг. Тому аналіз чутливості слід проводити окремо для цих моделей.

Таблиця 3.15

Аналіз чутливості отриманих моделей

Ураган	Параметр	Модель	P > 100	E > 2,0	швидкість	Щільність	Radio Flux 10.7
IRMA	Швидкість вітру	Лінійна	–0,63%	0,10%	–2,51%	0,23%	14,38%
		ANN	–0,65%	0,13%	–2,64%	0,18%	13,05%
		LSTM	-12,76%	-8,94%	-28,36%	40,35%	0,46%
	Тиск	Лінійна	0,02%	–0,04%	0,09%	0,02%	–1,36%
		ANN	0,02%	–0,04%	0,09%	0,02%	–1,36%
		LSTM	0,15%	-0,99%	-1,77%	-0,64%	0,20%
JOSE	Швидкість вітру	Лінійна	–0,26%	0,50%	9,27%	0,64%	11,27%
		ANN	–0,26%	0,50%	9,27%	0,64%	11,27%
		LSTM	-0,03%	-18,31%	1,89%	-4,21%	123,51%
	Тиск	Лінійна	0,01%	–0,04%	–0,42%	–0,04%	–0,53%
		ANN	0,00%	0,59%	3,63%	0,43%	5,24%
		LSTM	-0,02%	-0,43%	-6,93%	3,99%	-44,84%
KATIA	Швидкість вітру	Лінійна	–1,07%	–1,19%	17,69%	–1,17%	74,57%
		ANN	0,00%	–1,30%	8,80%	–0,64%	3,33%
		LSTM	4,77%	24,89%	547,95%	207,05%	0,80%
	Тиск	Лінійна	–0,02%	0,05%	0,66%	–0,07%	1,46%
		ANN	0,00%	–0,14%	2,65%	0,50%	6,98%
		LSTM	-3,33%	1,47%	7,15%	-9,92%	-8,46%

Як було підписано раніше, найкращі моделі були отримані для урагану IRMA моделями LSTM. Аналіз чутливості показує, що збільшення щільності протягом 4 годин на 10% призведе до збільшення швидкості вітру на 40%. Збільшення таких факторів, як P > 100, E > 2.0 і Speed призведе до зменшення швидкості вітру. Також цей розрахунок показує, що сонячна активність слабо впливає на тиск.

Фактором найбільшої чутливості для Jose є Radio Flux 10.7. Збільшення цього коефіцієнта на 10% призведе до збільшення швидкості на 123% і зниження тиску на 45%. Ці великі цифри можна пояснити поганою адекватністю моделі LSTM для цього урагану. Подібна ситуація спостерігається і для ураганів KATIA.

Для моделей Linear та ANN, як видно з таблиці 7, фактором, який має найбільший вплив на швидкість вітру ураганів, є Radio Flux 10.7. Його збільшення на 10% призводить до зростання швидкості вітру для урагану IRMA в середньому на 13%–14% за 42 години (лаг 7) і на 11% за 4,5 дня (лаг 18) для JOSE. Як видно з таблиці, показники лінійних моделей і нейронних мереж є достатньо близькими для всіх факторів і цих ураганів, що підтверджує адекватність моделей. Другий важливий показник – швидкість SW. Його збільшення на 10% збільшує швидкість урагану JOSE на 9% через 18 годин (затримка 3) і зменшує швидкість урагану IRMA на 2,5% через 3 дні. Інші фактори не впливають на ці два урагани.

Для Katia hurricane Radio Flux 10.7 становить 74% для лінійних моделей і лише 3% для нейронних мереж. Сильна різниця в чутливості нейромережевих і лінійних моделей також ставить під сумнів їх адекватність. Це може бути викликано невеликою кількістю даних, що завадило побудові адекватної моделі.

Як відомо, першопричиною вітру є перепад тиску, тому цікаво проаналізувати вплив параметрів СВ на тиск повітря. Якщо проаналізувати чутливість тиску для ураганів IRMA та JOSE, то можна побачити, що вони

менш чутливі до змін ПВ. Зокрема, зміна Radio Flux 10.7 на 10% викликає падіння тиску на 1,3% через 2,5 дні для урагану IRMA і практично не впливає на тиск урагану JOSE. Проте, як видно з рисунку 2, зазначений параметр зріс з 28 серпня по 4 вересня 2017 року з 82,4 до 140, тобто на 70%. Згідно з таблицею 7, зміна тільки одного з цих факторів повинна була викликати зміну тиску в зоні урагану на 0,7 / 0,1 · (–1,3%) = –9,5%, тобто від 1004 мб до 908 мб . Реальний зареєстрований тиск становив 914 мб (похибка прогнозу 0,6%). Для урагану JOSE розрахована зміна становить 971 мб, зареєстрована – 938 мб (похибка прогнозу – 3,5%). Таким чином, незважаючи на низьку чутливість тиску до зміни параметрів СВ, сильні коливання вхідних параметрів можуть викликати різке зниження тиску, а отже, і виникнення ураганів.

3.2.4. Прогнозування на основі піків

В результаті навчання на даних урагану Irma отримані такі результати (табл. 3.16). Слід зазначити, що для всіх 4 методів результати виявились однаковими, що свідчить про стабільність розрахунків.

Таблиця 3.16

Матриця помилок для навчального набору даних на урагані Irma

		Predicted	
		0	1
Actual	0	529	0
	1	4 (40%)	6 (60%)

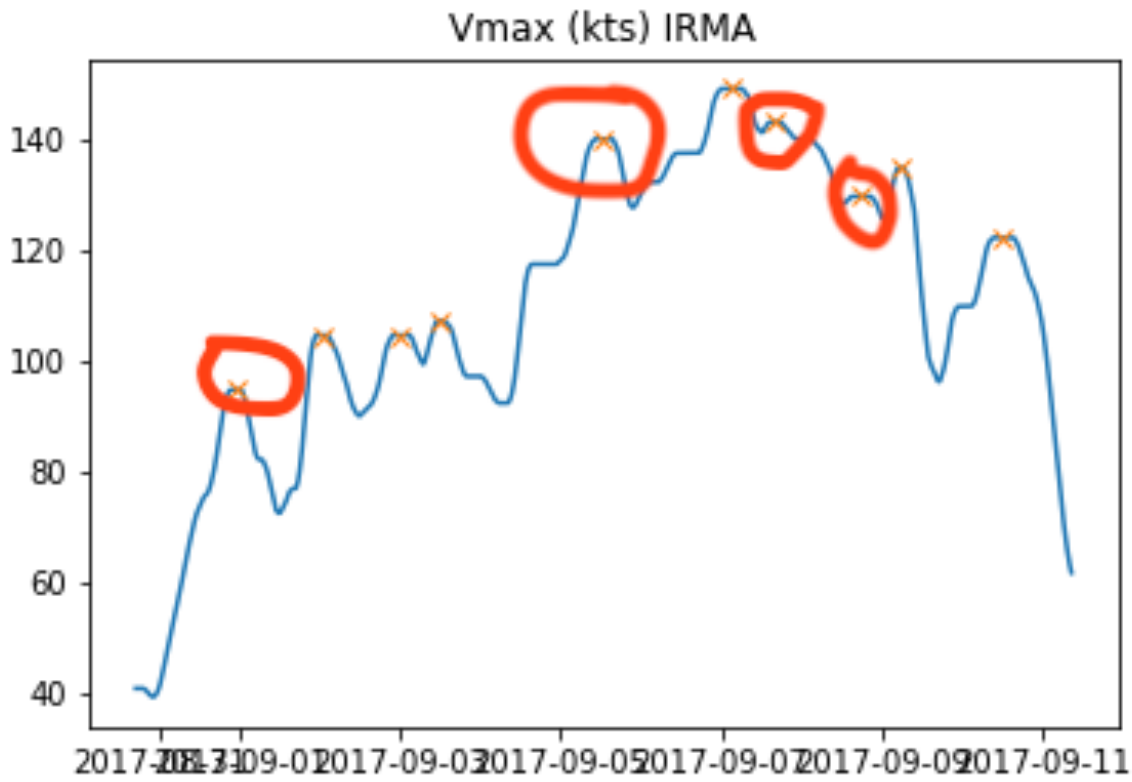


Рис.3.20. Нерозпізнані піки для урагану Irma

Як видно з таблиці, на навчальному наборі даних класифікатори жодного разу не помилились з відсутністю піків і 4 рази не змогли класифікувати піки в моменти часу: 31.08, 05.09, 07 та 08.09. (рис. 3.20). Це складає 40% всіх наявних піків.

Як видно з рисунку, останні 2 нерозпізнаних піки є невеликими і можливо не пов’язані із сонячним вітром. 2 перші є досить сильними і можливо є свідченнями або некоректності моделей, або спричинені іншими факторами, які не враховані в цій моделі. Для перевірки цього, навчені моделі були апробовані на тестових даних в якості яких виступили урагани Jose + Katia. Результати такого прогнозу представлені в таблиці 3.17.

Таблиця. 3.17

Матриця помилок для тестового набору даних ураганів Jose + Katia

		Predicted	
		0	1
Actual	0	1065	0
	1	1 (8%)	12 (92%)

При тестуванні навчені моделі показали знову однакові результати класифікації. Так з 1065 даних по відсутнім пікам не було жодної помилки. Із прогнозуванням наявних піків була лише одна помилка для урагану Jose на дату 17.09 (рис.3.21). Для урагану Katia єдиний пік був передбачено завчасно. Тобто точність для тестового набору склала 92%.

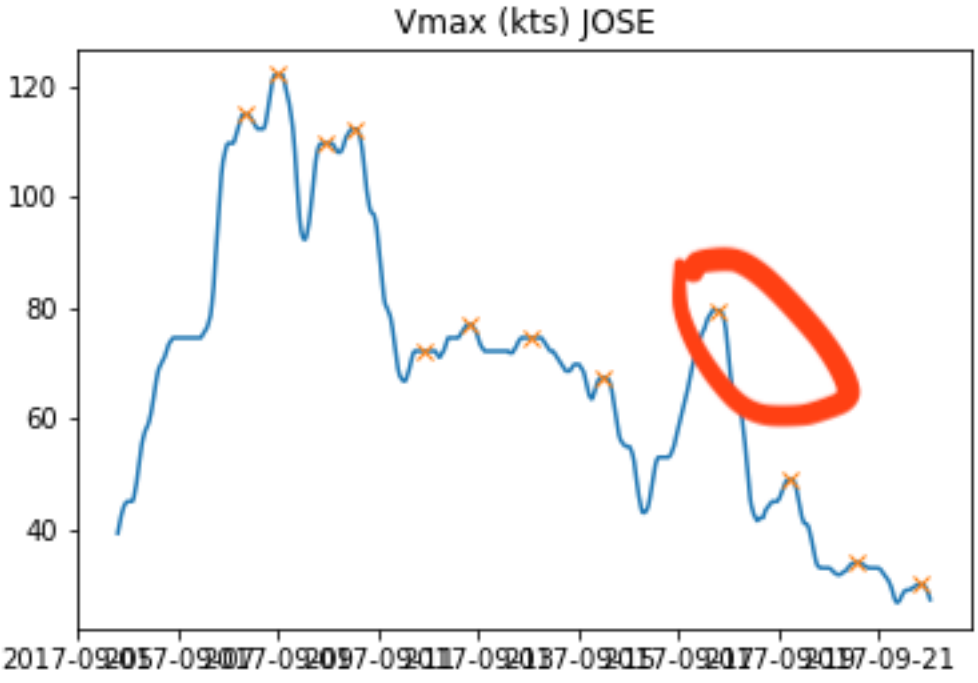


Рис.3.21. Нерозпізнані піки для урагану JOSE

Як видно з рисунку, цей нерозпізнаний пік знаходиться на останніх стадіях життя урагану Jose. Такий точний проноз свідчить про те, що вищезгадані 4 нерозпізнані піки на навчальній вибірці дійсно могли бути не пов'язаними із сонячним вітром. Для додаткового підтвердження цього було проведено повторне навчання де в якості навчальної вибірки виступив ураган Jose. Результати навчання представлені в таблиці 3.18.

В цьому випадку при навчанні виявився лише один хибний прогноз на аналогічно дату 17 вересня. Це підтверджує той факт, що цей пік не є спричинений сонячним вітром. Тестування на даних ураганів Irma + Katia показало наступні результати (таблиця 3.19)

Таблиця. 3.18

Матриця помилок для навчального набору даних на урагані Jose

		Predicted		
		0	1	
Actual	0	532	0	
	1	1 (8%)	11 (92%)	

Таблиця 3.19

Матриця помилок для тестового набору даних ураганів Jose + Katia

		Predicted	
		0	1
Actual	0	1062	0
	1	4 (36%)	7 (63%)

Нерозпізнаними виявились ті самі піки, на яких не змогли навчитись моделі у випадку, коли в якості навчальної вибірки виступили дані по урагану Irma. Отже зазначені на рис. піки не пов'язані із сонячним вітром. Тому можна зробити висновки, що більша частина піків на графіках швидкості вітру ураганів спричинена сонячним вітром. Використання класифікаційного підходу, запропонованого в цій роботі дає змогу спрогнозувати 100% піків, пов'язаних із сонячним вітром. Однак цей підхід не дає змогу передбачити інші піки, пов'язані із іншими факторами. Також стає зрозумілим той факт, що спалахи сонячного вітру виступають в якості каталізатора збільшення вітру ураганів. Однак абсолютні їх значення залежать від природи, розташування та інших факторів, які не враховані в цій моделі і можуть бути описані фізичними моделями динаміки ураганів. Зазначений підхід дає змогу здійснити прогноз зростання ураганів на 10 годин вперед.

3.3. Паводки

3.3.1. Результати розрахунків

Згідно нашої гіпотези, період затримки між паводком та спалахом сонячної активності може сягати 10 днів. Для перевірки цієї гіпотези був проведений наступний експеримент. Всі класифікаційні моделі та ансамблі моделей були навчені та протестовані для вхідних даних, що не містили часової затримки. Далі до вхідних параметрів додавались дані, що містили часову затримку в один день. Після чого моделі заново навчались та розраховувалась метрика recall. Ці ітерації продовжувались до 9 лагів. Формально задачі класифікації зводились до вигляду:

$$lag(0): Flood = F(X_1, \dots, X_9)$$

$$lag(1): Flood = F\left(X_1, \dots, X_9, X_{1,t-1}, \dots, X_{9,t-1}\right)$$

...

$$lag(9): Flood = F\left(X_1, \dots, X_9, X_{1,t-1}, \dots, X_{9,t-1}, \dots, X_{1,t-9}, \dots, X_{9,t-9}\right)$$

Отримані результати наведені в таблиці 3.20 та 3.21.

Таблиця 3.20

Точність recall для навчального набору при послідовному додаванні лагів

classifier	Lag 0	Lag 1	Lag 2	Lag 3	Lag 4	Lag 5	Lag 6	Lag 7	Lag 8	Lag 9
DecisionTreeClassifier	0,88	0,97	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
LogisticRegression	0,60	0,65	0,65	0,65	0,72	0,72	0,80	0,85	0,93	1,00
QuadraticDiscriminantAnalysis	0,68	0,70	0,83	0,93	0,97	1,00	1,00	1,00	1,00	1,00
GaussianNB	0,41	0,49	0,58	0,61	0,63	0,72	0,73	0,80	0,88	1,00
RandomForestClassifier	0,75	0,86	0,81	0,77	0,84	0,74	0,74	0,70	0,75	0,86
SVC	1,00	0,80	0,92	0,92	0,93	0,94	0,87	0,87	0,89	0,92
SGDClassifier	0,37	0,42	0,48	0,67	0,64	0,75	0,73	0,91	0,97	1,00
MLPClassifier	0,75	0,93	0,97	0,98	1,00	1,00	1,00	1,00	1,00	1,00
ExtraTreesClassifier	0,88	0,97	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
RadiusNeighborsClassifier	0,67	0,85	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
KNeighborsClassifier	0,63	0,64	0,69	0,80	0,81	0,83	0,87	0,85	0,85	0,90
OutputCodeClassifier	0,79	0,93	0,99	0,99	1,00	1,00	1,00	1,00	1,00	1,00
OneVsOneClassifier	0,58	0,67	0,61	0,65	0,71	0,73	0,82	0,89	0,97	1,00
OneVsRestClassifier	1,00	0,80	0,92	0,92	0,93	0,94	0,87	0,87	0,89	0,92
RidgeClassifier	0,55	0,67	0,62	0,66	0,70	0,74	0,82	0,87	0,96	1,00
PassiveAggressiveClassifier	0,00	0,55	0,60	0,65	0,64	0,66	0,82	0,96	0,98	1,00
GaussianProcessClassifier	0,00	0,00	0,00	0,91	1,00	1,00	0,61	0,65	0,74	0,88
AdaBoostClassifier	0,53	0,63	0,60	0,63	0,70	0,73	0,80	0,85	0,93	1,00
GradientBoostingClassifier	0,50	0,64	0,62	0,66	0,71	0,73	0,82	0,86	0,94	1,00
BaggingClassifier	0,95	0,80	0,93	0,95	0,94	0,94	0,91	0,93	0,90	0,91
BernoulliNB	0,00	0,57	0,59	0,64	0,65	0,71	0,73	0,78	0,89	0,97
LabelPropagation	0,88	0,95	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
LabelSpreading	0,82	0,94	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
LinearDiscriminantAnalysis	0,50	0,66	0,61	0,66	0,71	0,75	0,81	0,89	0,97	1,00
LinearSVC	0,58	0,67	0,61	0,65	0,71	0,73	0,82	0,89	0,97	1,00

Продовження таблиці 3.20

classifier	Lag 0	Lag 1	Lag 2	Lag 3	Lag 4	Lag 5	Lag 6	Lag 7	Lag 8	Lag 9
MultinomialNB	0,00	0,58	0,57	0,63	0,65	0,69	0,72	0,76	0,88	0,97
NearestCentroid	0,40	0,47	0,53	0,58	0,65	0,72	0,75	0,84	0,90	0,98
Perceptron	0,27	0,44	0,66	0,53	0,70	0,70	0,66	0,77	0,91	1,00
SVC	0,00	0,00	1,00	0,78	0,73	0,59	0,59	0,65	0,74	0,85
GaussianMixture	0,29	0,39	0,43	0,45	0,50	0,53	0,55	0,67	0,76	0,91

Таблиця 3.21

Точність recall для тестового набору при послідовному додаванні лагів

Classifier	**Lag 0**	**Lag 1**	**Lag 2**	**Lag 3**	**Lag 4**	**Lag 5**	**Lag 6**	**Lag 7**	**Lag 8**	**Lag 9**
DecisionTreeClassifier	0,16	0,37	0,40	0,43	0,45	0,52	0,59	0,64	0,70	0,84
LogisticRegression	0,03	0,29	0,44	0,48	0,53	0,55	0,64	0,74	0,84	0,97
QuadraticDiscriminantAnalysis	0,29	0,32	0,30	0,35	0,38	0,70	1,00	1,00	0,20	0,74
GaussianNB	0,30	0,45	0,51	0,51	0,54	0,61	0,65	0,68	0,75	0,90
RandomForestClassifier	0,08	0,09	0,18	0,38	0,51	0,55	0,92	0,97	1,00	1,00
SVC	0,06	0,20	0,20	0,36	0,49	0,74	0,92	1,00	1,00	1,00
SGDClassifier	0,33	0,35	0,43	0,67	0,49	0,49	0,62	0,64	0,80	0,97
MLPClassifier	0,14	0,34	0,45	0,43	0,53	0,53	0,65	0,68	0,85	0,93
ExtraTreesClassifier	0,15	0,33	0,30	0,41	0,48	0,62	0,80	0,85	0,97	1,00
RadiusNeighborsClassifier										
KNeighborsClassifier	0,22	0,30	0,34	0,32	0,36	0,48	0,54	0,69	0,80	0,95
OutputCodeClassifier	0,14	0,29	0,26	0,34	0,45	0,63	0,81	0,93	0,98	1,00
OneVsOneClassifier	0,07	0,34	0,45	0,51	0,54	0,57	0,53	0,65	0,76	0,92
OneVsRestClassifier	0,06	0,20	0,20	0,36	0,49	0,74	0,92	1,00	1,00	1,00
RidgeClassifier	0,05	0,32	0,46	0,50	0,53	0,53	0,57	0,58	0,69	0,87
PassiveAggressiveClassifier	0,23	0,39	0,53	0,49	0,35	0,55	0,64	0,60	0,78	0,93
GaussianProcessClassifier	0,00	0,00	0,00	0,00	0,30	0,59	1,00	1,00	1,00	1,00
AdaBoostClassifier	0,12	0,34	0,44	0,50	0,54	0,58	0,63	0,65	0,76	0,91
GradientBoostingClassifier	0,08	0,36	0,47	0,52	0,57	0,55	0,64	0,68	0,81	0,93
BaggingClassifier	0,04	0,19	0,17	0,29	0,48	0,59	0,85	0,99	1,00	1,00

Продовження таблиці 3.21

Classifier	Lag 0	Lag 1	Lag 2	Lag 3	Lag 4	Lag 5	Lag 6	Lag 7	Lag 8	Lag 9
BernoulliNB	0,02	0,36	0,42	0,46	0,52	0,62	0,67	0,75	0,81	0,96
LabelPropagation	0,10	0,30	0,34	0,30	0,46	0,51	0,56	0,64	0,75	0,84
LabelSpreading	0,14	0,31	0,36	0,36	0,49	0,52	0,58	0,63	0,77	0,86
LinearDiscriminantAnalysis	0,07	0,36	0,46	0,51	0,53	0,50	0,53	0,54	0,59	0,59
LinearSVC	0,07	0,34	0,45	0,51	0,54	0,57	0,53	0,65	0,76	0,92
MultinomialNB	0,00	0,23	0,35	0,42	0,53	0,65	0,69	0,75	0,84	0,97
NearestCentroid	0,63	0,54	0,59	0,58	0,55	0,56	0,62	0,62	0,73	0,88
Perceptron	0,33	0,33	0,39	0,34	0,51	0,56	0,58	0,60	0,78	0,92
SVC	0,00	0,00	0,00	0,09	0,25	0,99	1,00	1,00	1,00	1,00
GaussianMixture	0,30	0,43	0,54	0,43	0,47	0,28	0,33	0,67	0,37	0,48

3.3.2. Аналіз точності

Ці таблиці дають змогу проаналізувати динаміку зміни метрики при послідовному врахуванні нових лагів до вхідних параметрів. Значення recall порівнювалось для тестового та навчального наборів. Аналіз точності моделей оцінювався за такими ознаками: якщо помилка тестового та навчального наборів є близькою (маленька дисперсія) – це свідчить про те що модель добре навчилась та прогнозує невідомі значення на рівні відомих. А абсолютне значення свідчить наскільки точною є така модель. Якщо ж точність на навчальному наборі сягає 1, а на тестовому близька до 0.5 – це явна ознака перенавчання. Тобто ідеально прогнозуються відомі дані, а невідомі вгадуються 50/50 – абсолютна нездатність прогнозувати. Такі моделі мають бути усунені з аналізу. (таблиця 3.22.)

Згідно цієї таблиці можна бачити лише один класифікатор не справився із задачею – RadiusNeighborsClassifier. У всіх інших точність прогнозу зростає із збільшенням лагу. Це означає, що дійсно є присутньою суттєва часова затримка між спалахом на сонці та настанням паводку. Для аналізу, які фактори важливі при такому прогнозуванні, побудуємо дерево рішень рис 3.22. (recall = 0.84)

Таблиця 3.22

Дисперсія похибки між тестовим та навчальним наборами даних

Classifier	Lag 0	Lag 1	Lag 2	Lag 3	Lag 4	Lag 5	Lag 6	Lag 7	Lag 8	Lag 9
DecisionTreeClassifier	0,82	0,62	0,60	0,57	0,55	0,48	0,41	0,36	0,30	0,16
LogisticRegression	0,96	0,56	0,32	0,26	0,27	0,24	0,20	0,14	0,09	0,03
QuadraticDiscriminantAnalysis	0,58	0,54	0,63	0,62	0,61	0,30	0,00	0,00	0,80	0,26
GaussianNB	0,27	0,09	0,12	0,16	0,14	0,15	0,11	0,15	0,14	0,10
RandomForestClassifier	0,90	0,89	0,78	0,50	0,39	0,26	-0,24	-0,38	-0,33	-0,16
SVC	0,94	0,75	0,78	0,61	0,47	0,21	-0,06	-0,14	-0,12	-0,08
SGDClassifier	0,11	0,17	0,10	-0,01	0,23	0,36	0,16	0,30	0,18	0,03
MLPClassifier	0,81	0,64	0,54	0,56	0,47	0,47	0,35	0,32	0,15	0,07
ExtraTreesClassifier	0,83	0,66	0,70	0,59	0,52	0,38	0,20	0,15	0,03	0,00
RadiusNeighborsClassifier	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
KNeighborsClassifier	0,65	0,54	0,51	0,60	0,55	0,42	0,38	0,18	0,06	-0,06
OutputCodeClassifier	0,82	0,69	0,74	0,66	0,55	0,37	0,19	0,07	0,02	0,00
OneVsOneClassifier	0,88	0,49	0,26	0,22	0,24	0,22	0,36	0,26	0,21	0,08
OneVsRestClassifier	0,94	0,75	0,78	0,61	0,47	0,21	-0,06	-0,14	-0,12	-0,08
RidgeClassifier	0,91	0,52	0,26	0,24	0,24	0,29	0,30	0,34	0,28	0,13
PassiveAggressiveClassifier		0,30	0,12	0,25	0,46	0,16	0,22	0,37	0,21	0,07
GaussianProcessClassifier				1,00	0,70	0,41	-0,64	-0,53	-0,36	-0,14
AdaBoostClassifier	0,78	0,46	0,26	0,20	0,22	0,21	0,21	0,23	0,18	0,09
GradientBoostingClassifier	0,83	0,45	0,25	0,22	0,20	0,24	0,22	0,21	0,15	0,07
BaggingClassifier	0,96	0,77	0,82	0,70	0,49	0,37	0,07	-	-	-

Продовження таблиці 3.22

Classifier	**Lag 0**	**Lag 1**	**Lag 2**	**Lag 3**	**Lag 4**	**Lag 5**	**Lag 6**	**Lag 7**	**Lag 8**	**Lag 9**
								0,07	0,11	0,09
BernoulliNB		0,36	0,30	0,28	0,20	0,13	0,08	0,04	0,09	0,02
LabelPropagation	0,89	0,69	0,66	0,70	0,54	0,49	0,44	0,36	0,25	0,16
LabelSpreading	0,84	0,67	0,64	0,64	0,51	0,48	0,42	0,37	0,23	0,14
LinearDiscriminantAnalysis	0,86	0,46	0,25	0,23	0,25	0,33	0,35	0,39	0,39	0,41
LinearSVC	0,88	0,49	0,26	0,22	0,24	0,22	0,36	0,26	0,21	0,08
MultinomialNB		0,60	0,39	0,34	0,17	0,06	0,05	0,00	0,04	0,01
NearestCentroid	-0,55	-0,15	-0,12	0,01	0,15	0,22	0,17	0,26	0,19	0,10
Perceptron	-0,23	0,26	0,40	0,35	0,27	0,20	0,12	0,21	0,15	0,08
SVC			1,00	0,89	0,66	-0,68	-0,70	-0,53	-0,36	-0,18
GaussianMixture	-0,02	-0,10	-0,27	0,06	0,06	0,47	0,39	0,01	0,52	0,47

Важливим при класифікації є Індекс Джині, також відомий як домішка Джіні, обчислює кількість імовірності певної ознаки, яка неправильно класифікована при випадковому виборі. Якщо всі елементи пов'язані з одним класом, то його можна назвати чистим. Як видно з рисунку, для встановлення паводку першою перевіркою є Proton Density із затримкою в 9 днів. Якщо спалаху інтенсивності цього фактору в цей день не спостерігалося тоді перевіряється Ion Temperature із затримкою 0 днів. Якщо ж спостерігається спалах на Proton Density, то із 100% вірогідністю має настати паводок. На основі цього дерева рішень можна визначити і важливість факторів (таблиця 3.23):

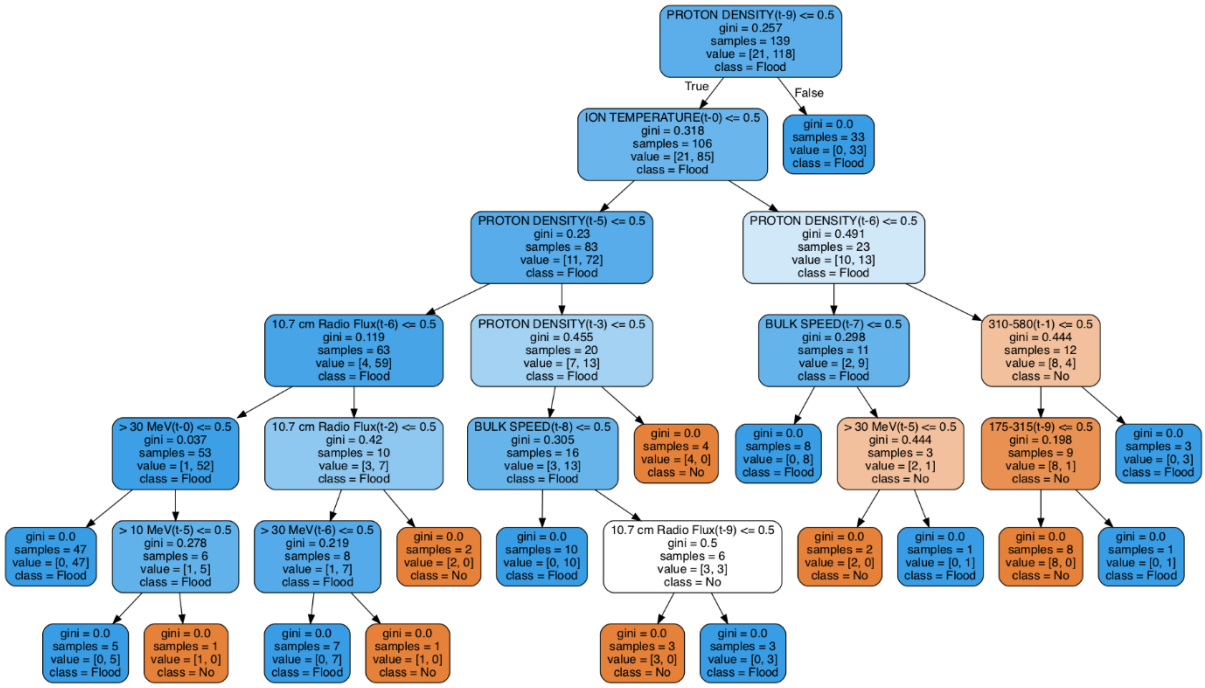


Рис.3.22. Дерево рішень прогнозу паводку при врахування лагової затримки від 0 до 9 днів.

Таблиця 3.23

Найбільш вагомі фактори при класифікації:

Фактор (лаг)	Важливість
PROTON DENSITY(t-3)	0.19
310-580(t-1)	0.10
ION TEMPERATURE(t-0)	0.09
10.7 cm Radio Flux(t-9)	0.08
PROTON DENSITY(t-6)	0.08
PROTON DENSITY(t-5)	0.07
10.7 cm Radio Flux(t-2)	0.07
PROTON DENSITY(t-9)	0.06
BULK SPEED(t-7)	0.05

Як видно з таблиці, не дивлячись на те, що перша перевірка стосується PROTON DENSITY(t-9), найбільш вагомими факторами є PROTON DENSITY(t-3), 310-580(t-1) та ION TEMPERATURE(t-0). Видно також, що різні фактори впливають на настання паводку з різними часовими затримками. Також видно, що спалах такого фактору як PROTON DENSITY може призводити до повеней з різними часовими затримками. Або необхідно декілька спалахів, щоб це призвело до паводку. Слід зазначити, що точність цього класифікатора на навчальному наборі становить 1, а на тестовому 0.84.

3.3.3. Побудова прогнозних моделей

Для побудови прогнозу паводків на n днів наперед необхідно вилучити із вхідних параметрів дані з лагами [0-(n-1)]:

$$Forecast(1\ day): Flood = F\left(X_{1,t-1}, \dots, X_{9,t-1}, , \dots, X_{9,t-1}, \dots, X_{1,t-9}, \dots, X_{9,t-9}\right)$$

...

$$Forecast(9\ days): Flood = F\left(X_{1,t-9}, \dots, X_{9,t-9}\right)$$

Як можна бачити, кількість вхідних параметрів буде зменшуватись, а отже це має призвести до зменшення точності прогнозу. В роботі були проаналізовані точності прогнозних класифікаційних моделей від 0 до 9 днів наперед. Також був побудований ансамбль моделей, що поєднував всі моделі шляхом hard voting та проаналізована динаміка його точності в залежності від затримки пронозу. Результати представлені в таблиці 3.24 та на рисунку 3.23.

Таблиця 3.24

Точність recall для прогнозних моделей для тестового набору даних

classifier	Lag 0	Lag 1	Lag 2	Lag 3	Lag 4	Lag 5	Lag 6	Lag 7	Lag 8	Lag 9
DecisionTreeClassifier	0,85	0,81	0,83	0,83	0,82	0,76	0,75	0,79	0,82	0,84
LogisticRegression	0,97	0,97	0,97	0,97	0,97	0,96	0,94	0,95	0,99	1,00
QuadraticDiscriminantAnalysis	0,68	0,44	1,00	1,00	1,00	1,00	1,00	1,00	0,00	0,00
GaussianNB	0,90	0,90	0,90	0,90	0,90	0,86	0,84	0,84	0,84	0,84
RandomForestClassifier	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,99	1,00
SVC	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,99	1,00	0,92
SGDClassifier	0,93	0,91	0,91	0,91	0,91	0,90	0,91	0,92	0,89	0,88
MLPClassifier	0,94	0,92	0,94	0,96	0,92	0,91	0,89	0,91	0,92	0,87
ExtraTreesClassifier	1,00	0,99	1,00	0,99	0,99	0,97	0,95	0,91	0,90	0,87
RadiusNeighborsClassifier	0,95	0,91	0,89	0,88	0,86	0,92	0,84	0,78	0,78	0,80
KNeighborsClassifier	1,00	0,99	1,00	1,00	1,00	1,00	0,97	0,96	0,91	0,92
OutputCodeClassifier	0,90	0,91	0,89	0,87	0,86	0,90	0,89	0,92	0,91	0,92
OneVsOneClassifier	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,99	1,00	0,92
OneVsRestClassifier	0,85	0,83	0,84	0,84	0,85	0,86	0,90	0,94	0,98	1,00
RidgeClassifier	0,94	0,93	0,93	0,91	0,88	0,90	0,87	0,91	0,91	1,00
PassiveAggressiveClassifier	1,00	0,99	1,00	1,00	1,00	0,98	1,00	1,00	1,00	1,00
GaussianProcessClassifier	0,95	0,92	0,91	0,92	0,91	0,91	0,89	0,90	0,85	0,84
AdaBoostClassifier	0,86	0,85	0,80	0,78	0,80	0,83	0,79	0,76	0,75	0,91
GradientBoostingClassifier	0,87	0,88	0,83	0,80	0,82	0,86	0,84	0,79	0,75	0,91
BaggingClassifier	0,62	0,62	0,65	0,68	0,74	0,84	0,85	0,89	0,91	1,00
BernoulliNB	0,90	0,91	0,89	0,87	0,86	0,90	0,89	0,92	0,91	0,92
LabelPropagation	0,96	0,93	0,95	0,93	0,94	0,93	0,91	0,92	0,91	1,00
LabelSpreading	0,90	0,87	0,85	0,80	0,81	0,80	0,76	0,71	0,72	0,67

Продовження таблиці 3.24

classifier	Lag 0	Lag 1	Lag 2	Lag 3	Lag 4	Lag 5	Lag 6	Lag 7	Lag 8	Lag 9
LinearDiscriminantAnalysis	0,93	0,95	0,92	0,91	0,93	0,86	0,86	0,86	0,84	0,87
LinearSVC	1,00	1,00	1,00	1,00	1,00	1,00	1,00	.00	1,00	1,00
MultinomialNB	0,91	0,94	0,91	0,90	0,93	0,89	0,91	0,91	0,88	0,86
NearestCentroid	0,91	0,91	0,89	0,85	0,86	0,88	0,86	0,88	0,91	0,91
Perceptron	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,99	0,94	0,87
VotingClassifier	0,97	0,97	0,97	0,97	0,97	0,94	0,92	0,93	0,92	0,92

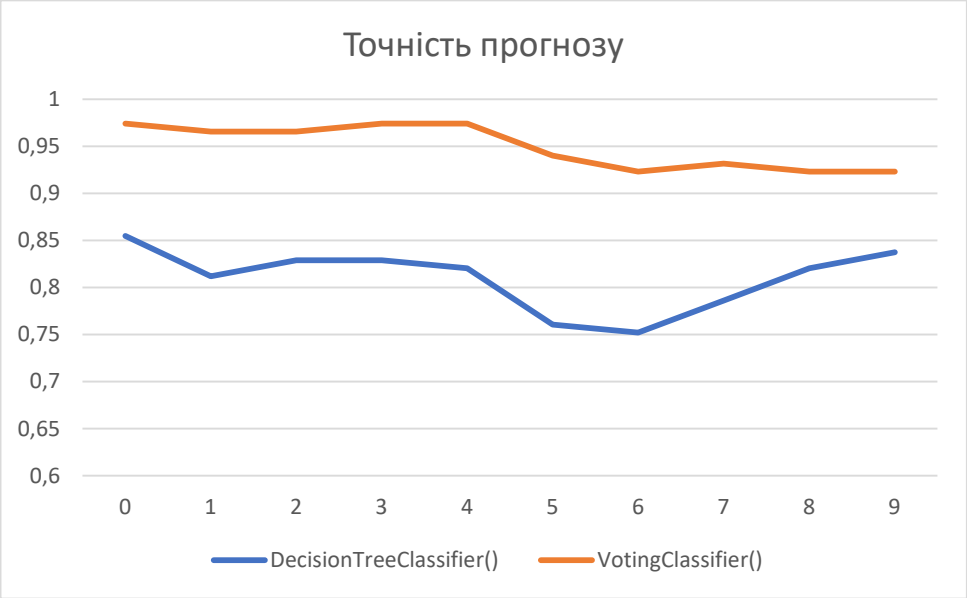


Рис 3.23. Зміна точності класифікаційних моделей Decision Tree та VotingClassifier залежно від дальності пронозу.

Як видно з таблиці та рисунку точність ансамблю моделей VotingClassifier є найвищою та поступово спадає із збільшенням дальності пронозу. Слід зазначити, що точність на тестовому наборі 0.97 свідчить про високу точність та відсутність перенавчання. Тобто такий ансамбль моделей може бути використаний для прогнозу паводків до 9 днів наперед. Недоліком є те, що на основі нього не можна побудувати дерево рішення.

Точність Decision Tree коливається на одному рівні в межах похибки моделі. Це пояснюється тим, що перший критерій перевірки це PROTON DENSITY з лагом 9. Тому і точність моделі практично не залежить від вилучення факторів з малими лагами. Такий підхід дозволяє побудувати дерево рішень для прогнозу на будь який лаг. Тобто в такому підході прогноз для лагів від 0 до 9 вимагає побудови 10 різних дерев рішень. Для випадку 0-9 лагів дерево рішень побудоване на рис 3. Побудуємо для прикладу дерево для прогнозу на 5 та 9 днів наперед: (рис.3.24)

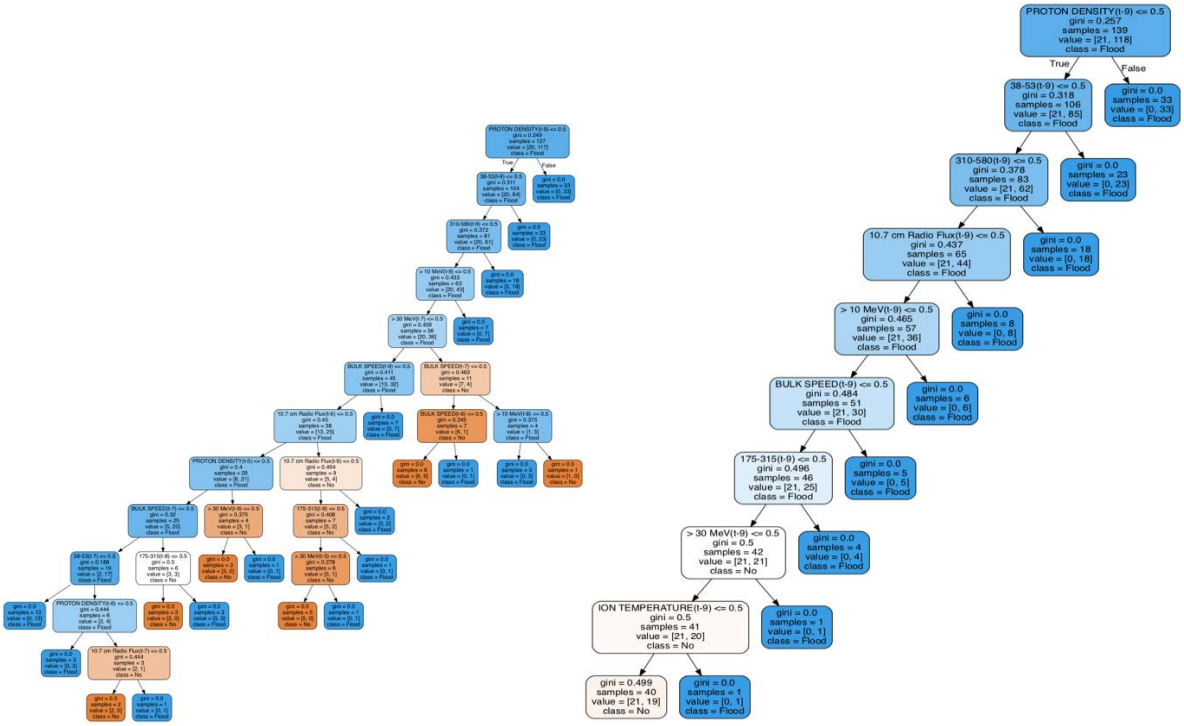


Рис.3.24. Дерево рішень на 5 та 9 днів вперед.

Розрахована важливість факторів наведена в таблиці 3.25.

Як видно з рисунка 3.24, перший тест для PROTON DENSITY залишається, але найважливішими факторами для прогнозування повені за 9 днів є $310\ \text{keV} \leq \text{DF} \leq 580\ \text{keV}$, і $38\ \text{keV} \leq \text{DF} \leq 53\ \text{keV}$. Таким чином, на основі отриманого ансамблю класифікаційних моделей можна прогнозувати до 9 днів наперед з точністю 92% . А за допомогою дерева рішень можна обґрунтовувати та будувати рекомендації для прогнозування паводків.

Таблиця 3.25

Найбільш вагомі фактори при прогнозуванні на 5 та 9 днів наперед:

Прогноз на 5 днів		Прогноз на 9 днів	
Фактор (лаг)	Важливість	Фактор (лаг)	Важливість
BULK SPEED (t-7)	0,10	310 keV ≤ DF ≤ 580 keV(t-9)	0,19
175-315(т-8)	0,09	38 keV ≤ DF ≤ 53 keV(t-9)	0,15
310-580(t-9)	0,08	PROTON DENSITY (t-9)	0,13
38-53(t-9)	0,06	10,7 см Radio Flux (t-9)	0,12
> 30 MeB(t-7)	0,06	BULK SPEED (t-9)	0,12
PROTON DENSITY (t-5)	0,06	175 keV ≤ DF ≤ 315 keV(t-9)	0,12
PROTON DENSITY (t-9)	0,05	*IPF*> 10 MeB(t-9)	0,12
BULK SPEED (t-6)	0,05	ION TEMPERATURE (t-9)	0,03
> 30 MeB(t-5)	0,05	*IPF*> 30 MeB(t-9)	0,03

3.3.4. Обговорення результатів

Потенційне теоретичне (фізичне) пояснення механізму, який міг би пояснити розглянуту взаємодію в цій роботі, було представлено кількома авторами. Відповідно до [129, 130], високоенергетичні частинки від Сонця захоплюють повітряні маси гідродинамічним тиском і безпосередньо впливають на атмосферні процеси. Якщо в місці контакту з повітряними масами відбувається насичення вологою, то можуть утворюватися хмари і випадати опади, при цьому механізм утворення опадів пояснюється принципом валентності електронів. Автори стверджують, що поява хмар і опадів, а також поява спекотних хвиль і сухих періодів зумовлена насамперед електромагнітними характеристиками сонячного вітру, розташуванням Сонця, від якого він випромінюється, і його хімічна

структура. Вищевказаний механізм пояснюється циркуляцією векторів міжпланетних магнітних полів. Prikryl [131] обговорили дві раптові повені в Словаччині, що відбулися після прибуття двох високошвидкісних потоків сонячного вітру з корональних дір. В наступній праці ці автори довели, що сильні опади, що призводять до повеней і раптових повеней у Японії, Австралії та континентальній частині США, як правило, слідують за надходженням високошвидкісних потоків сонячного вітру з корональних дір. Вони припустили, що низхідні атмосферні гравітаційні хвилі можуть спровокувати утворення серії конвективних осередків, які спричинили сильні опади та повені. Відповідно до цих раніше опублікованих результатів, статистичні результати, представлені в цьому дослідженні, показують, що повені, спричинені опадами, мають тенденцію слідувати за раптовими надходженнями потоків сонячно заряджених частинок.

3.4. Висновки до розділу 3

Використання комплексного підходу, що об'єднує традиційні статистичні методи та інноваційні технології машинного навчання такі як ANFIS, ANN, та LSTM, показали високу точність та адекватність, а аналіз чутливості відкриває нові можливості для створення ефективних систем раннього попередження, що можуть мінімізувати ризики та наслідки природних катастроф.

В результаті проведених досліджень:

- На основі розроблених методів що базуються на ANFIS та LSTM, що відрізняються від інших досліджень врахуванням часових зсувів та нелінійних взаємодій між параметрами сонячної активності та атмосферними умовами було підтверджено зв'язок між сонячною активністю та частотою і розподілом лісових пожеж у різних географічних регіонах, вдалось покращити точність прогнозування цих подій до 87% (лаг 2) малих та 93% (лаг 1) великих лісових пожеж за

допомогою ANFIS та до 92% (лаг 3) малих та 87% (лаг 2) великих за допомогою LSTM. А коефіцієнти кореляції отриманих моделей встановили 0,91 та 0,92 для прогнозу температури та вологості відповідно. Також для всіх моделей був проведений аналіз чутливості та визначений вплив факторів сонячного активності на ці кризові явища.

- Було проведено аналіз ураганів та їх зв'язку з сонячною активністю, з використанням передових методів прогнозування, включаючи рекурентні нейронні мережі (LSTM), нейронні мережі та лінійні моделі. Для аналізу були вибрані урагани IRMA, JOSE та KATIA, що дозволило дослідити залежності між характеристиками цих ураганів і параметрами сонячної активності. Результати показали, що моделі LSTM забезпечують найкраще відтворення динаміки ураганів, виокремлюючи їх поведінку з високою точністю $R^2 = 0,99$ – LSTM, 0,90 – ANN, 0.86 – Лінійні з врахуванням лагу 4 дні. Прогнозування на основі піків досягнуло точності 92%. Це дало змогу будувати прогнозні моделі до чотирьох днів.
- Було проведено аналіз впливу сонячної активності на повені, застосовуючи серію класифікаційних моделей та ансамблів моделей для визначення часової затримки між сонячними спалахами та настанням поводків, що відрізняється від інших досліджень більш глибоким аналізом часових зсувів та впливу різних сонячних параметрів. Цей підхід дозволив ідентифікувати специфічні фактори, які мають найбільший вплив на повені, та встановити, що період затримки може сягати до 10 днів, що дало змогу підвищити точність прогнозування поводків на основі ансамблю класифікаторів до 97% (на день вперед) та 92% (на 9 днів вперед) і надало теоретичне підґрунтя для розробки ефективних систем раннього попередження, спроможних мінімізувати ризики та збитки, пов'язані з цими природними катастрофами.

Основні наукові результати розділу опубліковані в працях [120 – 124].

РОЗДІЛ 4.
РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ

4.1. Розробка інформаційної технології

Перш за все, визначимо вимоги до архітектури та функціональних залежностей, в нашому випадку це використання інференсів з малими і середніми даними. Далі на основі цих інференсів визначимо оптимальну побудову з точки зору інженерії програмного забезпечення платформу і імплементуємо ітераційну складову для інтеграції і планування безпечних туристичних подорожей, а також можливість незалежної роботи платформи.

Ключові аспекти системи:

oIntegration: інтеграція з сторонніми сервісами за допомогою REST API, можливість додатково інтеграції з брокрерами повідомлень або базами даних – сторонніми додатками і комплексами .

oSimplistic and Intuitive: система повинна бути простою та зрозумілою у використанні, легкою в освоєнні для нових користувачів. Містити Web інтерфейс для користувача.

oFlexibility: система повинна підтримувати можливість зміни і розширення, додавання нових моделей у модуль прогнозів.

oReliability: система повинна бути надійною, зберігати усі повідомлення, навіть у разі відмови або аварійного закриття застосунку або платформи.

oSecurity: система повинна мати достатній рівень безпеки відповідно існуючим на сьогодення стандартам.

Функціональні вимоги:

Система повинна мати функціонал керування платформою через веб застосунок, на якій буде відображатись хід виконання інференсу (прогнозу), проміжні результати і можливість редагування інференсу.

Система повинна мати можливість розширення – наприклад додавання двофакторної авторизації для входу в систему, безпарольний доступ (Kerberos). Додавання нових моделей прогнозу і потоку їх виконання. Можливості перегляду потоку виконання прогнозу. Також ми маємо передбачити можливість виконання інференсів в контейнерізованому середовищі

В попередніх етапах дослідження визначено оптимальні моделі для прогнозування лісових пожеж, паводків і ураганів. Всі відповідні інференси використовують малі і середні данні. Приведемо інференси до потоку виконання з точки зору програмної інженерії, а саме опишемо процес прогнозування за допомогою UML дігамми. На рис. 4.1 подано діаграму виконання інференсу

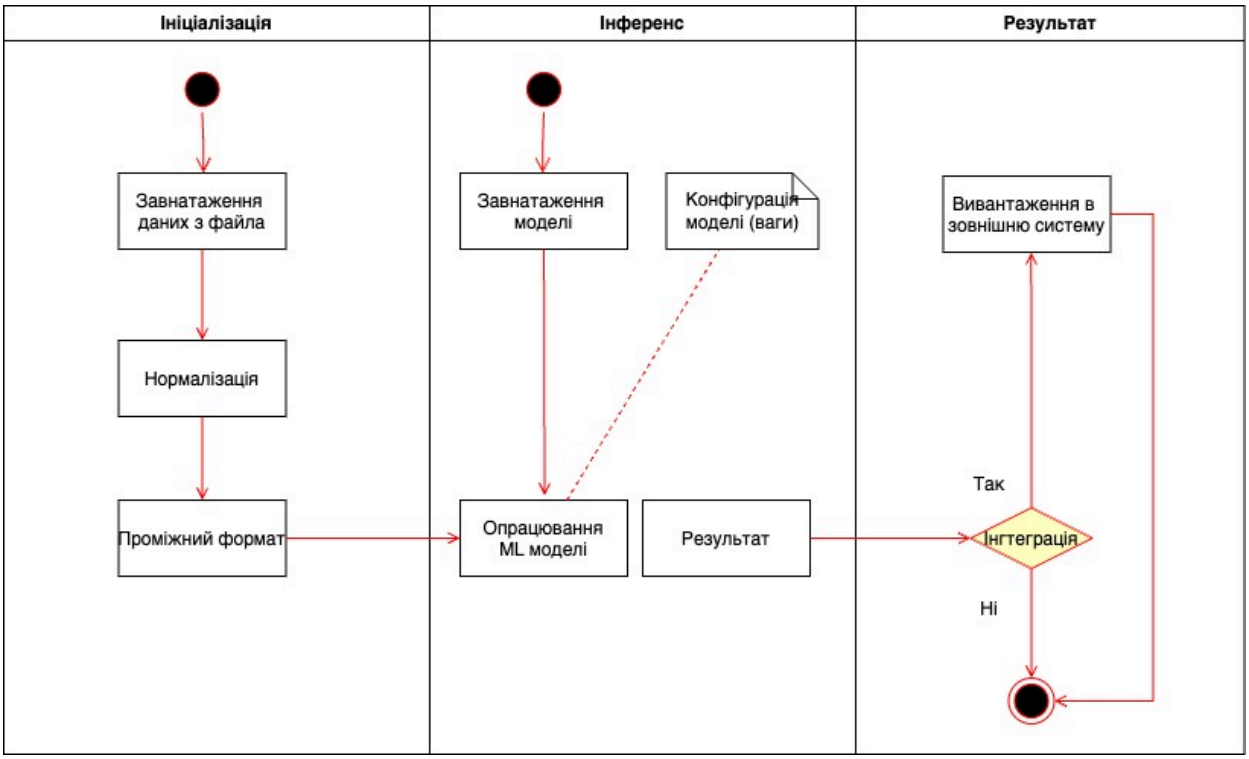


Рис. 4.1 UML діаграмf виконання інференсу

З UML діаграми видно, що виконання інференсу складається з завантаження даних, формуванню проміжного формату з нормалізацією

даних. Завантаженню в модель і отриманню прогноза. З можливістю збереження або вивантаження в інші зовнішні системи. Для машинного навчання є зміст використовувати набір практик MLOps (скорочено від Machine Learning Operations).

MLOps — це практика, яка об'єднує машинне навчання (ML) та операційні процеси (Operations) для автоматизації й оптимізації циклу розробки, розгортання та підтримки моделей машинного навчання. Основна мета MLOps — забезпечити ефективну співпрацю між командами розробників, дослідників даних і IT-спеціалістів для більш швидкого й надійного впровадження моделей у виробниче середовище.

Аспекти MLOps:

1. **Автоматизація процесів** — MLOps автоматизує процеси підготовки даних, тренування моделей, оцінки якості, розгортання та моніторингу. Це знижує потребу в ручних операціях і підвищує ефективність.
2. **Управління життєвим циклом моделей** — в MLOps важливо організувати процеси від створення і тренування моделей до їхнього розгортання та оновлення, враховуючи етапи перевірки, тестування та оптимізації.
3. **Безперервна інтеграція та розгортання (CI/CD)** — використовуються практики безперервної інтеграції та розгортання, щоб зробити процес оновлення моделей безперебійним. Це дозволяє швидко вносити зміни та впроваджувати нові версії моделей без зупинки роботи системи.
4. **Моніторинг і підтримка якості** — MLOps передбачає постійний моніторинг продуктивності моделей у реальному часі, що дозволяє вчасно реагувати на зміни в даних чи поведінці моделей. Це допомагає уникнути деградації якості моделі.

5. **Репродуктивність і контроль версій** — MLOps забезпечує відстеження версій даних, моделей і конфігурацій, що дозволяє відтворювати результати експериментів і контролювати зміни.
6. **Управління даними та безпека** — MLOps також стосується управління доступом до даних, конфіденційності, а також захисту від можливих загроз безпеці, особливо під час роботи з великими обсягами чутливих даних.

Відповідно до принципів розробки програмного забезпечення, здається, що має зміст використати принцип не повторюватись і звести до спільного потоку виконання прогнозу. Але таке узагальнення приведе до ускладнень з розширення і подальшої модифікації системи. Дані проблеми відомі і описані при формуванні підходів і практик MLOps [140]. спільного потоку виконання прогнозу. Але таке узагальнення приведе до ускладнень з розширення і подальшої модифікації системи. Дані проблеми відомі і описані при формуванні підходів і практик MLOps [140]. Для отримання технології визначимо додаткові критерії які не описані в підході MLOps, а саме платформу координатора для виконання інференсів і порівняємо з іншими платформами.

Також ми не розглядаємо хмарні платформи, як наприклад AWS Sagemaker, AWS Bedrock. Данні хмарні сервіси відносяться до SaaS та PaaS відповідно. Також даним хмарним сервісам притаманна надлишковість як для платформ великих даних. Тому надалі будемо вважати, що це частина платформи великих даних, хоча інтерфейс і інтеграційні інтерфейси максимально спрощені для використання у якості SaaS або PaaS. Також їх не ефективно використовувати при невеликих інфернсах, за рахунок додавання нативними хмарними сервісами координатора.

У таблиці 4.1 представлено порівняння платформ і надмірностей для кожної з платформ.

Таблиця 4.1

Порівняння платформ і надмірності

Вид даних	Платформа	Складність реалізації	Надмірність
Малі	Скрипти, програми	Не складно для одного інференса	Мінімальна. Прі складних потоках виконання потребує координатора з ациклічним графом виконання. Мінімальна
Середні	Програми і координатори	Використання готових координаторів	Ациклічним графом виконання. Прийнятна надмірність
Великі	Big Data з координатором	Платформа на який видувається інференс зазвичай є цілою екосистемою	Прийнятна тільки при великій кількості даних і інтересів

Як видно з таблиці використання лише координатора або платформи координатора доцільне для іференсів з малими і середніми даними, що додає мінімальну надлишковість при розробці або плануванні програмного забезпечення або рішень на основі машинного навчання.

4.2. Архітектура та особливості системи забезпечення безпекових рекомендацій.

На основі запропонованої інформаційної технології в дисертаційній роботі запропоновано інформаційну технологію на основі якої побудовано платформу "Безпечний туризм", що спрямований на наданні прогнозів природних катастроф. Есенція платформи полягає у визначенні ступеню небезпеки, які допоможуть користувачам інших існуючих систем уникнути потенційних ризиків під час їхніх подорожей і завчасно інформувати про небезпеку.

Визначимо варіанти інтеграції з вже існуючими системами, що спростить нам задачу на порядок. Існуючи системи вже мають мобільні і інші додатки і засоби інформування користувачів. Платформа має підтримувати декілька рівней інтеграції з зовнішніми платформами.

На рисунку 4.2 представлена схема такої взаємодії з трьома можливими шляхами оновлень сторонніх додатків.

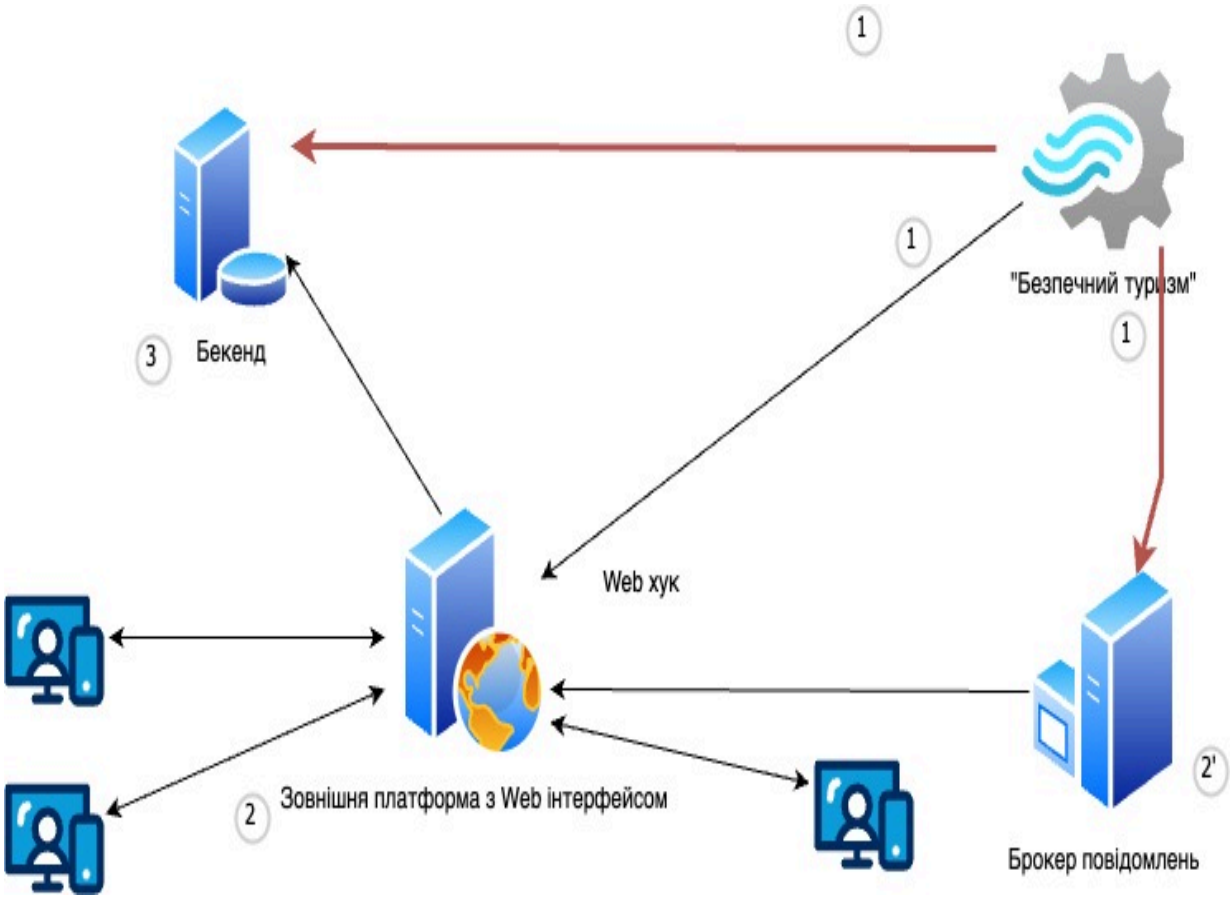


Рис. 4.2. Взаємодія з зовнішніми платформами [132]

Використовуючи сервіс орієнтований підхід (SOA) виберемо координатор з наявним Web інтерфейсом і REST, з можливістю адміністрування з допомогою Web інтерфейсу. Так як весь потік виконання інференса написаний на мові програмування Python, для зменшення складності оберем координатор з підтримкою мови програмування Python.

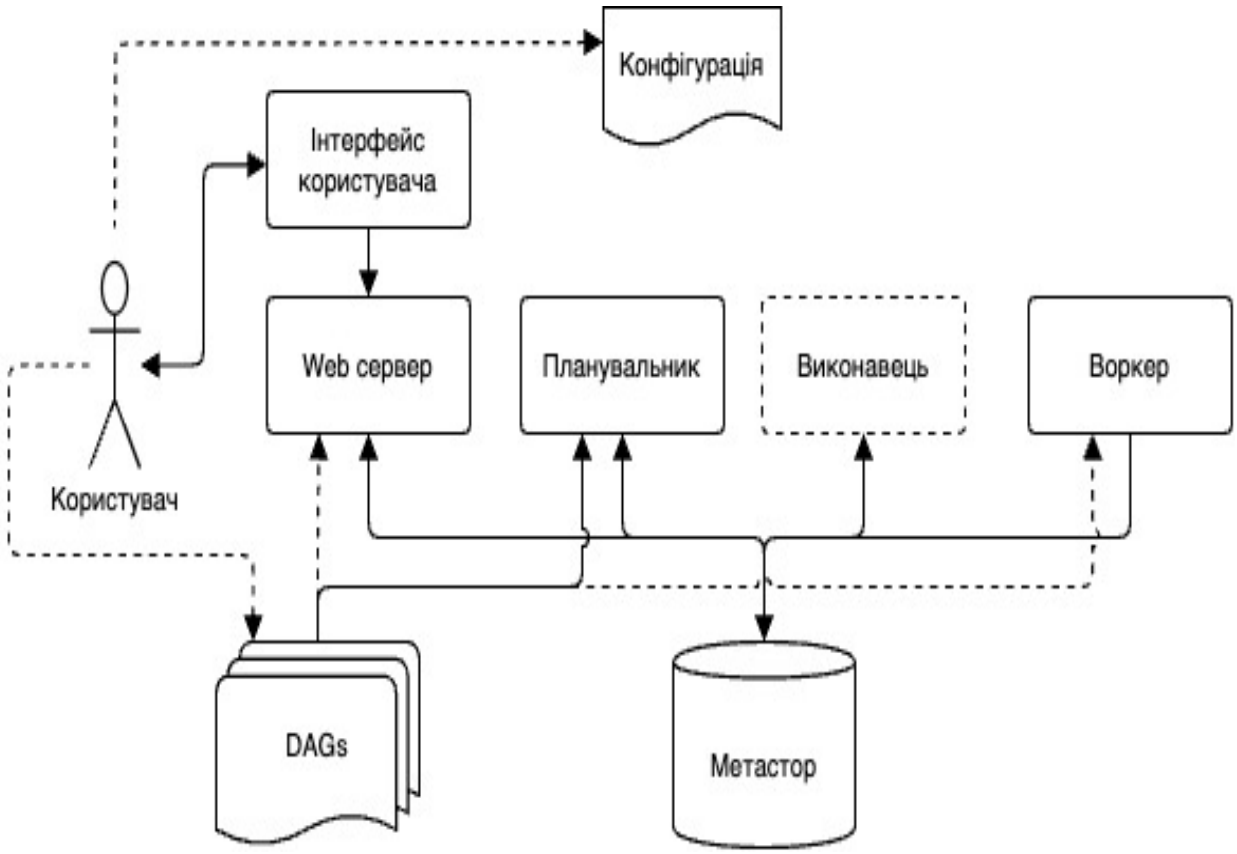


Рис. 4.3 Сервіс координатора

Сервіс координатора має містити запис про проходження кожного інференса з логіюванням, постійною базою даних та метасховищем. Представлення платформи координатора на рис 4.4. Вбудований веб сервер має містити як інтерфейс користувача і адміністратора сервісу, так можливість інтеграції REST API для сторонніх платформ і додатків. Планувальник має підтримувати можливість запуску за розкладом.

У нашому випадку з врахуванням зазначених властивостей визначимо Apache Airflow, як сервіс. На основі нього побудуємо відповідну платформу з трьома окремим інференсами і можливість інтеграції з сторонніми сервісами за допомогою баз даних, Web API і брокера повідомлень.

Відповідний сервіс задовольняє всім нашим вимогам, а також зменшує час на додавання нових версій координатора або розгортання виправлень.

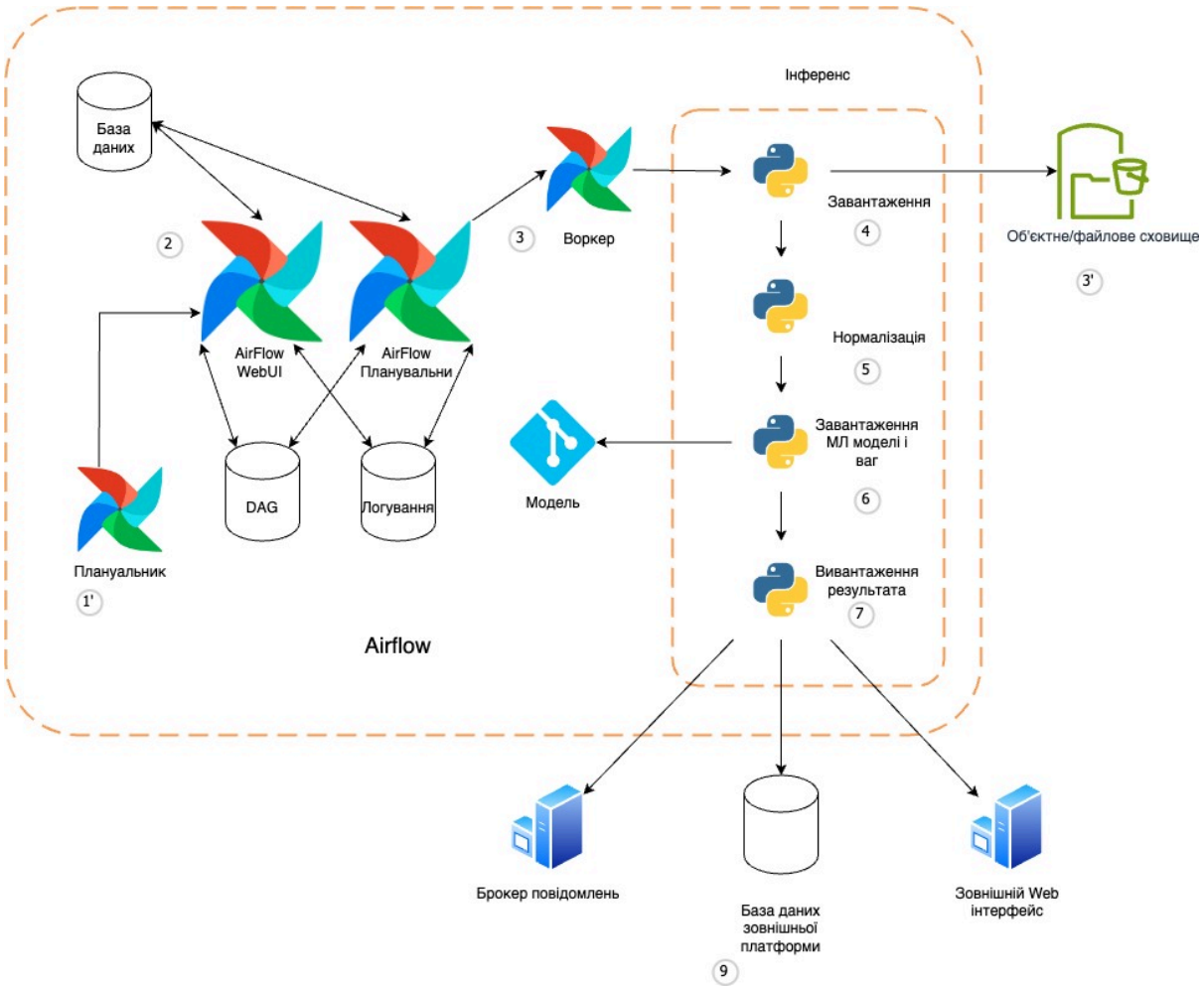


Рис. 4.4. SOA платформа

На рисунку 4.4 представлена схема реалізації даної платформи. Отже, розроблення та впровадження платформу інформаційної системи "Безпечний туризм" значно сприяє підвищенню рівня безпеки мандрівників, надаючи оброблену інформацію про потенційні ризики та ефективно уникати небезпечних ситуацій під час подорожей.

4.3. Обговорення результатів дослідження

В контексті практичної значимості розробленої платформи "Безпечний туризм", можна порівняти отримані результати з наявними даними про тенденції використання додатків та технологій у туристичній галузі. Сучасні дослідження підкреслюють зростаючу потребу в мобільних

рішеннях, що забезпечують безконтактні взаємодії та підвищують безпеку мандрівників [134, 135].

Значна частина мандрівників виявила бажання використовувати існуючі мобільні додатки, зі збільшенням функціоналу на відміну від окремих додатків або окремих платформ. Це підкреслює значення розробки систем, які може розширювати можливості існуючих систем [136, 137].

Понад це, дослідження показують, що користувачі мобільних додатків охочіше користуються послугами готелю, такими як замовлення їжі в номер або відвідування ресторану готелю, якщо для цього доступні мобільні додатки. Вони також очікують відповідний рівень безпеки, що є важливою функцією, яку може надавати інформаційна система "Безпечний туризм" вже існуючим додаткам і системам [138, 139].

Отже, порівняння результатів дослідження з вищезгаданими даними підкреслює актуальність і практичну значущість розробленої системи, а також відкриває шляхи для подальшого її вдосконалення та адаптації до змінюваних потреб і очікувань мандрівників.

Водночас, у статті [139] розглянуто різні методології та підходи до створення систем, що надають туристичним системам рекомендації, з метою підвищення їхньої безпеки під час подорожей.

4.4. Висновки до розділу 4

В результаті проведених досліджень розроблено оптимальну архітектуру інформаційної технології для ML задач при малих та середніх вхідних даних. Що дозволило планування безпечних туристичних подорожей іншими системами з гнучкими можливостями інтеграції. Реалізація архітектури була реалізована за допомогою сервісно-орієнтованої архітектури (SOA), використання проміжного програмного забезпечення та веб-сервісів, що на відміну від інших систем забезпечило гнучкість, масштабованість та легкість інтеграції з іншими сервісами та вже

наявними системами. Це дало змогу створити максимально ефективну і адаптивну систему, яка відповідає актуальним потребам без надлишковості реалізації притаманній сучасній методології MLOps та побудови платформ і систем з ML. Оптимізований розроблений підхід є ефективним для малих і великих вхідних даних.

В результаті проведених досліджень:

- Вдосконалено MLOps методологію і підхід для систем з малим і середнім обсягом вхідних даних з урахуванням передачі датафреймів на проміжних етапах в інференсах. Запропонована технологія складається з денормалізацією етапів виконання інференесу суперечить принципу не повторюваності (DRY) для побудови програмного забезпечення, при дотриманні якого виростає складність, тим самим порушуючи принцип не ускладнювати виконання програмного потоку (KISS). Для виконання етапів інфернесу використано імперативний підхід на відміну від декларативного підходу.
- Оптимізовано UML модель для побудови програмного забезпечення в рамках запропонованої технології за допомогою використання координатора, що на відміну від ML платформ або платформ Big Data призначених для кола конкретно визначених задач, є платформою все в одному і оптимальне для використання при малих і середніх даних.
- Вдосконалено підходи для реалізації платформи, включаючи напрацювання SDLC, DevOps для кінцевого користувача шляхом реалізації REST API, що дає змогу інтеграцію з іншими програмними продуктами або рішеннями.
- Вдосконалено підхід MLOps, отримано його оптимальний варіант для реалізації з використанням малих і великих даних вхідних даних, включаючи напрацювання SDLC та Data Science. Що дало

змогу отримати варіацію MLOps підходів, від загального до конкретного.

– Застосовані сучасні метод сервіс орієнтованої архітектури (SOA) для побудови технологій і програмної реалізації в рази зменшують складність побудови і операційної діяльності для технології. Це дозволило на відміну від існуючих підходів зосередитись на реалізації прогнозування за допомогою моделей, ідентифікацій та мінімізації специфічних загроз. Що забезпечило більш ефективне використання обчислювального ресурсу (hardware resource). Також використання SOA в рази зменшило час побудови технології і платформи, складність за рахунок перевикористання наявних сервісів.

Основні наукові результати розділу опубліковані в працях [83, 84].

ВИСНОВКИ

У дисертаційній роботі поставлено і розв’язано актуальну науково-прикладну задачу розробки інноваційної інформаційної технології для планування безпечних туристичних подорожей, призначеної для організації безпечних туристичних поїздок. Основу цієї системи становлять розробленні методи прогнозування природних небезпек, включно з лісовими пожежами, ураганами та повенями, які реалізовані за допомогою штучного інтелекту. Інтеграція цих моделей у інформаційні системи дозволяє оцінювати потенційні загрози та інформувати мандрівників про ризики в різних частинах світу. Головна ціль полягає в підвищенні безпеки та освіченості туристів, надаючи їм можливість адаптуватися до змін умов середовища і відповідно коригувати свої маршрути, що в свою чергу знижує шанси на негативний вплив природних катастроф на їхній туристичний досвід.

Основні результати дисертаційної роботи:

- Проведено огляд наукових досліджень підтвердив важливість індивідуалізації інформаційних систем для планування подорожей, використання сучасних технологічних рішень для оперативного попередження про небезпеки, інтеграції супутникових даних та математичних моделей для точнішого прогнозування природних катастроф, таких як лісові пожежі, урагани та паводки. Аналіз літературних джерел дозволив визначити потребу в удосконаленні інформаційних систем, зокрема в плані вдосконалення та інтеграції новітніх методів прогнозування кризових явищ, що, в свою чергу, забезпечує більшу безпеку та інформованість користувачів, зменшуючи ризик негативного впливу природних катастроф на досвід подорожей.
- Проведено дослідження впливу сонячної активності на лісові пожежі з шляхом вдосконалення ANFIS, ANN та LSTM моделей. Аналіз даних

сонячної активності за допомогою кореляційного та лагового аналізу з використанням сплайн-інтерполяції дозволив виявити часові затримки (лаг 4 дні) між піками сонячної активності та виникненням лісових пожеж. Запропоновані моделі, що враховують нелінійні взаємодії між параметрами сонячної активності та атмосферними умовами, значно покращили точність прогнозування пожеж. Моделі на основі ANFIS досягли точності 87% (лаг 2) для малих і 93% (лаг 1) для великих пожеж, тоді як LSTM забезпечили точність 92% (лаг 3) для малих і 87% (лаг 2) для великих пожеж. Коефіцієнти кореляції для прогнозу температури та вологості склали 0,91 і 0,92 відповідно, що свідчить про високу надійність моделей. Моделі можуть інтегруватись у системи планування подорожей для підвищення точності прогнозування кризових явищ, пов'язаних із лісовими пожежами.

- Проведено дослідження впливу сонячної активності на урагани з шляхом вдосконалення LSTM, нейронних мереж (ANN) та лінійних моделей. Аналіз даних сонячної активності за допомогою лагового аналізу (5 днів) дозволив встановити взаємозв'язок між піками сонячної активності та інтенсивністю ураганів. Для цього використовувалися дані ураганів IRMA, JOSE та KATIA. Моделі LSTM показали найкращі результати з точністю відтворення динаміки ураганів ($R^2 = 0,99$), в той час як ANN досягли точності $R^2 = 0,90$, а лінійні моделі — $R^2 = 0,86$. Прогнозування на основі піків сонячної активності дало можливість з високою точністю (до 92%) прогнозувати поведінку ураганів до чотирьох днів наперед. Ці результати дозволяють значно покращити ефективність прогнозування ураганів і пропонують нові можливості для раннього попередження про такі природні катастрофи.
- Шляхом вдосконалення ансамблей класифікаційних моделей та моделей на основі дерев рішень проведено дослідження впливу сонячної активності на паводки. Запропоновані моделі забезпечили

точність до 97% (на 1 день вперед) і 92% (на 9 днів вперед). Врахування нелінійних взаємодій між параметрами сонячної активності та атмосферними умовами дозволило краще передбачати екстремальні погодні умови, що призводять до паводків.

– У роботі було вдосконалено MLOps технологію для систем з малим і середнім обсягом вхідних даних. Запропонований підхід використовує імперативну модель, що суперечить традиційним принципам програмування (DRY), але дозволяє знизити складність виконання програмних потоків. Крім того, було оптимізовано UML модель для побудови програмного забезпечення через інтеграцію координатора, що робить запропоновану технологію універсальною платформою “все в одному”, особливо ефективною для роботи з малими і середніми даними.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Garcia, A., O. Arbelaitz, M.T. Linaza, P. Vansteenwegen, and W. Souffriau. 2010. "Personalized Tourist Route Generation." In *Current Trends in Web Engineering*, edited by F. Daniel and F.M. Facca, 6385: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-642-16985-4_47.
2. Zhuang, Yu, Shuili Yang, Asif Razzaq, and Zeeshan Khan. 2021. "Environmental Impact of Infrastructure-Led Chinese Outward FDI, Tourism Development and Technology Innovation: A Regional Country Analysis." *Journal of Environmental Planning and Management* 0 (0): 1-33.
3. Zhen, S., and W. Gao. 2017. "Geological Tourist Route Planning of Henan Province Based on Geological Relics Zoning." *Geology, Ecology, and Landscapes* 1 (1): 66-69.
4. Khamsing, N., K. Chindaprasert, R. Pitakaso, W. Sirirak, and C. Theeraviriya. 2021. "Modified ALNS Algorithm for a Processing Application of Family Tourist Route Planning: A Case Study of Buriram in Thailand." *Computation* 9 (2): 23.
5. Gavalas, D., C. Konstantopoulos, K. Mastakas, G. Pantziou, and N. Vathis. 2015. "Heuristics for the Time Dependent Team Orienteering Problem: Application to Tourist Route Planning." *Computers & Operations Research* 62: 36-50.
6. Ayala, I., L. Mandow, M. Amor, and L. Fuentes. 2017. "A Mobile and Interactive Multiobjective Urban Tourist Route Planning System." *Journal of Ambient Intelligence and Smart Environments* 9 (1): 129-144.
7. Ayala, I., L. Mandow, M. Amor, and L. Fuentes. 2012. "An Evaluation of Multiobjective Urban Tourist Route Planning with Mobile Devices." In *International Conference on Ubiquitous Computing and Ambient Intelligence*, 387-394. Berlin, Heidelberg: Springer.

8. Zhou, X., B. Sun, S. Li, and S. Liu. 2020. "Tour Route Planning Algorithm Based on Precise Interested Tourist Sight Data Mining." *IEEE Access* 8: 153134-153168.
9. Nadi, S., and M. R. Delavar. 2011. "Multi-criteria, Personalized Route Planning Using Quantifier-Guided Ordered Weighted Averaging Operators." *International Journal of Applied Earth Observation and Geoinformation* 13 (3): 322-335.
10. Nguyen, M.D., X.B. To, and H. Le. 2023. "Integration of Mobile and Meb GIS Technologies to Promote Smart and Sustainable Tourism in Vietnam." *Inzynieria Mineralna* (2). https://doi.org/10.29227/IM-2023-02-34.
11. Cena, F., N. Mauro, L. Ardissono, F. Ferrero, S. Ferrigno, A. Rapp, C. Mattutino, and R. Keller. 2023. "CARES: An Inclusive Personalized Touristic System for Autism." In *UMAP 2023 – Adjunct Proceedings of the 31st ACM Conference on User Modeling, Adaptation and Personalization*, 363-366. https://doi.org/10.1145/3563359.3596665.
12. Chaidee, K., K. Boontasri, E. Chaidee, W. Wongchai, S. Phamon, and S. Preedee. 2023. "Chiang Rai Phra That Nine Choms Attraction Promoting Mobile Application for Cultural Tourism." In *2023 20th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, 189696. https://doi.org/10.1109/ECTI-CON58255.2023.10153256.
13. Escudeiro, N., P. Escudeiro, B. Cunha, and M.C. Gouveia. 2023. "Inclusive Cultural Heritage Tourism." In *Smart Innovation, Systems and Technologies*, vol. 340, 207-217. https://doi.org/10.1007/978-981-19-9960-4_19.
14. Zejda, D., and M. Pásková. 2023. "Destination Management Support System and Intelligent Destination Guide for Natural Destinations." In *Public Recreation and Landscape Protection – With Environment Hand in Hand? Proceedings of the 14th Conference*, 130-134. https://doi.org/10.11118/978-80-7509-904-4-0130.

15. Vyklyuk, Y., M.M. Radovanović, G. Stanojević, M.D. Petrović, N.B. Ćurčić, M. Milenković, S.M. Milićević, B. Milovanović, A.A. Yamashkin, A.M. Pešić, and D. Lukić. 2020. "Connection of Solar Activities and Forest Fires in 2018: Events in the USA (California), Portugal, and Greece." *Sustainability (Switzerland)* 12 (24): art. no. 10261. https://doi.org/10.3390/su122410261.
16. Iwamoto, K., N. Suenaga, S. Yoshioka, and F. Ortega-Culaciati. 2024. "3D Thermal Structural and Dehydration Modeling in the Southern Chile Subduction Zone and Its Relationship to Interplate Earthquakes and the Volcanic Chain." *Geoscience Letters* 11 (1): art. no. 3. https://doi.org/10.1186/s40562-023-00318-2.
17. Ripepe, M., and G. Lacanna. 2024. "Volcano Generated Tsunami Recorded in the Near Source." *Nature Communications* 15 (1): art. no. 1802. https://doi.org/10.1038/s41467-024-45937-1.
18. de Vasconcelos, L.M.T., H.G. Borba, N.M.G. Neto, A.C.S. Alexandre, J.L. de Araújo Veras, and S.E.G. de Medeiros. 2021. "Factors Associated with Shark Attacks and Deaths: A Cross-Sectional Study." *Online Brazilian Journal of Nursing* 20: 1-11. https://doi.org/10.17665/1676-4285.20216506.
19. Kono, I.S., V.C.F. Pandolfi, M.N.A.D. Marchi, N. Freitas, R.L. Freire. 2024. "Unveiling the Secrets of Snakes: Analysis of Environmental, Socioeconomic, and Spatial Factors Associated with Snakebite Risk in Paraná, Southern Brazil." *Toxicon* 237: art. no. 107552. https://doi.org/10.1016/j.toxicon.2023.107552.
20. Furusawa, M., and S. Inukai. 2019. "The Great East Japan Earthquake (2011): Using the One Health Approach to Minimise the Impact on the Livestock Industry and Human Health." *Revue scientifique et technique (International Office of Epizootics)* 38 (1): 103–111. https://doi.org/10.20506/rst.38.1.2945.

21. Kazemi, Z., A.J. Jafari, M. Farzadkia, J. Hosseini, P. Amini, A. Shahsavani, M. Kermani. 2024. "Estimating the Health Impacts of Exposure to Air Pollutants and the Evaluation of Changes in Their Concentration Using a Linear Model in Iran." *Toxicology Reports* 12: 56-64. https://doi.org/10.1016/j.toxrep.2023.12.006.
22. Rahman, G.R., S.Y. Liang, L. Tian, S.S. Sin, G.N. Jasani. 2024. "Trends and Characteristics of Terrorist Attacks Against Nightclub Venues over 5 Decades." *Disaster Medicine and Public Health Preparedness* 18: art. no. e12. https://doi.org/10.1017/dmp.2023.236.
23. Zhang, H., L. Jiao, C. Li, Z. Deng, Z. Wang, Q. Jia, X. Lian, Y. Liu, Y. Hu. 2024. "Global Environmental Impacts of Food System from Regional Shock: Russia-Ukraine War as an Example." *Humanities and Social Sciences Communications* 11 (1): art. no. 191. https://doi.org/10.1057/s41599-024-02667-5.
24. Haghani, M., M. Coughlan, B. Crabb, A. Dierickx, C. Feliciani, R. van Gelder, P. Geoerg, N. Hocaoglu, S. Laws, R. Lovreglio, Z. Miles, A. Nicolas, W.J. O'Toole, S. Schaap, T. Semmens, Z. Shahhoseini, R. Spaaij, A. Tatrai, J. Webster, and A. Wilson. 2023. "A Roadmap for the Future of Crowd Safety Research and Practice: Introducing the Swiss Cheese Model of Crowd Safety and the Imperative of a Vision Zero Target." *Safety Science* 168: art. no. 106292. https://doi.org/10.1016/j.ssci.2023.106292.
25. Heinselman, P.L., P.C. Burke, L.J. Wicker, A.J. Clark, J.S. Kain, J. Gao, N. Yussouf, T.A. Jones, P.S. Skinner, C.K. Potvin, K.A. Wilson, B.T. Gallo, M.L. Flora, J. Martin, G. Creager, K.H. Knopfmeier, Y. Wang, B.C. Matilla, D.C. Dowell, E.R. Mansell, B. Roberts, K.A. Hoogewind, D.R. Stratman, J. Guerra, A.E. Reinhart, C.A. Kerr, and W. Miller. 2024. "Warn-on-Forecast System: From Vision to Reality." *Weather and Forecasting* 39 (1): 75–95. https://doi.org/10.1175/WAF-D-23-0147.1.

26. Atiyah, A.H., I.B. Falih, and Z.O. Ibraheem. 2020. "The Combination of Histopathology and Analysis of Accuweather Survey of Canine Spirocercosis in Tuwerij Area, Karbala Governorate, Iraq." *International Journal of Pharmaceutical Research* 12 (1): 993–1000. https://doi.org/10.31838/IJPR/2020.12.01.186.
27. Windy. 2024. Accessed [date]. https://www.windy.com/.
28. Radovanovic, M., and J.F.P. Gomes. 2009. *Solar Activity and Forest Fires*. Nova Science Publishers Inc.
29. Nikolov, N. 2006. "Global Forest Resources Assessment 2005 – Report on Fires in the Balkan Region." Forestry Department, FAO of the UN, Fire Management Working Papers FM/11/E. Accessed December 26, 2013. http://www.fao.org/docrep/009/j7567e/j7567e00.htm.
30. Schmuck, G., J. San-Miguel-Ayanz, A. Camia, T. Durrant, R. Boca, C. Whitmore, et al. 2012. *Forest Fires in Europe, Middle East and North Africa 2011*. Joint Research Centre of the European Commission. Accessed December 26, 2013. http://forest.jrc.ec.europa.eu/media/cms_page_media/9/forest-fires-in-europe-2011.pdf.
31. Hall, L.B. 2007. "Precipitation Associated with Lightning-Ignited Wildfires in Arizona and New Mexico." *International Journal of Wildland Fire* 16 (2): 242–254.
32. Kourtz, P.H., and J.B. Todd. 1991. "Predicting the Daily Occurrence of Lightning-Caused Forest Fires." Forestry Canada, Petawawa National Forestry Institute, Chalk River, Ontario. Information Report PI-X-112. Accessed December 26, 2013. http://cfs.nrcan.gc.ca/publications/?id=10706.
33. Sannikov, S.N., A.I. Zakharov, L.G. Smol'nikova, and N.S. Sannikova. 2010. "Forest Fires Caused by Lightning as an Indicator of Connections Between Atmosphere, Lithosphere, and Biosphere." *Russian Journal of Ecology* 41 (1): 1–6.

34. Cumming, S.G. 2001. "Forest Type and Wildfire in the Alberta Boreal Mixedwood: What Do Fires Burn?" *Ecological Applications* 11 (1): 97-110.
35. Wotton, M.B., J.B. Stocks, and L.D. Martell. 2005. "An Index for Tracking Sheltered Forest Floor Moisture Within the Canadian Forest Fire Weather Index System." *International Journal of Wildland Fire* 14 (2): 169-182.
36. Viegas, D.X. 1998. "Forest Fire Propagation." *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 356: 2907–2928.
37. Guyette, P.R., C.M. Stambaugh, C.D. Dey, and R-M. Muzika. 2012. "Predicting Fire Frequency with Chemistry and Climate." *Ecosystems* 15 (2): 322–335.
38. Gomes, J.F.P., and M. Radovanovic. 2008. "Solar Activity as a Possible Cause of Large Forest Fires – A Case Study: Analysis of the Portuguese Forest Fires." *Science of the Total Environment* 394 (1): 197–205.
39. SolarMonitor.org. 2017. "AIA 193Å 20170827 19:27." [Digital Image]. Accessed [date]. https://solarmonitor.org/full_disk.php?date=20170827&type=saia_00193&region.
40. Japan Aerospace Exploration Agency Earth Observation Research Center. 2017. "Tropical Cyclones Track 2017 Season." [Data set]. Accessed [date]. http://sharaku.eorc.jaxa.jp/cgi-bin/typ_db/typ_track.cgi?lang=e&area=AT.
41. Frank, M.W., and S.G. Young. 2007. "The Interannual Variability of Tropical Cyclones." *Monthly Weather Review* 135 (10): 3587–3598. https://doi.org/10.1175/MWR3435.1.
42. Morozova, A.L., M.I. Pudovkin, and P. Thejll. 2002. "Variations of Atmospheric Pressure During Solar Proton Events and Forbush Decreases for Different Latitudinal and Synoptic Zones." *International Journal of Geomagnetism and Aeronomy* 3 (2): 181–189. Accessed [date].

http://elpub.wdcb.ru/journals/ijga/v03/gai00369/gai00369.htm.

43. Hodges, R.E., T.H. Jagger, and J.B. Elsner. 2014. "The Sun-Hurricane Connection: Diagnosing the Solar Impacts on Hurricane Frequency Over the North Atlantic Basin Using a Space–Time Model." *Natural Hazards* 73 (2): 1063–1084. https://doi.org/10.1007/s11069-014-1120-9.
44. Elsner, J. B., and S. P. Kavlakov. 2001. "Hurricane Intensity Changes Associated with Geomagnetic Variation." *Atmospheric Science Letters* 2 (1–4): 86–93. http://dx.doi.org/10.1006/asle.2001.0043.
45. Hodges, R., and J. Elsner. 2012. "The Spatial Pattern of the Sun-Hurricane Connection Across the North Atlantic." *ISRN Meteorology* 2012: 517962. http://dx.doi.org/10.5402/2012/517962.
46. Elsner, J., T. Jagger, M. Dickinson, and D. Rowe. 2008. "Improving Multiseason Forecasts of North Atlantic Hurricane Activity." *Journal of Climate* 21 (6): 1209–1219. https://doi.org/10.1175/2007JCLI1731.1.
47. Haigh, J. D. 1996. "The Impact of Solar Variability on Climate." *Science* 272 (5264): 981–984. https://doi.org/10.1126/science.272.5264.981.
48. Prikryl, P., R. Bruntz, T. Tsukijihara, K. Iwao, D. B. Muldrew, V. Rušin, et al. 2017. "Tropospheric Weather Influenced by Solar Wind Through Atmospheric Vertical Coupling Downward Control." *Journal of Atmospheric and Solar-Terrestrial Physics*. Advance online publication. https://doi.org/10.1016/j.jastp.2017.07.023.
49. Prikryl, P., K. Iwao, D. Muldrew, V. Rusin, M. Rybansky, and R. Bruntz. 2016. "A Link Between High-Speed Solar Wind Streams and Explosive Extratropical Cyclones." *Journal of Atmospheric and Solar-Terrestrial Physics* 149: 219–231. http://dx.doi.org/10.1016/j.jastp.2016.04.002.
50. Prikryl, P., D. B. Muldrew, and G. J. Sofko. 2009. "The Influence of Solar Wind on Extratropical Cyclones – Part 2: A Link Mediated by Auroral Atmospheric Gravity Waves?" *Annales Geophysicae* 27: 31–57. http://dx.doi.org/10.5194/angeo-27-31-2009.

51. Prikryl, P., V. Rusin, and M. Rybanský. 2009. "The Influence of Solar Wind on Extratropical Cyclones – Part 1: Wilcox Effect Revisited." *Annales Geophysicae* 27: 1–30. http://dx.doi.org/10.5194/angeo-27-1-2009.
52. Veretenenko, S. V. 2017. "Comparative Analysis of Short-Term Effects of Solar and Galactic Cosmic Rays on the Evolution of Baric Systems at Middle Latitudes." *Geomagnetism and Aeronomy* 81 (2): 281–284. https://doi.org/10.3103/S1062873817020460.
53. Veretenenko, S., and P. Thejll. 2004. "Effects of Energetic Solar Proton Events on the Cyclone Development in the North Atlantic." *Journal of Atmospheric and Solar-Terrestrial Physics* 66 (5): 393–405. https://doi.org/10.1016/j.jastp.2003.11.005.
54. Artamonova, I. V., and S. V. Veretenenko. 2013. "Effect of Solar and Galactic Cosmic Rays on the Duration of Macrosynoptic Processes." *Geomagnetism and Aeronomy* 53 (1): 5–9. https://doi.org/10.1134/S0016793213010039.
55. Mendoza, B., and M. Pazos. 2009. "A 22 yr Hurricane Cycle and Its Relation with Geomagnetic Activity." *Journal of Atmospheric and Solar-Terrestrial Physics* 71 (17–18): 2047–2054. https://doi.org/10.1016/j.jastp.2009.09.012.
56. Wheeler, D. 2001. "A Verification of UK Gale Forecasts by the 'Solar Weather Technique': October 1995–September 1997." *Journal of Atmospheric and Solar-Terrestrial Physics* 63 (1): 29–34. https://doi.org/10.1016/S1364-6826(00)00155-3.
57. Vyklyuk, Y., M. Radovanović, B. Milovanović, T. Leko, M. Milenković, Z. Milošević, et al. 2017. "Hurricane Genesis Modelling Based on the Relationship Between Solar Activity and Hurricanes." *Natural Hazards* 85 (2): 1043–1062. https://doi.org/10.1007/s11069-016-2620-6.
58. Vyklyuk, Y., M.M. Radovanović, G.B. Stanojević, B. Milovanović, T. Leko, M. Milenković, et al. 2017. "Hurricane Genesis Modelling Based

on the Relationship Between Solar Activity and Hurricanes II." *Atmospheric and Solar-Terrestrial Physics*. Advance online publication. https://doi.org/10.1016/j.jastp.2017.09.008.

59. Gaume, E., V. Bain, P. Bernardara, O. Newinger, M. Barbuc, A. Bateman, L. Blaškovičová, G. Bloschl, M. Borga, A. Dumitrescu, et al. 2009. "A Compilation of Data on European Flash Floods." *Journal of Hydrology* 367: 70–78.
60. Nitka, Weronika, and Krzysztof Burnecki. 2019. "Impact of Solar Activity on Precipitation in the United States." *Physica A* 527: 121387.
61. Milovanović, B., and M. Radovanović. 2009. "Повезаност Сунчеве активности и циркулације атмосфере у периоду 1891-2004." *Journal of the Geographical Institute 'Jovan Cvijić' SASA* 59/1: 35-48.
62. Ma, L.H., Y.B. Han, and Z.Q. Yin. 2010. "Possible Influence of the 11-Year Solar Cycle on Precipitation in Huashan Mountain of China Over the Last 300 Years." *Earth Moon Planets* 107: 219–224. http://dx.doi.org/10.1007/s11038-010-9367-y.
63. Wilcox, J.M., P.H. Scherrer, L. Svalgaard, W.O. Roberts, and R.H. Olson. 1973. "Solar Magnetic Sector Structure: Relation to Circulation of the Earth's Atmosphere." *Science* 180: 185–186.
64. Wilcox, J.M., P.H. Scherrer, L. Svalgaard, W.O. Roberts, R.H. Olson, and R.L. Jenne. 1974. "Influence of Solar Magnetic Sector Structure on Terrestrial Atmospheric Vorticity." *Journal of Atmospheric Sciences* 31: 581–588.
65. Maliniemi, V., T. Asikainen, and K. Mursula. 2018. "Decadal Variability in the Northern Hemisphere Winter Circulation: Effects of Different Solar and Terrestrial Drivers." *Journal of Atmospheric and Solar-Terrestrial Physics* 179: 40–54. http://dx.doi.org/10.1016/j.jastp.2018.06.012.
66. Vyklyuk, Y., M. Radovanovic, B. Milovanovic, T. Leko, M. Milenkovic, Z. Milošević, A. Milanovic Pesic, D. Jakovljevic. 2016. "Hurricane

Genesis Modelling Based on the Relationship Between Solar Activity and Hurricanes." *Natural Hazards*. http://dx.doi.org/10.1007/s11069-016-2620-6.

67. Vyklyuk, Y., M.M. Radovanović, G.B. Stanojević, B. Milovanović, T. Leko, M. Milenković, M. Petrović, A.A. Yamashkin, A. Milanović Pešić, D. Jakovljević, S. Malinovic Milicevic. 2017. "Hurricane Genesis Modelling Based on the Relationship Between Solar Activity and Hurricanes II." *Journal of Atmospheric and Solar-Terrestrial Physics* 180: 159–164. https://doi.org/10.1016/j.jastp.2017.09.008.
68. Srećković, V., M. Šulić, V. Vujčić, D. Jevremović, V. Vyklyuk. 2017. "The Effects of Solar Activity: Electrons in the Terrestrial Lower Ionosphere." *Journal of the Geographical Institute 'Jovan Cvijić'* 67 (3): 221–233.
69. Nina, A., V.M. Čadež, J. Bajčetić, M. Andrić, G. Jovanović. 2017. "Responses of the Ionospheric D-Region to Periodic and Transient Variations of the Ionizing Solar Lyα Radiation." *Journal of the Geographical Institute 'Jovan Cvijić'* 67 (3): 235–248.
70. Haigh, J.D. 1996. "The Impact of Solar Variability on Climate." *Science* 272: 981–984.
71. Svensmark, H., and E. Friis-Christensen. 1997. "Variation of Cosmic Ray Flux and Global Cloud Coverage—A Missing Link in Solar–Climate Relationships." *Journal of Atmospheric and Solar-Terrestrial Physics* 59: 1225–1232.
72. Carslaw, K.S., R.G. Harrison, and J. Kirkby. 2002. "Cosmic Rays, Clouds, and Climate." *Science* 298: 1732–1737.
73. Gray, L.J., M. Beer, J. Geller, J.D. Haigh, M. Lockwood, K. Matthes, U. Cubasch, D. Fleitmann, G. Harrison, L. Hood, J. Luterbacher, G.A. Meehl, D. Shindell, B. van Geel, and W. White. 2010. "Solar Influences on Climate." *Reviews of Geophysics* 48: RG4001. http://dx.doi.org/10.1029/2009RG000282.

74. Solheim, J.-E., K. Stordahl, and O. Humlum. 2012. "The Long Sunspot Cycle 23 Predicts a Significant Temperature Decrease in Cycle 24." *Journal of Atmospheric and Solar-Terrestrial Physics* 80: 267–284. http://dx.doi.org/10.1016/j.jastp.2012.02.008.
75. Veretenenko, S., and P. Thejll. 2013. "Influence of Energetic Solar Proton Events on the Development of Cyclonic Processes at Extratropical Latitudes." *Journal of Physics: Conference Series* 409: 012237. http://dx.doi.org/10.1088/1742-6596/409/1/012237.
76. Veretenenko, S., and P. Thejll. 2004. "Effects of Energetic Solar Proton Events on the Cyclone Development in the North Atlantic." *Journal of Atmospheric and Solar-Terrestrial Physics* 66 (5): 393–405.
77. Bhattacharyya, S., and R. Narasimha. 2005. "Possible Association Between Indian Monsoon Rainfall and Solar Activity." *Geophysical Research Letters* 32: L05813.
78. Kirby, C., and T.J. Marsh, eds. 1990. *Water Quality in the Environment*. Swindon: Natural Environment Research Council, 34.
79. UK Centre for Ecology and Hydrology. 2022. "National River Flow Archive." Accessed February 15, 2022. https://nrfa.ceh.ac.uk/uk-river-flow-regimes.
80. Kingston, D.G., G.R. McGregor, D.M. Hannah, and D.M. Lawler. 2007. "Large-Scale Climatic Controls on New England River Flow." *Journal of Hydrometeorology* 8 (3): 367–379. doi: 10.1175/JHM584.1.
81. Laizé, C.L.R., and D.M. Hannah. 2010. "Modification of Climate–River Flow Associations by Basin Properties." *Journal of Hydrology* 389: 186–204. DOI:10.1016/j.jhydrol.2010.05.048.
82. Hannaford, J., and G. Buys. 2012. "Trends in Seasonal River Flow Regimes in the UK." *Journal of Hydrology* 475: 158–174.
83. Шаховська, Н., Сидор П.. 2022. "Розроблення архітектури системи планування безпечних туристичних подорожей." *Вісник Хмельницького*

національного університету. Технічні науки. №1 (305): 96-101.

84. Сидор, П.О., Виклюк Я.І.. 2024. "Мобільна система інформаційної підтримки з рекомендаціями для безпечних подорожей." *Науковий вісник НЛТУ України* 34 (3) с. 103-109.
85. Hazewinkel, M. 2013. *Encyclopaedia of Mathematics: Monge – Ampère Equation – Rings and Algebras*. Springer. ISBN: 978-0-7923-2976-3.
86. Radovanović, M. 2010. "Forest Fires in Europe from July 22nd to 25th 2009." *Archives of Biological Sciences* 62 (2): 419-424.
87. Ducic, V., M. Milenkovic, and M. Radovanovic. 2008. "Contemporary Climate Variability and Forest Fires in Deliblatska Pescara." *Journal of the Geographical Institute Jovan Cvijic SASA* No. 58: 59-74.
88. Radovanović, M. 2010. "Forest Fires in Europe from July 22nd to 25th 2009." *Archives of Biological Sciences* 62 (2): 419-424.
89. Radovanovic, M. 2011. "Solar Activity – Climate Change and Natural Disasters in Mountain Regions." In *Sustainable Development in Mountain Regions*, edited by Zhelezov G. Springer Science+Business Media B.V., 9-17.
90. Boxall, M. et al. 2009. *ESS Guidelines on Seasonal Adjustment*. Eurostat. Accessed December 26, 2013. http://epp.eurostat.ec.europa.eu/cache/ITY_OFFPUB/KS-RA-09-006/EN/KS-RA-09-006-EN.PDF.
91. Bell, W.R., S.H. Holan, and T.S. McElroy. 2012. *Economic Time Series: Modeling and Seasonality*. Chapman and Hall/CRC.
92. Hansen, B.E. 2014. *Econometrics*. University of Wisconsin, Department of Economics, p. 378.
93. Labitzke, K. 2003. "The Global Signal of the 11-Year Sunspot Cycle in the Atmosphere: When Do We Need the QBQ?" *Meteorologische Zeitschrift* 12 (4): 209-216.
94. Lenskiy, A.A., and S. Seol. 2012. "The Analysis of R/S Estimation Algorithm with Applications to WiMAX Network Traffic." *International*

Journal of Multimedia and Ubiquitous Engineering 7 (3): 27-34.

95. Viegas, D.X. 1998. "Forest Fire Propagation." *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 356: 2907–2928.

96. Özger, M. 2011. "Prediction of Ocean Wave Energy from Meteorological Variables by Fuzzy Logic Modeling." *Expert Systems with Applications* 38 (5): 6269-6274.

97. Peters, E.E. 2003. *Fractal Market Analysis: Applying Chaos Theory to Investment and Economics*. John Wiley & Sons.

98. Radovanović, M., Y. Vyklyuk, A. Jovanović, D. Vuković, M. Milenković, M. Stevančević, et al. 2013. "Examination of the Correlations Between Forest Fires and Solar Activity Using Hurst Index." *Journal of the Geographical Institute Jovan Cvijic SASA* 63 (3): 23-32.

99. Amini, M., K.C. Abbaspour, and C.A. Johnson. 2010. "A Comparison of Different Rule-Based Statistical Models for Modeling Geogenic Groundwater Contamination." *Environmental Modelling & Software* 25 (12): 1650-1657.

100. Bektas Ekici, B., and U.T. Aksoy. 2011. "Prediction of Building Energy Needs in Early Stage of Design by Using ANFIS." *Expert Systems with Applications* 38 (5): 5352-5358.

101. Güneri, A.F., T. Ertay, and A. Yücel. 2011. "An Approach Based on ANFIS Input Selection and Modeling for Supplier Selection Problem." *Expert Systems with Applications* 38 (12): 14907-14917.

102. Kurtulus, B., and N. Flipo. 2012. "Hydraulic Head Interpolation Using ANFIS—Model Selection and Sensitivity Analysis." *Computers & Geosciences* 38 (1): 43-51.

103. Shiri, J., O. Kisi, H. Yoon, K.-K. Lee, and A.H. Nazemi. 2013. "Predicting Groundwater Level Fluctuations with Meteorological Effect Implications—A Comparative Study Among Soft Computing Techniques."

Computers & Geosciences 56: 32-44.

104. Soltani, F., R. Kerachian, and E. Shirangi. 2010. "Developing Operating Rules for Reservoirs Considering the Water Quality Issues: Application of ANFIS-Based Surrogate Models." *Expert Systems with Applications* 37 (9): 6639-6645.

105. Talebizadeh, M., and A. Moridnejad. 2011. "Uncertainty Analysis for the Forecast of Lake Level Fluctuations Using Ensembles of ANN and ANFIS Models." *Expert Systems with Applications* 38 (4): 4126-4135.

106. Tan, Z., C. Quek, and P.Y.K. Cheng. 2011. "Stock Trading with Cycles: A Financial Application of ANFIS and Reinforcement Learning." *Expert Systems with Applications* 38 (5): 4741-4755.

107. Yilmaz, I., and O. Kaynar. 2011. "Multiple Regression, ANN (RBF, MLP) and ANFIS Models for Prediction of Swell Potential of Clayey Soils." *Expert Systems with Applications* 38 (5): 5958-5966.

108. Abraham, A. 2005. "Adaptation of Fuzzy Inference System Using Neural Learning." In *Fuzzy Systems Engineering: Theory and Practice*, edited by N. Nedjah and L. de Macedo Mourelle. Springer Verlag, 53–83.

109. Bektas Ekici, B., and U.T. Aksoy. 2011. "Prediction of Building Energy Needs in Early Stage of Design by Using ANFIS." *Expert Systems with Applications* 38 (5): 5352-5358.

110. Vyklyuk, Y. 2013. "Simulation of Spatial Form of Urban Systems by Diffusion Methods." *Journal of the Geographical Institute 'Jovan Cvijic' SASA* 63, #1: 89–100, #2: 67–77.

111. Hang, Houjun, Xing Yao, Qingqing Li, and Michel Artiles. 2017. "Cubic B-Spline Curves with Shape Parameter and Their Applications." *Mathematical Problems in Engineering*, Article ID 3962617, 7 pages.

112. Hyndman, Rob J. 2010. "Moving Averages." In *International Encyclopedia of Statistical Science*, edited by Miodrag Lovric, Springer, 866-869.

113. Miljanovic, Milos. 2012. "Comparative Analysis of Recurrent and Finite Impulse Response Neural Networks in Time Series Prediction." *Indian Journal of Computer Science and Engineering (IJCSE)* Vol. 3 No. 1 Feb-Mar: 180-191.

114. Kingma, Diederik P., and Jimmy Ba. 2014. "Adam: A Method for Stochastic Optimization." arXiv preprint arXiv:1412.6980.

115. Greff, K., R.K. Srivastava, J. Koutník, B.R. Steunebrink, and J. Schmidhuber. 2017. "LSTM: A Search Space Odyssey." *IEEE Transactions on Neural Networks and Learning Systems* 28 (10): 2222–2232. https://ieeexplore.ieee.org/document/7508408.

116. Safavian, S.R., and D. Landgrebe. 1991. "A Survey of Decision Tree Classifier Methodology." *IEEE Transactions on Systems, Man, and Cybernetics* 21 (3): 660-674.

117. Raileanu, Laura Elena, and Kilian Stoffel. 2004. "Theoretical Comparison between the Gini Index and Information Gain Criteria." *Annals of Mathematics and Artificial Intelligence* 41: 77–93. doi: 10.1023/B:AMAI.0000018580.96245.c6.

118. Graczyk, M., T. Lasota, B. Trawiński, and K. Trawiński. 2010. "Comparison of Bagging, Boosting and Stacking Ensembles Applied to Real Estate Appraisal." In *Intelligent Information and Database Systems*, edited by N.T. Nguyen, M.T. Le, and J. Świątek, Lecture Notes in Computer Science, vol 5991. Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-642-12101-2_35.

119. Trostianchyn, A., Z. Duriagina, I. Izonin, R. Tkachenko, V. Kulyk, and O. Pavliuk. 2021. "Sm-Co Alloys Coercivity Prediction Using Stacking Heterogeneous Ensemble Model." *Acta Metallurgica Slovaca* 27 (4): 195-202. https://doi.org/10.36547/ams.27.4.1173.

120. Malinović-Milićević S., Vyklyuk Y, Radovanović M.M., Milenković M., Milanović A.P., Milovanović B., Popović T., Sydor P., and Petrović M.D..

2024. "Applying Machine Learning in the Investigation of the Link Between the High-Velocity Streams of Charged Solar Particles and Precipitation-Induced Floods." *Environmental Monitoring and Assessment* 196: 400.

121. Сидор, П.О., and Я.І. Виклюк. 2024. "Ансамблеві моделі прогнозування повеней у Великій Британії на основі сонячної активності." *Вісник Хмельницького національного університету. Технічні науки.* 2024, №2 (333) с. 218-231.

122. Vyklyuk, Y., M.M. Radovanović, P. Sydor. 2018. "Hurricane Forecasting Using by Parallel Calculations & Machine Learning." In *2018 IEEE 1st International Conference on System Analysis and Intelligent Computing (SAIC)*, Proceedings, October 31, 2018, Kyiv, Ukraine, Article number 8516872.

123. Виклюк, Я.І., П.О. Сидор, Н. Е. Кунанець, В.В. Пасічник. 2018. "Прогнозування лісових пожеж на основі ANFIS та паралельних розрахунків." In *Інтелектуальні системи прийняття рішень і проблеми обчислювального інтелекту*, Херсон: Видавництво ФОП Вишемирський В. С., May 21-27, 2018, Залізний Порт, pp. 41-42

124. Сидор, П.О., Виклюк Я.І. "Прогнозування лісових пожеж в Португа–лії." In *Інтелектуальні системи прийняття рішень і проблеми обчислювального інтелекту: Матеріали міжнародної наукової конференції*.

125. Jang, J.-S.R., C.-T. Sun, and E. Mizutani. 1997. *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. Prentice Hall.

126. Vyklyuk, Y., Vuković D., and Jovanović A.. 2013. "Forex Prediction with Neural Network: USD/EUR Currency Pair." *Actual Problems of Economics* 2013, #10, 261-273.

127. Elith, J., J.R. Leathwick, and T. Hastie. 2008. "A Working Guide to Boosted Regression Trees." *Journal of Animal Ecology* 77: 802–813.

128. Pianosi, F., K. Beven, J. Freer, J.W. Hall, J. Rougier, D.B. Stephenson,

and T. Wagener. 2016. "Sensitivity Analysis of Environmental Models: A Systematic Review with Practical Workflow." *Environmental Modelling & Software* 79: 214–232.

129. Stevancevic, M., M. Radovanovic, and N. Todorovic. 2004. "The Possibility of Application of Electromagnetic Method in Mid-term Weather Forecasting." In *Collection of Papers EkoIst'04 Ecological Truth*, May 30 – June 2, 2004, Bor, 396-399. (in Serbian)

130. Stevancevic, M., M. Radovanovic, and N. Todorovic. 2006. "Analysis of Characteristic Mistakes in the Heliocentric Electromagnetic Long-term Forecast." In *Tourist Valorisation of Tara*, Theme Collection of Geographic Institute "Jovan Cvijic" Serbian Academy for Sciences and Art and Sport-Recreative Center Bajina Bašta, Belgrade, 101-110. (in Serbian)

131. Prikryl, P., R. Bruntz, T. Tsukijihara, K. Iwao, D.B. Muldrew, V. Rušin, M. Rybanský, M. Turˇ na, and P. Št'astný. 2018. "Tropospheric Weather Influenced by Solar Wind Through Atmospheric Vertical Coupling Downward Control." *Journal of Atmospheric and Solar-Terrestrial Physics* 171: 94–110.

132. Savchuk, V.V., N.E. Kunanec, V.V. Pasichnyk, P. Popiel, R. Weryńska-Bieniasz, G. Kashaganova, and A. Kalizhanova. 2017. "Safety Recommendation Component of Mobile Information Assistant of the Tourist." In *Proceedings on SPIE 10445, Photonics Applications in Astronomy, Communications, Industry, and High Energy Physics Experiments*, 104455Z. https://doi.org/10.117/12.2280833.

133. Vyklyuk, Y., M.M. Radovanović, G.B. Stanojević, B. Milovanović, T. Leko, M. Milenković, M. Petrović, A.A. Yamashkin, A. Milanović Pešić, D. Jakovljević, and S. Malinović. 2018. "Hurricane Genesis Modelling Based on the Relationship Between Solar Activity and Hurricanes II." *Journal of Atmospheric and Solar-Terrestrial Physics* 180: 159–164. https://doi.org/10.1016/j.jastp.2017.09.008

134. Horváth, A., and M. L. Lopatny. 2022. "Tourism Security And Taking Responsibility In The Shadow Of The Covid19 Pandemic – Who Is Responsible?" *Geojournal of Tourism and Geosites* 40 (1): 292–301. https://doi.org/10.30892/gtg.40135-831.

135. Moorthy, T. S. D., N. Nimalan, S. Sridevi, and B. Nevetha. 2024. "Prevention Equipment for COVID-19 Spread Using IoT and Multimedia-Based Solutions." In *Lecture Notes in Networks and Systems*, vol. 785, 105–119. https://doi.org/10.1007/978-981-99-6544-1_9.

136. Nikolaichuk, Olha. 2021. "Trends of Development of Domestic Hospitality Industry in COVID-19 Conditions." *The Journal of VN Karazin Kharkiv National University. Series: International Relations. Economics. Country Studies. Tourism* 13: 108–114. https://doi.org/10.26565/2310-9513-2021-13-11.

137. Ali, F., L. Ali, Z. Gao, A. Terrah, and G. Turktarhan. 2024. "Determinants of User's Intentions to Book Hotels: A Comparison of Websites and Mobile Apps." *Aslib Journal of Information Management* 76 (1): 16–41. https://doi.org/10.1108/AJIM-05-2022-0239.

138. Power, D. J., R. Sharda, and F. Burstein. 2015. "Model-driven Decision Support Systems: Concepts and Research Directions." *Decision Support Systems*, Vol. 43, Issue 3, 1044–1061. https://doi.org/10.1016/j.dss.2005.05.030.

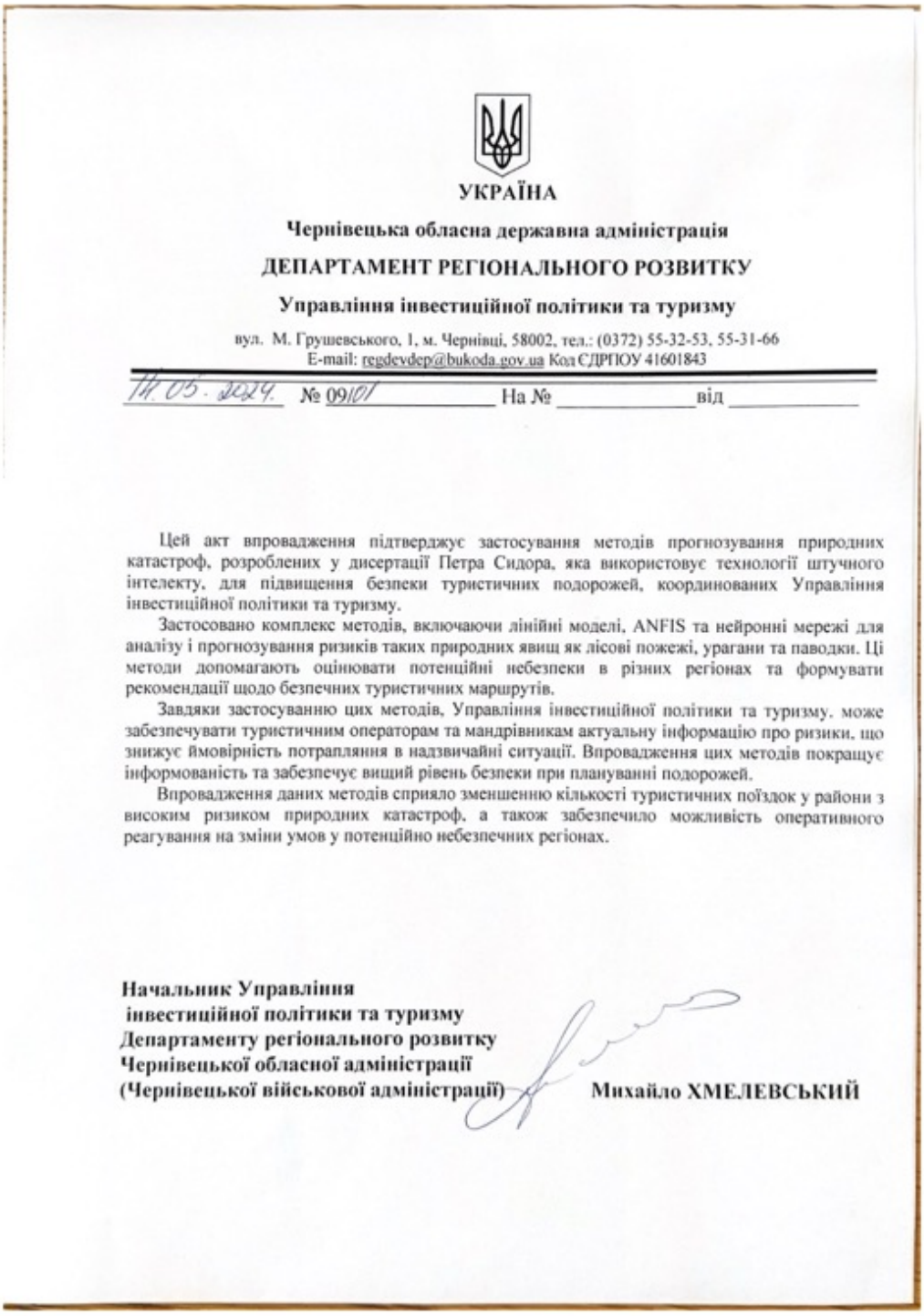
139. Huang, C. D., J. Goo, K. Nam, and C. W. Yoo. 2017. "Smart Tourism Technologies in Travel Planning: The Role of Exploration and Exploitation." *Information & Management*, Vol. 54, Issue 6, 757-770. https://doi.org/10.1016/j.im.2016.11.010.

140. D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-Franc¸ois Crespo, Dan Dennison Hidden Technical Debt in Machine Learning Systems *Proceedings of the 28th International Conference on Neural Information Processing Systems,* 2015. V2, pp. 2503–2511

ДОДАТКИ

ДОДАТОК А

Акти впровадження результатів дисертаційної роботи



УКРАЇНА

Чернівецька обласна державна адміністрація

ДЕПАРТАМЕНТ РЕГІОНАЛЬНОГО РОЗВИТКУ

Управління інвестиційної політики та туризму

вул. М. Грушевського, 1, м. Чернівці, 58002, тел.: (0372) 55-32-53, 55-31-66
E-mail: regdevdep@bukoda.gov.ua Код ЄДРПОУ 41601843

11.05.2024 № 09/01 На № _____ від _____

Цей акт впровадження підтверджує застосування методів прогнозування природних катастроф, розроблених у дисертації Петра Сидора, яка використовує технології штучного інтелекту, для підвищення безпеки туристичних подорожей, координованих Управління інвестиційної політики та туризму.

Застосовано комплекс методів, включаючи лінійні моделі, ANFIS та нейронні мережі для аналізу і прогнозування ризиків таких природних явищ як лісові пожежі, урагани та паводки. Ці методи допомагають оцінювати потенційні небезпеки в різних регіонах та формувати рекомендації щодо безпечних туристичних маршрутів.

Завдяки застосуванню цих методів, Управління інвестиційної політики та туризму, може забезпечувати туристичним операторам та мандрівникам актуальну інформацію про ризики, що знижує ймовірність потрапляння в надзвичайні ситуації. Впровадження цих методів покращує інформованість та забезпечує вищий рівень безпеки при плануванні подорожей.

Впровадження даних методів сприяло зменшенню кількості туристичних поїздок у райони з високим ризиком природних катастроф, а також забезпечило можливість оперативного реагування на зміни умов у потенційно небезпечних регіонах.

Начальник Управління
інвестиційної політики та туризму
Департаменту регіонального розвитку
Чернівецької обласної адміністрації
(Чернівецької військової адміністрації) **Михайло ХМЕЛЕВСЬКИЙ**

Громадська спілка «Регіональна туристична організація «Гостинна Буковина»

код ЄДРПОУ 42850755, адреса-58029, м.Чернівці, вул.Марка Черемшини, буд.9
e-mail – rtohb2019@gmail.com, тел./факс (0372)543616

Дата: 02 травня 2024 року м.Чернівці

Цей акт впровадження підтверджує застосування результатів досліджень дисертації Петра Сидора, присвяченої розробці методів прогнозування природних катастроф за допомогою технологій штучного інтелекту, для підвищення безпеки туристичних подорожей, організованих ГС «РТО «Гостинна Буковина».

Впроваджено інформаційну систему, розроблену на основі алгоритмів дисертації Сидора П.О., яка аналізує потенційні ризики природних катастроф (лісові пожежі, урагани, паводки) в регіонах пропонованих туристичних маршрутів. Система використовує прогнозні моделі для оповіщення менеджерів та клієнтів про можливі небезпеки.

Завдяки впровадженню, ГС «РТО «Гостинна Буковина» може запропонувати клієнтам більш безпечні маршрути, що знижує ризик потрапляння в надзвичайні ситуації під час подорожей, збільшуючи задоволеність клієнтів та їх довіру до агентства.

Впровадження системи дозволило скоротити кількість туристичних подорожей в райони з високим рівнем ризику природних катастроф, а також покращило інформаційну підтримку клієнтів перед подорожами.

З повагою Голова ГС «РТО«Гостинна Буковина» **Скрипник В.В.**

ГЕОГРАФСКИ ИНСТИТУТ „ЈОВАН ЦВИЈИЋ"
СРПСКЕ АКАДЕМИЈЕ НАУКА И УМЕТНОСТИ
Ђуре Јакшића 9, 11000 Београд, Србија
Телефон: 011 2636 276, Факс: 011 2637 597
Ел. пошта: general@gi.sanu.ac.rs

GEOGRAPHICAL INSTITUTE "JOVAN CVIJIC"
OF THE SERBIAN ACADEMY OF SCIENCES AND ARTS
Đure Jakšića 9, 11000 Belgrade, Serbia
Phone: 381 11 2636 276, Fax: 381 11 2637 597
E. mail: general@gi.sanu.ac.rs

Belgrade, April 29, 2024
No 55-1/24

This implementation act formalizes the application of natural disaster forecasting methods developed by Petro Sydor in his dissertation, which utilizes advanced artificial intelligence technologies. These methods are being implemented in the scientific research of the Institute of Geography to enhance the accuracy of natural disaster analysis and forecasting in Serbia.

Application of Methods:

1. **Linear models**, **ANFIS**, and **neural networks** are used for detailed analysis and forecasting of natural phenomena, including forest fires, hurricanes, and floods.
2. The use of these methods allows for accurately identifying potential risks and assessing their impact on various regions of Serbia, aiding in planning appropriate response and adaptation measures.

Practical Significance of Implementation: With the integration of these methods, the Institute of Geography can enhance the effectiveness of its scientific research in environmental safety and natural resource management. Implementing modern technologies in the Institute's research work will aid in developing new recommendations for governmental bodies and the private sector regarding the reduction of natural disaster risks.

Results of Implementation: The use of cutting-edge forecasting methods has improved the quality of scientific publications submitted by researchers at the Institute, increased the number of citations and international recognition, and enhanced interdisciplinary interaction within scientific projects related to geography and ecology.

Signatures of the Parties:
Director of the Geograpical Institute "Jovan Cvijić" SASA,
Dr Milan Radovanović

(signature) Radovanović (name) M. Radovanović

ДОДАТОК Б

Список опублікованих праць за темою дисертації та відомості про апробацію результатів дисертації

Наукові праці, в яких опубліковані основні наукові результати дисертації

[1] Malinović-Milićević S., Vyklyuk Y., Radovanović M. M., Milenković M., Pešić A.M., Milovanović B., Popović T., Sydor P., Petrović M. D., Applying machine learning in the investigation of the link between the high-velocity streams of charged solar particles and precipitation-induced floods. Environmental Monitoring and Assessment 2024. V196. 400. ISSN: 01676-369 (Scopus, **Q2).** *(Особистий внесок: розробка математичної моделі прогнозування паводків)*

[2] Шаховська Н., Сидор П., Розроблення архітектури системи планування безпечних туристичних подорожей Вісник Хмельницького національного університету. Технічні науки. 2022. №1. (305). С.96-101 *(Особистий внесок: розробка інформаційної технології)*

[3] Сидор П.О., Виклюк Я.І., Ансамблеві моделі прогнозування повеней у Великій Британії на основі сонячної активності. Вісник Хмельницького національного університету. Технічні науки. 2024. №2. (333). С. 218-231 *(Особистий внесок: розробка математичної моделі прогнозування паводків*

Наукові праці, які засвідчують апробацію матеріалів дисертації

[4] Vyklyuk Y., Radovanović M. M., Sydor P. Hurricane Forecasting Using by Parallel Calculations & Machine Learning 2018 IEEE 1st International Conference on System Analysis and Intelligent Computing, SAIC 2018 – Proceedings 31 October 2018 Kyiv, 2018, Article number 8516872 *(Особистий внесок: Створення математичної моделі прогнозування ураганів)*

[5] Виклюк Я.І., Сидор П.О., Кунанець Н. Е., Пасічник В.В. Прогнозування лісових пожеж на основі ANFIS та паралельних розрахунків. Інтелектуальні

системи прийняття рішень і проблеми обчислювального інтелекту: Матеріали міжнародної наукової конференції. Херсон: Видавництво ФОП Вишемирський В. С. с. Залізний Порт 21- 27 травня 2018 р. С.41-42 *(Особистий внесок: створення математичної моделі прогнозування лісових пожеж*

[6] Виклюк Я.І. Сидор П.О. Прогнозування лісових пожеж в Португалії. С. 25-32.

[7] Виклюк Я.І. Сидор П.О. Комп'ютерне моделювання та програмне забезпечення інформаційних систем і технологій (КМПЗ_2024) – : зб. наук. праць (тези доповідей та вибрані статті) IV Міжнародної науково-практичної конф. КМПЗ_2024. – (Чернівці, 30 травня – 01 червня 2024) / наук. ред. і відп. за вип. проф. В.М Заяць.- Львів: ЛНУ імені Івана Франка, 2024. – 342 с. *(Особистий внесок: створення математичної моделі прогнозування лісових пожеж)*

Наукові праці, які додатково відображають наукові результати дисертації

[8] Сидор П.О., Виклюк Я.І., Мобільна система інформаційної підтримки з рекомендаціями для безпечних подорожей. Науковий вісник НЛТУ України 2024. том 34. №3. С.103-109 *(Особистий внесок: алгоритм створення мобільного додатку)*

ДОДАТОК В

Лістинг програм на мові програмування Python, які розроблені в дисертаційному дослідженні

forest_data_load.py

```
import pandas as pd
from datetime import datetime
from pandas import concat
from datetime import timedelta
from dateutil.relativedelta import relativedelta

# tensorboard --logdir=logs


def parse(x):
    y=x.split()
    z=" ".join(y[:-1])
    t=int(y[-1])
    h=t//100
    m=t%100
    z=z+" "+str(h)+" "+str(m)
    return datetime.strptime(z, '%Y %m %d %H %M')

def parse2(x):
    x=x.split()
    y=x[0]+" "+x[2]
    #print("====",y)
    return datetime.strptime(y, '%Y-%m-%d %H:%M:%S')

def parse3(x):
    x=x.split()
    y=x[0]+" "+x[2]
    #print("====",y)
    t=datetime.strptime(y, '%Y-%m-%d %H:%M:%S')
    return t+timedelta(hours=7)+relativedelta(years=-1)

def parse4(x):
    x=x.split()
    y=x[0]+" "+x[2]
    #print("====",y)
```

```
    t=datetime.strptime(y, '%Y-%m-%d %H:%M:%S')

    return t+timedelta(hours=-3)+relativedelta(years=-1)

def parse5(x):
    x=x.split()
    y=x[0]+" "+x[2]
    #print("====",y)
    t=datetime.strptime(y, '%Y-%m-%d %H:%M:%S')

    return t+timedelta(hours=-1)+relativedelta(years=-1)


dt1=pd.read_excel('DATABASE_FOR_CALIFORNIA_GREECE_
PORTUGAL_FIRES_2018.xlsx', 'Sheet2', parse_dates = [['year', 'month', 'day',
'time']], index_col=0, date_parser=parse)
dt1.index.name = 'date'
dt2=pd.read_excel('DATABASE_FOR_CALIFORNIA_GREECE_
PORTUGAL_FIRES_2018.xlsx', 'Sheet3', parse_dates = [['year', 'month', 'day',
'time']], index_col=0, date_parser=parse)
dt2.index.name = 'date'
dt3=pd.read_excel('DATABASE_FOR_CALIFORNIA_GREECE_
PORTUGAL_FIRES_2018.xlsx', 'Sheet4', parse_dates = [['year', 'month', 'day',
'time']], index_col=0, date_parser=parse)
dt3.index.name = 'date'
dt4=pd.read_excel('DATABASE_FOR_CALIFORNIA_GREECE_
PORTUGAL_FIRES_2018.xlsx', 'Sheet5', parse_dates = [['DATE', 'TIME']],
index_col=0, date_parser=parse2)
dt4.index.name = 'date'
#California
dt_target1=pd.read_excel('CALIFORNIA_DATABASE.xlsx', 'Sheet4',
parse_dates = [['day/month', 'Time']], index_col=0, date_parser=parse3)
dt_target1.index.name = 'date'
dt_target2=pd.read_excel('CALIFORNIA_DATABASE.xlsx', 'Sheet5',
parse_dates = [['day/month', 'Time']], index_col=0, date_parser=parse3)
dt_target2.index.name = 'date'
dt_target3=pd.read_excel('CALIFORNIA_DATABASE.xlsx', 'Sheet6',
parse_dates = [['day/month', 'Time']], index_col=0, date_parser=parse3)
dt_target3.index.name = 'date'
#Greece
dt_target4=pd.read_excel('GREECE_DATABASE.xlsx', 'Sheet2', parse_dates =
[['day/month', 'Time']], index_col=0, date_parser=parse4)
dt_target4.index.name = 'date'
```

```
#Portugal
dt_target5=pd.read_excel('PORTUGAL_DATABASE.xlsx', 'Sheet2',
parse_dates = [['day/month', 'Time']], index_col=0, date_parser=parse5)
dt_target5.index.name = 'date'


#Join data:
dt1_j=dt1[[ dt1.columns[1], dt1.columns[3]]]
dt2_j=dt2[[dt2.columns[1], dt2.columns[2], dt2.columns[4], dt2.columns[5],
dt2.columns[6], dt2.columns[7], dt2.columns[8]]]
dt3_j=dt3[[dt3.columns[3], dt3.columns[4], dt3.columns[5]]]
dt4_j=dt4
dt_t1_j=dt_target1[[dt_target1.columns[0], dt_target1.columns[2],
dt_target1.columns[6]]]
dt_t2_j=dt_target2[[dt_target2.columns[0], dt_target2.columns[2],
dt_target2.columns[6]]]
dt_t3_j=dt_target3[[dt_target3.columns[0], dt_target3.columns[2],
dt_target3.columns[6]]]
dt_t4_j=dt_target4[[dt_target4.columns[0], dt_target4.columns[2],
dt_target4.columns[6]]]
dt_t5_j=dt_target5[[dt_target5.columns[0], dt_target5.columns[2],
dt_target5.columns[6]]]

DS=dt1_j.join(dt2_j, how='outer')
DS=DS.join(dt3_j, how='outer' )
DS=DS.join(dt4_j, how='outer' )
DS1=DS.join(dt_t1_j, how='outer' )
DS2=DS.join(dt_t2_j, how='outer' )
DS3=DS.join(dt_t3_j, how='outer' )
DS4=DS.join(dt_t4_j, how='outer' )
DS5=DS.join(dt_t5_j, how='outer' )

writer = pd.ExcelWriter('DataSet.xlsx')
DS1.to_excel(writer, 'CALIFORNIA1')
DS2.to_excel(writer, 'CALIFORNIA2')
DS3.to_excel(writer, 'CALIFORNIA3')
DS4.to_excel(writer, 'GREECE')
DS5.to_excel(writer, 'PORTUGAL')
writer.save()
```

forest_lstm.py

```
import pandas as pd
import datetime
from datetime import datetime
import matplotlib.pyplot as plt
import seaborn as sns
from pandas import concat
from pandas import DataFrame, Series
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from keras.callbacks import TensorBoard
from datetime import timedelta
from dateutil.relativedelta import relativedelta

# tensorboard --logdir=logs

T_DISCRETTE = 30
date_before = datetime.strptime('2018, 8, 12, 0, 0', '%Y, %m, %d, %H, %M')  # California
sheet = 'PORTUGAL'

T_LAG = 60 * 24 * 4
N_LAG = int(T_LAG / T_DISCRETTE)
TRAIN_TEST = 0.7
n_neurons = 50
epochs = 100
batch_size = 1


# перетворення часових рядів

def series_to_supervised(in_data, tar_data, n_in=1, t_d=1, dropnan=True):
    n_vars = in_data.shape[1]
    cols, names = list(), list()
    # ввід послідовності (t-n, ... t-1)
    for i in range(1, n_in + 1):
        cols.append(in_data.shift(i))
        names += [('%s(t-%d)' % (in_data.columns[j], i * t_d)) for j in
range(n_vars)]
```

```
    # Звдення
    cols.append(tar_data)
    names += list(tar_data.columns)
    agg = concat(cols, axis=1)
    agg.columns = names

    # Фільтрація NaN значень
    if dropnan:
        agg.dropna(inplace=True)

    return agg


DS = pd.read_excel('DataSet.xlsx', sheet, index_col=0)
DS.index = pd.to_datetime(DS.index)

print(DS.head())
DS = DS[DS.index < date_before]  # Обрізка даних дати

# Усереднення данних
DS_res = DS.resample(str(T_DISCRETTE) +
'T').max().interpolate(method='pchip')
# ковзне вікно
# DS_res=DS_res.rolling(4).mean()


DF_C = DS_res.corr()

sns.heatmap(DF_C);

writer = pd.ExcelWriter('Correlation_' + sheet + '.xlsx')
DF_C.to_excel(writer, sheet)
writer.save()

f, ax = plt.subplots(len(DS_res.columns))
for i, c in enumerate(DS_res.columns):
    ax[i].plot(DS_res[c])

plt.figure()
# plt.plot(dt[dt.columns[1]])
d = DS[DS.columns[1]].copy()
d.dropna(inplace=True, how='all')
plt.plot(d)
```

```
plt.plot(DS_res[DS_res.columns[1]])
# plt.plot(DS_res[DS_res.columns[1]])


f, (ax1, ax2) = plt.subplots(2)
d1 = DS[DS.columns[2]].copy()
d2 = DS[DS.columns[3]].copy()
d1.dropna(inplace=True, how='all')
d2.dropna(inplace=True, how='all')

ax1.plot(d1)
ax2.plot(d2)
ax1.plot(DS_res[DS_res.columns[2]])
ax2.plot(DS_res[DS_res.columns[3]])

f, (ax1, ax2, ax3, ax4, ax5) = plt.subplots(5)
d1 = DS[DS.columns[4]].copy()
d2 = DS[DS.columns[5]].copy()
d3 = DS[DS.columns[6]].copy()
d4 = DS[DS.columns[7]].copy()
d5 = DS[DS.columns[8]].copy()
d1.dropna(inplace=True, how='all')
d2.dropna(inplace=True, how='all')
d3.dropna(inplace=True, how='all')
d4.dropna(inplace=True, how='all')
d5.dropna(inplace=True, how='all')

ax1.plot(d1)
ax2.plot(d2)
ax3.plot(d3)
ax4.plot(d4)
ax5.plot(d5)
ax1.plot(DS_res[DS_res.columns[4]])
ax2.plot(DS_res[DS_res.columns[5]])
ax3.plot(DS_res[DS_res.columns[6]])
ax4.plot(DS_res[DS_res.columns[7]])
ax5.plot(DS_res[DS_res.columns[8]])

f, (ax1, ax2, ax3) = plt.subplots(3)
d1 = DS[DS.columns[9]].copy()
d2 = DS[DS.columns[10]].copy()
d3 = DS[DS.columns[11]].copy()
```

```
d1.dropna(inplace=True, how='all')
d2.dropna(inplace=True, how='all')
d3.dropna(inplace=True, how='all')

ax1.plot(d1)
ax2.plot(d2)
ax3.plot(d3)

ax1.plot(DS_res[DS_res.columns[9]])
ax2.plot(DS_res[DS_res.columns[10]])
ax3.plot(DS_res[DS_res.columns[11]])

plt.figure()
d1 = DS[DS.columns[12]].copy()
d1.dropna(inplace=True, how='all')
plt.plot(d1)
plt.plot(DS_res[DS_res.columns[12]])

f, (ax1, ax2, ax3) = plt.subplots(3)
d1 = DS[DS.columns[13]].copy()
d2 = DS[DS.columns[14]].copy()
d3 = DS[DS.columns[15]].copy()

d1.dropna(inplace=True, how='all')
d2.dropna(inplace=True, how='all')
d3.dropna(inplace=True, how='all')

ax1.plot(d1)
ax2.plot(d2)
ax3.plot(d3)

ax1.plot(DS_res[DS_res.columns[13]])
ax2.plot(DS_res[DS_res.columns[14]])
ax3.plot(DS_res[DS_res.columns[15]])

plt.show()

# зображення нормалізованих графіків
scaler_DS = MinMaxScaler(feature_range=(0, 1))
d = DS_res.copy()
d.dropna(inplace=True)
d = scaler_DS.fit_transform(d)
plt.figure()
```

```
plt.plot(d[:, 0:2])
plt.figure()
plt.plot(d[:, 2:4])
plt.figure()
plt.plot(d[:, 4:9])
plt.figure()
plt.plot(d[:, 9:12])
plt.figure()
plt.plot(d[:, 12:13])
plt.figure()
plt.plot(d[:, 13:16])

plt.show()
print(DS_res.columns[1:])

in_d, tr_d = DS_res[DS_res.columns[1:]], DS_res[DS_res.columns[-3:]]
DS_LAG = series_to_supervised(in_d, tr_d, N_LAG, T_DISCRETTE)

data_x, data_y = DS_LAG[DS_LAG.columns[:-3]],
DS_LAG[DS_LAG.columns[-3:]]
scaler_x = MinMaxScaler(feature_range=(0, 1))
scaler_y = MinMaxScaler(feature_range=(0, 1))

scaled_x = scaler_x.fit_transform(data_x)
scaled_y = scaler_y.fit_transform(data_y)

n_obs = int(TRAIN_TEST * data_x.shape[0])
train_x, train_y = scaled_x[:n_obs, :], scaled_y[:n_obs, :]
test_x, test_y = scaled_x[n_obs:, :], scaled_y[n_obs:, :]

# n_obs=int((1-TRAIN_TEST)*data_x.shape[0])
# test_x, test_y = scaled_x[:n_obs,:], scaled_y[:n_obs,:]
# train_x, train_y = scaled_x[n_obs:,:], scaled_y[n_obs:,:]

train_x_LSTM = train_x.reshape((train_x.shape[0], N_LAG, in_d.shape[1]))
test_x_LSTM = test_x.reshape((test_x.shape[0], N_LAG, in_d.shape[1]))

batch_size = int(train_x.shape[0] * 0.1)
# batch_size=1 найкращий результат

model = Sequential()
model.add(LSTM(n_neurons, input_shape=(train_x_LSTM.shape[1],
train_x_LSTM.shape[2])))
```

```
model.add(Dense(train_y.shape[1]))  # activation='sigmoid'
model.compile(loss='mean_squared_error', optimizer='adam')
# model.compile(loss='mae', optimizer='adam')
tensorboard = TensorBoard(log_dir="logs", histogram_freq=0,
write_graph=True, write_images=False)
history = model.fit(train_x_LSTM, train_y, epochs=epochs,
batch_size=batch_size, validation_data=(test_x_LSTM, test_y),
                verbose=2, shuffle=False, callbacks=[tensorboard])
# plot history
plt.plot(history.history['loss'], label='train')
plt.plot(history.history['val_loss'], label='test')
plt.ylabel('loss')
plt.xlabel('epoch')
# plt.plot(history.history['acc'], label='acc')
# plt.plot(history.history['val_acc'], label='acc test')


plt.legend()
plt.show()

forecast_train = model.predict(train_x_LSTM)
forecast_test = model.predict(test_x_LSTM)
forecast_train = scaler_y.inverse_transform(forecast_train)
forecast_test = scaler_y.inverse_transform(forecast_test)

'''
forecast=DataFrame(concatenate((forecast_train,forecast_test), axis=0))
forecast.index=data_y.index
forecast.columns=['Forecast '+i for i in data_y.columns]
res=data_y.join(forecast, how='outer')
'''

forecast = DataFrame(forecast_train)
# forecast=DataFrame(forecast_test)

forecast.index = data_y.index[:n_obs]
forecast.columns = ['Train ' + i for i in data_y.columns]
# forecast.columns=['Test '+i for i in data_y.columns]

res = data_y.join(forecast, how='outer')

forecast = DataFrame(forecast_test)
# forecast=DataFrame(forecast_train)
```

```
forecast.index = data_y.index[n_obs:]
forecast.columns = ['Test ' + i for i in data_y.columns]
# forecast.columns=['Train '+i for i in data_y.columns]

res = res.join(forecast, how='outer')

f, (ax1, ax2, ax3) = plt.subplots(3)
ax1.plot(res[res.columns[0]])
ax2.plot(res[res.columns[1]])
ax3.plot(res[res.columns[2]])
ax1.plot(res[res.columns[3]])
ax2.plot(res[res.columns[4]])
ax3.plot(round(res[res.columns[5]] * 10) / 10)
ax1.plot(res[res.columns[6]])
ax2.plot(res[res.columns[7]])
ax3.plot(round(res[res.columns[8]] * 10) / 10)

plt.show()

'''
writer = pd.ExcelWriter('DataSet.xlsx')
DS.to_excel(writer,'Sheet1')
#DS_res.to_excel(writer,'Sheet2')
#DS_LAG.to_excel(writer,'Sheet3')
res.to_excel(writer,'Sheet4')
writer.save()


# приведення моделі в JSON і збереження/серіалізація моделі
model_json = model.to_json()
with open("model.json", "w") as json_file:
    json_file.write(model_json)
# приведення моделі в to HDF5 і збереження/серіалізація моделі
model.save_weights("model.h5")
print("Saved model to disk")


'''
# Операційна секція.
'''
from keras.models import model_from_json
# Завантаження моделі
json_file = open('model.json', 'r')
```

```
loaded_model_json = json_file.read()
json_file.close()
loaded_model = model_from_json(loaded_model_json)
# Додавання ваг до моделі
loaded_model.load_weights("model.h5")
print("Loaded model from disk")

# Перевірка моделі і тестування даних
loaded_model.compile(loss='binary_crossentropy', optimizer='rmsprop',
metrics=['accuracy'])
score = loaded_model.evaluate(X, Y, verbose=0)
print("%s: %.2f%%" % (loaded_model.metrics_names[1], score[1]*100))
'''
```

modeling_analyses_fire_lstm.py

```
import pandas as pd
import numpy as np
from sklearn import preprocessing
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from keras.layers import Dropout
from keras.callbacks import EarlyStopping
from sklearn.metrics import mean_squared_error
from sklearn import linear_model
from sklearn.model_selection import cross_val_predict, cross_validate
from sklearn.model_selection import KFold
from keras.layers.convolutional import Conv1D, Conv2D
from keras.layers.convolutional import MaxPooling1D, MaxPooling2D
from keras.layers import Flatten
from keras import regularizers


def interpolate(ds, method='time'):
    """
    Інтерполяція DataFrame з відріззанням хвостів зпереду і ззаду
    :param ds: DataFrame
    :param method: Метод
    :return:
    """
```

```
    df = pd.DataFrame()
    for c in ds.columns:
        ts = ds[c]
        date_begin = ts[~np.isnan(ts)].index[0]
        date_end = ts[~np.isnan(ts)].index[-1]
        # print(c, date_begin, date_end)
        ts = ts[np.logical_and(ts.index >= date_begin, ts.index <=
date_end)].interpolate(method=method)
        df = df.join(ts, how='outer')
    return df


def my_plt(dt, plt, col, n, l='', title=''):
    """
    Аналіз графіків
    :param dt: DataFrame
    :param plt: Фігура для виводу
    :param col: перелік індексів полів для виводу графіків
    :param n: Чи знищувати порожні рядки
    :param l: Розташування легенди
    :param title: Назва графіка
    :return:
    """
    font = {'size': 10}
    plt.rc('font', **font)
    plt.subplots_adjust(bottom=0.05)
    x = dt.index
    leg = []
    plt.title(title)
    plt.xlabel('Date')
    for i in col:
        y = dt[dt.columns[i]].values
        if n:
            yn = preprocessing.normalize(y[~np.isnan(y)].reshape(1, -1))
            y[~np.isnan(y)] = yn[0]
        plt.plot(x, y)
        leg = leg + ["%s" % dt.columns[i]]
        # leg=leg+["%s" % dt.columns[i]]
    plt.legend(leg, loc=l)
    plt.xlim(xmin=x[0], xmax=x[-1])
    # plt.xticks(np.arange(0, len(X), 8), [X[i][:2]+'.'+X[i][3:-4] for i in
range(0,len(X), 8)], rotation=0)
```

```
def lag_correlation_ts(y, x, lag):
    """
    Лагова кореляція для 2 DateSeries
    :param y: fixed
    :param x: shifted
    :param lag:
    :return:
    """
    r = [0] * (lag + 1)
    y = y.copy()
    x = x.copy()
    y.name = "y"
    x.name = "x"

    for i in range(0, lag + 1):
        ds = y.copy().to_frame()
        ds = ds.join(x.shift(i), how='outer')
        r[i] = ds.corr().values[0][1]
    return r


def lag_correlation(df_x, df_y, lag=10, file=None):
    """
    Лагова перевірка
    :param df_x: DataFrame вхідних полів
    :param df_y: DataFrame вихідних полів
    :param lag: перевірочнй лаг
    :param file: файл виводу результатів
    :return:
    """
    if file is not None:
        print(file)
        writer = pd.ExcelWriter(file)
        for s in df_y.columns:
            l = pd.DataFrame()
            for x in df_x.columns:
                c = lag_correlation_ts(df_y[s], df_x[x], lag)
                df_c = pd.DataFrame(c)
                df_c.columns = [x]
                l = l.join(df_c, how='outer')
            l.to_excel(writer, s)
```

```
        writer.save()
    else:
        print("No file")
    return None


def series_to_supervised(in_data, tar_data, n_in=1, dropnan=True,
target_dep=False):
    """
    Перетворення до навчальної вибірки з врахуванням лагу
    :param in_data: Вхідні поля
    :param tar_data: Вихідне поле (одне)
    :param n_in: Лаговий зсув
    :param dropnan: Чи знищувати порожні рядки
    :param target_dep: Чи враховувати лаг вхідного поля В разі врахування
вхідні почнуться з лагу 1
    :return: Навчальну вибірку. Останнє поле – вихідне
    """

    n_vars = in_data.shape[1]
    cols, names = list(), list()
    # input sequence (t-n, ... t-1)
    # for i in range(n_in, -1, -1):
    if target_dep:
        i_start = 1
    else:
        i_start = 0
    for i in range(i_start, n_in + 1):
        cols.append(in_data.shift(i))
        names += [('%s(t-%d)' % (in_data.columns[j], i)) for j in range(n_vars)]

    if target_dep:
        for i in range(n_in, -1, -1):
            cols.append(tar_data.shift(i))
            names += [('%s(t-%d)' % (tar_data.name, i))]
    else:
        # put it all together
        cols.append(tar_data)
        # print(tar_data.name)
        names.append(tar_data.name)
    agg = pd.concat(cols, axis=1)
    agg.columns = names
```

```
    # drop rows with NaN values
    if dropnan:
        agg.dropna(inplace=True)

    return agg

def CNN2d_model(train_x, filter=64):
    """
    Згорткова мережа 2d
    :param train_x: навчальна вибірка
    :param neurons: Кількість нейронів
    :return: модель
    """
    # activity_regularizer = regularizers.l2(0.001)
    activity_regularizer = None

    model = Sequential()
    model.add(Conv2D(filter, kernel_size=(5, 1),
activity_regularizer=activity_regularizer, activation='relu',
input_shape=train_x.shape[1:]))
    model.add(MaxPooling2D(pool_size=(2, 1)))
    model.add(Conv2D(filter, kernel_size=(5, 1), activation='relu'))
    model.add(MaxPooling2D(pool_size=(2, 1)))
    model.add(Dropout(rate=0.25))

    model.add(Flatten())
    model.add(Dense(int(filter/2), activation='relu'))
    model.add(Dropout(0.2))
    model.add(Dense(1))
    model.compile(optimizer='adam', loss='mse')
    return model


def CNN_model(train_x, filter=64):
    """

    :param train_x: навчальна вибірка
    :param neurons: Кількість нейронів
    :return: модель
    """

    n_features=1
    model = Sequential()
```

```
    # activity_regularizer=regularizers.l2(0.001)
    activity_regularizer = None
    model.add(Conv1D(filters=filter, kernel_size=5,
activity_regularizer=activity_regularizer, activation='relu',
input_shape=(train_x.shape[1], n_features)))
    model.add(Conv1D(filters=filter, kernel_size=5, activation='relu' ))
    model.add(MaxPooling1D(pool_size=2))
    model.add(Dropout(0.2))
    # model.add(Conv1D(filters=filter*2, kernel_size=1, activation='relu' ))
    # model.add(Conv1D(filters=filter*2, kernel_size=1, activation='relu' ))
    # model.add(MaxPooling1D(pool_size=1))
    # model.add(Dropout(0.2))
    model.add(Flatten())
    model.add(Dense(50, activation='relu'))
    model.add(Dense(1))
    model.compile(optimizer='adam', loss='mse')
    return model

def LSTM_model(train_x_lstm, train_y_lstm, neurons=10):
    """
    Побудова LSTM мережі
    :param train_x_lstm:
    :param train_y_lstm:
    :param neurons:
    :return: модель
    """
    multy_layer = True
    # activity_regularizer=regularizers.l2(0.001)
    activity_regularizer = None
    ''' return_sequences=False,, activation='relu', '''
    model = Sequential()
    model.add(LSTM(neurons, return_sequences=multy_layer,
activity_regularizer=activity_regularizer, input_shape=(train_x_lstm.shape[1],
train_x_lstm.shape[2])))
    model.add(Dropout(0.2))
    if multy_layer:
        model.add(LSTM(neurons))
        model.add(Dropout(0.2))
        # model.add(Dense(neurons, kernel_initializer='normal', activation='relu'))
        # model.add(Dropout(0.2))
    model.add(Dense(train_y_lstm.shape[1]))
    # activation='sigmoid'
    model.compile(loss='mse', optimizer='adam')
```

```
    return model


def LSTM_crossvalidation(DF_X, DF_Y, n_splits=10, lag_in=1, neurons=10,
epochs=400, patience=0, target_dep=False,
                         only_output=False, shuffle=False, verbose=2):
    """

    :param DF_X:
    :param DF_Y:
    :param n_splits:
    :param lag_in:
    :param neurons:
    :param epochs:
    :param patience:
    :param target_dep:
    :param only_output:
    :param shuffle:
    :param verbose:
    :return:
    """

    DF_SV = series_to_supervised(DF_X, DF_Y, lag_in, target_dep=target_dep)
    if only_output and target_dep:
        y_LSTM, x_LSTM = DF_SV[DF_SV.columns[-1:]],
DF_SV[DF_SV.columns[-(lag_in + 1):-1]]
    else:
        y_LSTM, x_LSTM = DF_SV[DF_SV.columns[-1:]],
DF_SV[DF_SV.columns[:-1]]
    # print(x_LSTM.columns)

    scaler_x_LSTM = MinMaxScaler(feature_range=(0, 1))
    scaler_y_LSTM = MinMaxScaler(feature_range=(0, 1))

    scaled_x_LSTM = scaler_x_LSTM.fit_transform(x_LSTM)
    scaled_y_LSTM = scaler_y_LSTM.fit_transform(y_LSTM)

    train_y = scaled_y_LSTM
    if target_dep:
        resh_x_train = scaled_x_LSTM.shape[0]
        resh_y_train = lag_in
        resh_z_train = 1 + scaled_x_LSTM.shape[1] // (lag_in + 1)
    else:
```

```
        resh_x_train = scaled_x_LSTM.shape[0]
        resh_y_train = lag_in + 1
        resh_z_train = scaled_x_LSTM.shape[1] // (lag_in + 1)



    train_x_LSTM = scaled_x_LSTM.reshape((resh_x_train, resh_y_train,
resh_z_train))

    cw_r_test = np.zeros(train_y.shape)
    cw_r_train = np.zeros(train_y.shape)
    batch_size = int(train_y.shape[0] * .1)
    call = []
    if patience > 0:
        reduce_lr = EarlyStopping(monitor='val_loss', patience=10, verbose=0,
mode='auto', restore_best_weights=True)
        call.append(reduce_lr)

    h = []
    wrong_model = 0
    i = 0
    # crossvalidation
    kf = KFold(n_splits=n_splits)
    kf.shuffle=shuffle
    for train_index, test_index in kf.split(train_y):
        i += 1
        # print("TRAIN:", train_index, "TEST:", test_index)
        # print(train_x_LSTM.shape,train_y.shape)
        tr_X = train_x_LSTM[train_index, :, :]
        tr_Y = train_y[train_index]
        ts_X = train_x_LSTM[test_index, :, :]
        ts_Y = train_y[test_index]
        # print(tr_X.shape,tr_Y.shape,ts_X.shape,ts_Y.shape)

        model = LSTM_model(tr_X, tr_Y, neurons)
        history = model.fit(tr_X, tr_Y, epochs=epochs, batch_size=batch_size,
validation_data=(ts_X, ts_Y), verbose=0,
                            shuffle=False, callbacks=call)
        h.append(history.epoch[-1])
        yhat_test = model.predict(ts_X)
        yhat_train = model.predict(tr_X)
        mse_test = np.mean((yhat_test – ts_Y) ** 2)
        mse_train = np.mean((yhat_train – tr_Y) ** 2)
```

```
        if mse_test < mse_train:
            wrong_model += 1

        if verbose>0:
            if mse_test < mse_train:
                print("#", i, "Wrong LSTM model for output ", DF_Y.name)
                print("Train loss", mse_train)
                print("Test loss", mse_test)
            else:
                print("#", i, "OK LSTM model for output ", DF_Y.name)

        del model
        cw_r_test[test_index] = yhat_test
        cw_r_train[train_index] = yhat_train
    if wrong_model > 0:
        print("Wrong LSTM model for output ", DF_Y.name, wrong_model * 100
// n_splits, "% models wrong")
    else:
        print("OK for all LSTM model")

    forecast_test = scaler_y_LSTM.inverse_transform(cw_r_test)
    forecast_train = scaler_y_LSTM.inverse_transform(cw_r_train)

    res_test = pd.DataFrame(forecast_test)
    res_test.index = y_LSTM.index
    res_test.columns = ['CV_Test_LSTM']

    res_train = pd.DataFrame(forecast_train)
    res_train.index = y_LSTM.index
    res_train.columns = ['CV_Train_LSTM']

    mse_test = np.mean((res_test.values – y_LSTM.values) ** 2)
    mse_train = np.mean((res_train.values – y_LSTM.values) ** 2)
    return res_train, res_test, mse_train, mse_test, h


    #     h.append(history.epoch[-1])
    #     yhat = model.predict(ts_X)
    #     yhat = scaler_y_LSTM.inverse_transform(yhat)
    #     del model
    #     cw_r[test_index] = yhat
    # if wrong_model > 0:
```

```
    #     print("Wrong LSTM model for output ", DF_Y.name, wrong_model *
100 / n_splits, "% models wrong")
    #
    # forecast = pd.DataFrame(cw_r)
    # forecast.index = y_LSTM.index
    # forecast.columns = ['CV_Test_LSTM']
    # return forecast, h



def Linear_crossvalidation(DF_X, DF_Y, lag_in=1, n_splits=10,
target_dep=False, only_output=False, shuffle=False, verbose=2):
    """

    :param DF_X: DataFrame вхідних полів
    :param DF_Y: DataFrame вихідного поля
    :param lag_in: лаг вхідних полів
    :param n_splits: Кількіть розділень
    :param target_dep: врахування вихідного поля
    :param only_output: тільки вихідне поле як вхідні параметри
    :param shuffle: Чи перемішувати кросвалідацію
    :return: прогноз навчальної вибірки, прогноз тренувальної вибірки, MSE
навчальної вибірки, MSE тренувальної вибірки
    """


    DF_SV = series_to_supervised(DF_X, DF_Y, lag_in, target_dep=target_dep)
    if only_output and target_dep:
        y, x = DF_SV[DF_SV.columns[-1:]], DF_SV[DF_SV.columns[-(lag_in +
1):-1]]
    else:
        y, x = DF_SV[DF_SV.columns[-1:]], DF_SV[DF_SV.columns[:-1]]
    scaler_x = MinMaxScaler(feature_range=(0, 1))
    scaler_y = MinMaxScaler(feature_range=(0, 1))

    scaled_x = scaler_x.fit_transform(x)
    scaled_y = scaler_y.fit_transform(y)

    kf = KFold(n_splits=n_splits)
    kf.shuffle=shuffle
    i = 0

    cw_r_test = np.zeros(scaled_y.shape)
```

```
    cw_r_train = np.zeros(scaled_y.shape)
    wrong_model = 0
    for train_index, test_index in kf.split(scaled_y):
        i += 1
        tr_X = scaled_x[train_index, :]
        tr_Y = scaled_y[train_index]
        ts_X = scaled_x[test_index, :]
        ts_Y = scaled_y[test_index]
        # print(tr_X.shape,tr_Y.shape,ts_X.shape,ts_Y.shape)
        # exit()
        model = linear_model.LinearRegression()
        model.fit(tr_X, tr_Y)
        yhat_test = model.predict(ts_X)
        yhat_train = model.predict(tr_X)
        mse_test=np.mean((yhat_test – ts_Y) ** 2)
        mse_train=np.mean((yhat_train – tr_Y) ** 2)
        del model

        if mse_test < mse_train:
            wrong_model += 1

        if verbose>0:
            if mse_test < mse_train:
                print("#", i, "Wrong Linear model for output ", DF_Y.name)
                print("Train loss", mse_train)
                print("Test loss", mse_test)
            else:
                print("#", i, "OK Linear model for output ", DF_Y.name)

        cw_r_test[test_index] = yhat_test
        cw_r_train[train_index] = yhat_train
    if wrong_model > 0:
        print("Wrong Linear model for output ", DF_Y.name, wrong_model * 100
// n_splits, "% models wrong")
    else:
        print("OK for all Linear model")

    forecast_test = scaler_y.inverse_transform(cw_r_test)
    forecast_train = scaler_y.inverse_transform(cw_r_train)

    res_test = pd.DataFrame(forecast_test)
    res_test.index = y.index
    res_test.columns = ['CV_Test_Linear']
```

```
    res_train = pd.DataFrame(forecast_train)
    res_train.index = y.index
    res_train.columns = ['CV_Train_Linear']

    mse_test = np.mean((res_test.values – y.values) ** 2)
    mse_train = np.mean((res_train.values – y.values) ** 2)
    return res_train, res_test, mse_train, mse_test




def CNN1D_crossvalidation(DF_X, DF_Y, n_splits=10, lag_in=2, filter=64,
epochs=400, patience=0, target_dep=False,
                          only_output=False, shuffle=False, verbose=2):
    """

    :param DF_X: DataSet входів
    :param DF_Y: DataSet входів
    :param n_splits: кількість розбиттів
    :param lag_in: Величина лагу
    :param neurons: кількість нейронів
    :param epochs: максимальна кількість епох
    :param patience: коли зупинити навчання
    :param target_dep: чи врховувати історію вихідного поля
    :param only_output: тільки вихідна поле на вхід
    :param shuffle: чи мішати кросвалідаційну вибірку
    :return:
    """
    DF_SV = series_to_supervised(DF_X, DF_Y, lag_in, target_dep=target_dep)
    if only_output and target_dep:
        y_CNN, x_CNN = DF_SV[DF_SV.columns[-1:]],
DF_SV[DF_SV.columns[-(lag_in + 1):-1]]
    else:
        y_CNN, x_CNN = DF_SV[DF_SV.columns[-1:]],
DF_SV[DF_SV.columns[:-1]]
    # print(x_CNN.columns)

    scaler_x_CNN = MinMaxScaler(feature_range=(0, 1))
    scaler_y_CNN = MinMaxScaler(feature_range=(0, 1))

    scaled_x_CNN = scaler_x_CNN.fit_transform(x_CNN)
    scaled_y_CNN = scaler_y_CNN.fit_transform(y_CNN)
```

```
    train_y = scaled_y_CNN
    train_x_CNN = scaled_x_CNN
    train_x_CNN = train_x_CNN.reshape((train_x_CNN.shape[0],
train_x_CNN.shape[1], 1))

    cw_r_test = np.zeros(train_y.shape)
    cw_r_train = np.zeros(train_y.shape)
    batch_size = int(train_y.shape[0] * .1)
    call = []
    if patience > 0:
        reduce_lr = EarlyStopping(monitor='val_loss', patience=10, verbose=0,
mode='auto', restore_best_weights=True)
        call.append(reduce_lr)

    h = []
    wrong_model = 0
    i = 0
    # crossvalidation
    kf = KFold(n_splits=n_splits)
    kf.shuffle=shuffle
    for train_index, test_index in kf.split(train_y):
        i += 1
        tr_X = train_x_CNN[train_index, :]
        tr_Y = train_y[train_index]
        ts_X = train_x_CNN[test_index, :]
        ts_Y = train_y[test_index]
        # print(tr_X.shape,tr_Y.shape,ts_X.shape,ts_Y.shape)
        # exit()
        model = CNN_model(tr_X, filter)
        history = model.fit(tr_X, tr_Y, epochs=epochs, batch_size=batch_size,
validation_data=(ts_X, ts_Y), verbose=0,
                            shuffle=False, callbacks=call)
        h.append(history.epoch[-1])
        yhat_test = model.predict(ts_X)
        yhat_train = model.predict(tr_X)
        mse_test=np.mean((yhat_test – ts_Y) ** 2)
        mse_train=np.mean((yhat_train – tr_Y) ** 2)
        if mse_test < mse_train:
            wrong_model += 1

        if verbose>0:
            if mse_test < mse_train:
                print("#", i, "Wrong CNN1D model for output ", DF_Y.name)
```

```
            print("Train loss", mse_train)
            print("Test loss", mse_test)
        else:
            print("#", i, "OK CNN1D model for output ", DF_Y.name)

        del model
        cw_r_test[test_index] = yhat_test
        cw_r_train[train_index] = yhat_train
    if wrong_model > 0:
        print("Wrong CNN1D model for output ", DF_Y.name, wrong_model *
100 // n_splits, "% models wrong")
    else:
        print("OK for all CNN1D model")

    forecast_test = scaler_y_CNN.inverse_transform(cw_r_test)
    forecast_train = scaler_y_CNN.inverse_transform(cw_r_train)

    res_test = pd.DataFrame(forecast_test)
    res_test.index = y_CNN.index
    res_test.columns = ['CV_Test_CNN1D']

    res_train = pd.DataFrame(forecast_train)
    res_train.index = y_CNN.index
    res_train.columns = ['CV_Train_CNN1D']

    mse_test = np.mean((res_test.values – y_CNN.values) ** 2)
    mse_train = np.mean((res_train.values – y_CNN.values) ** 2)
    return res_train, res_test, mse_train, mse_test, h


def CNN2D_crossvalidation(DF_X, DF_Y, n_splits=10, lag_in=1, filter=10,
epochs=400, patience=0, target_dep=False,
                          only_output=False, shuffle=False, verbose=2):
    """

    :param DF_X:
    :param DF_Y:
    :param n_splits:
    :param lag_in:
    :param filter:
    :param epochs:
    :param patience:
    :param target_dep:
```

```
    :param only_output:
    :param shuffle:
    :param verbose:
    :return:
    """

    DF_SV = series_to_supervised(DF_X, DF_Y, lag_in, target_dep=target_dep)
    if only_output and target_dep:
        y_CNN, x_CNN = DF_SV[DF_SV.columns[-1:]],
DF_SV[DF_SV.columns[-(lag_in + 1):-1]]
    else:
        y_CNN, x_CNN = DF_SV[DF_SV.columns[-1:]],
DF_SV[DF_SV.columns[:-1]]

    scaler_x_CNN = MinMaxScaler(feature_range=(0, 1))
    scaler_y_CNN = MinMaxScaler(feature_range=(0, 1))

    scaled_x_CNN = scaler_x_CNN.fit_transform(x_CNN)
    scaled_y_CNN = scaler_y_CNN.fit_transform(y_CNN)

    train_y = scaled_y_CNN
    if target_dep:
        resh_x_train = scaled_x_CNN.shape[0]
        resh_y_train = lag_in
        resh_z_train = 1 + scaled_x_CNN.shape[1] // (lag_in + 1)
    else:
        resh_x_train = scaled_x_CNN.shape[0]
        resh_y_train = lag_in + 1
        resh_z_train = scaled_x_CNN.shape[1] // (lag_in + 1)


    train_x_CNN = scaled_x_CNN.reshape((resh_x_train, resh_y_train,
resh_z_train, 1))



    cw_r_test = np.zeros(train_y.shape)
    cw_r_train = np.zeros(train_y.shape)
    batch_size = int(train_y.shape[0] * .1)
    call = []
    if patience > 0:
        reduce_lr = EarlyStopping(monitor='val_loss', patience=10, verbose=0,
mode='auto', restore_best_weights=True)
        call.append(reduce_lr)
```

```
    h = []
    wrong_model = 0
    i = 0
    # crossvalidation
    kf = KFold(n_splits=n_splits)
    kf.shuffle=shuffle
    for train_index, test_index in kf.split(train_y):
        i += 1
        tr_X = train_x_CNN[train_index, :, :, :]
        tr_Y = train_y[train_index]
        ts_X = train_x_CNN[test_index, :, :, :]
        ts_Y = train_y[test_index]
        # print(tr_X.shape,tr_Y.shape,ts_X.shape,ts_Y.shape)

        model = CNN2d_model(tr_X, filter)
        history = model.fit(tr_X, tr_Y, epochs=epochs, batch_size=batch_size,
validation_data=(ts_X, ts_Y), verbose=0,
                            shuffle=False, callbacks=call)
        h.append(history.epoch[-1])
        yhat_test = model.predict(ts_X)
        yhat_train = model.predict(tr_X)
        mse_test = np.mean((yhat_test – ts_Y) ** 2)
        mse_train = np.mean((yhat_train – tr_Y) ** 2)

        if mse_test < mse_train:
            wrong_model += 1

        if verbose>0:
            if mse_test < mse_train:
                print("#", i, "Wrong CNN2D model for output ", DF_Y.name)
                print("Train loss", mse_train)
                print("Test loss", mse_test)
            else:
                print("#", i, "OK CNN2D model for output ", DF_Y.name)

        del model
        cw_r_test[test_index] = yhat_test
        cw_r_train[train_index] = yhat_train
    if wrong_model > 0:
        print("Wrong CNN2D model for output ", DF_Y.name, wrong_model *
100 // n_splits, "% models wrong")
    else:
```

```
        print("OK for all CNN2D model")

    forecast_test = scaler_y_CNN.inverse_transform(cw_r_test)
    forecast_train = scaler_y_CNN.inverse_transform(cw_r_train)

    res_test = pd.DataFrame(forecast_test)
    res_test.index = y_CNN.index
    res_test.columns = ['CV_Test_CNN2D']

    res_train = pd.DataFrame(forecast_train)
    res_train.index = y_CNN.index
    res_train.columns = ['CV_Train_CNN2D']

    mse_test = np.mean((res_test.values – y_CNN.values) ** 2)
    mse_train = np.mean((res_train.values – y_CNN.values) ** 2)
    return res_train, res_test, mse_train, mse_test, h
```

hurricane_data_load.py:

```
import pandas as pd
from datetime import datetime

def parse(x):
    y=x.split()
    t=int(y[-1])
    h=t//100
    m=t%100
    z=" ".join(y[:-1])
    z=z+" "+str(h)+" "+str(m)

    #print(x, x.split(), z, h, m, z+" "+str(h)+" "+str(m))
    return datetime.strptime(z, '%Y %m %d %H %M')

def parse2(x):

    #print(x, x.split(), z, h, m, z+" "+str(h)+" "+str(m))
    return datetime.strptime(x, '%Y %m %d')

def parse3(x):
    y=x.split()
```

```
    Y=y[0][:4]
    M=y[0][4:7]
    D=y[0][7:]
    t=int(y[-1])
    h=t//10000
    m=int(y[-1][-4:-2])
    z=Y+" "+M+ " " + D +" "+str(h)+" "+str(m)
    #print(z)

    #print(x, x.split(), z, h, m, z+" "+str(h)+" "+str(m))
    return datetime.strptime(z, '%Y %b %d %H %M')

dt_GOES15=pd.read_excel('Input_DATABASES .xlsx', 'GOES-15',parse_dates
= [['YEAR', 'MONTH', 'DAY', 'HHMM']], index_col=0, date_parser=parse)
dt_GOES15.index.name = 'date'
dt_GOES15=dt_GOES15.drop(["Day", "Day.1"], axis=1)
#print(dt_GOES15.head())

dt_GOES13=pd.read_excel('Input_DATABASES .xlsx', 'GOES-13',parse_dates
= [['YEAR', 'MONTH', 'DAY', 'HHMM']], index_col=0, date_parser=parse)
dt_GOES13.index.name = 'date'
dt_GOES13=dt_GOES13.drop(["Day", "Day.1"], axis=1)
#print(dt_GOES13.head())

dt_FLUX=pd.read_excel('Input_DATABASES .xlsx', 'Flux Protons',parse_dates
= [['YEAR', 'MONTH', 'DAY', 'HHMM']], index_col=0, date_parser=parse)
dt_FLUX.index.name = 'date'
dt_FLUX=dt_FLUX.drop(["Day", "Day.1", "S", "S.1"], axis=1)
#print(dt_FLUX.head())

dt_SD=pd.read_excel('Input_DATABASES .xlsx', 'Solar Data',parse_dates =
[['YEAR', 'MONTH', 'DAY']], index_col=0, date_parser=parse2)
dt_SD.index.name = 'date'
#print(dt_SD.head())

dt_WP=pd.read_excel('Input_DATABASES .xlsx', 'Wind Plasma',parse_dates =
[['YEAR', 'MONTH', 'DAY','HHMM']], index_col=0, date_parser=parse)
dt_WP.index.name = 'date'
dt_WP=dt_WP.drop(["Day", "Day.1", "S"], axis=1)
#print(dt_WP.head(350))

dt_IRMA=pd.read_excel('20191104_HURRICANES_2017.xlsx',
'IRMA',parse_dates = [['date', 'time UTC']], index_col=0, date_parser=parse3)
```

```
dt_IRMA.index.name = 'date'
dt_IRMA=dt_IRMA.drop(["CI"], axis=1)
#print(dt_IRMA.head())

dt_JOSE=pd.read_excel('20191104_HURRICANES_2017.xlsx',
'JOSE',parse_dates = [['date', 'time UTC']], index_col=0, date_parser=parse3)
dt_JOSE.index.name = 'date'
dt_JOSE=dt_JOSE.drop(["CI"], axis=1)
#print(dt_JOSE.head())

dt_KATIA=pd.read_excel('20191104_HURRICANES_2017.xlsx',
'KATIA',parse_dates = [['date', 'time UTC']], index_col=0, date_parser=parse3)
dt_KATIA.index.name = 'date'
dt_KATIA=dt_KATIA.drop(["CI"], axis=1)
#print(dt_KATIA.head())

DS_GOES15=dt_GOES15.copy()
DS_GOES15=DS_GOES15.join(dt_SD, how='outer' )
DS_GOES15=DS_GOES15.join(dt_WP, how='outer' )
DS_GOES15=DS_GOES15.join(dt_IRMA, how='outer' )
DS_GOES15=DS_GOES15.join(dt_JOSE, how='outer' )
DS_GOES15=DS_GOES15.join(dt_KATIA, how='outer' )

DS_GOES13=dt_GOES13.copy()
DS_GOES13=DS_GOES13.join(dt_SD, how='outer' )
DS_GOES13=DS_GOES13.join(dt_WP, how='outer' )
DS_GOES13=DS_GOES13.join(dt_IRMA, how='outer' )
DS_GOES13=DS_GOES13.join(dt_JOSE, how='outer' )
DS_GOES13=DS_GOES13.join(dt_KATIA, how='outer' )

DS_FLUX=dt_FLUX.copy()
DS_FLUX=DS_FLUX.join(dt_SD, how='outer' )
DS_FLUX=DS_FLUX.join(dt_WP, how='outer' )
DS_FLUX=DS_FLUX.join(dt_IRMA, how='outer' )
DS_FLUX=DS_FLUX.join(dt_JOSE, how='outer' )
DS_FLUX=DS_FLUX.join(dt_KATIA, how='outer' )

writer = pd.ExcelWriter('DataSet.xlsx')
DS_GOES15.to_excel(writer,'DS_GOES15')
DS_GOES13.to_excel(writer,'DS_GOES13')
DS_FLUX.to_excel(writer,'DS_FLUX')
writer.save()
```

hurican_analysis.py

```
import pandas as pd
import os
import matplotlib.pyplot as plt
import seaborn as sns
import data_analysis

# tensorboard --logdir=logs


T_DISCRETTE = 30
# sheet='DS_GOES15'
# sheet='DS_GOES13'
sheet = 'DS_FLUX'

# створення папки для файлів моделей
if not os.path.exists("./" + sheet):
    try:
        os.mkdir("./" + sheet)
    except OSError:
        print("Creation of the directory %s failed" % sheet)
    else:
        print("Successfully created the directory %s " % sheet)

DS = pd.read_excel('DataSet.xlsx', sheet, index_col=0)
DS.index = pd.to_datetime(DS.index)

# DS=DS[DS.index<date_before] #Обрізка даних дати


# Усереднення данних
DS_res = DS.resample(str(T_DISCRETTE) + 'T').max()
DS_res = data_analysis.interpolate(DS_res, method='pchip')

# ковзне вікно
# DS_res=DS_res.rolling(4).mean()
print(DS.head())

DF_C = DS_res.corr()

sns.heatmap(DF_C)
plt.show()
```

```
writer = pd.ExcelWriter(sheet + '/Correlation.xlsx')
DS_res.to_excel(writer, "DataSet")
DF_C.to_excel(writer, sheet)
writer.save()

# plt.figure(1)
# plt.subplot(211)
# data_analysis.my_plt(DS_res, plt,[0,1,2,3,4,5], True,"upper right", sheet)
# plt.subplot(212)
# data_analysis.my_plt(DS_res, plt,[6,7], True,"upper right", sheet)
#
# plt.figure(2)
# plt.subplot(211)
# data_analysis.my_plt(DS_res, plt,[8,10], True,"upper right")
# plt.subplot(212)
# data_analysis.my_plt(DS_res, plt,[9,11], True,"upper right")
#
# plt.figure(3)
# plt.subplot(211)
# data_analysis.my_plt(DS_res, plt,[12,14,16], False,"upper right")
# plt.subplot(212)
# data_analysis.my_plt(DS_res, plt,[13,15, 17], False,"upper right")


plt.figure(1)
data_analysis.my_plt(DS_res, plt, [0, 1], True, "upper right", sheet)

plt.figure(2)
plt.subplot(211)
data_analysis.my_plt(DS_res, plt, [2, 4], True, "upper right")
plt.subplot(212)
data_analysis.my_plt(DS_res, plt, [3, 5], True, "upper right")

plt.figure(3)
plt.subplot(211)
data_analysis.my_plt(DS_res, plt, [6, 8, 10], False, "upper right")
plt.subplot(212)
data_analysis.my_plt(DS_res, plt, [7, 9, 11], False, "upper right")

plt.show()
```

hurricane_lag_analysis.py

```
import pandas as pd
import data_analysis as da


folder_i=2
folder=['DS_GOES15', 'DS_GOES13','DS_FLUX']
fields={folder[0]: ['P >30', 'Radio Flux 10.7', 'proton density', 'bulk speed','ion
temperature'],
      folder[1]: ['P >30', 'Radio Flux 10.7', 'proton density', 'bulk speed','ion
temperature'],
      folder[2]: ['> 30 MeV', 'Radio Flux 10.7', 'proton density', 'bulk speed','ion
temperature'],
      }

dt=pd.read_excel(folder[folder_i]+'/Correlation.xlsx', 'DataSet', index_col=0)
dt_input=dt[fields[folder[folder_i]]]
dt_output=dt[dt.columns[-6:]]

print(dt.columns)
print(dt_input.columns)
print(dt_output.columns)

da.lag_correlation(dt_input, dt_output, folder[folder_i]+'/Lag_Correlation.xlsx',
lag=20)
```

hurricane_mod.py

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
from pandas import DataFrame
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from numpy import concatenate
import data_analysis as da

folder_i = 0
```

```
folder = ['DS_GOES15', 'DS_GOES13', 'DS_FLUX']
fields = {folder[0]: ['P >30', 'Radio Flux 10.7', 'proton density', 'bulk speed', 'ion
temperature'],
        folder[1]: ['P >30', 'Radio Flux 10.7', 'proton density', 'bulk speed', 'ion
temperature'],
        folder[2]: ['> 30 MeV', 'Radio Flux 10.7', 'proton density', 'bulk speed',
'ion temperature'],
        }

dt = pd.read_excel(folder[folder_i] + '/Correlation.xlsx', 'DataSet', index_col=0)
dt_input = dt[fields[folder[folder_i]]]
dt_output = dt[dt.columns[-6:]]
leg = [['Real Data'] for i in range(len(dt_output.columns))]

results = {}
for tr_i, tr in enumerate(dt_output.columns):
   plt.figure(tr_i)
   plt.title(tr)
   plt.plot(dt_output[tr])
   results[tr] = pd.DataFrame(dt_output[tr])

print(results.keys())

for m_i, m in enumerate(folder):
   dt = pd.read_excel(m + '/Correlation.xlsx', 'DataSet', index_col=0)
   dt_input = dt[fields[m]]
   print("Model:", m)

   for tr_i, tr in enumerate(dt_output.columns):
      plt.figure(tr_i)

      # res=da.Linear_single_model(dt_input, dt_output[tr], target_dep=True,
lag_in=10, train_size=0.7)
      # res=da.Linear_crossvalidation(dt_input, dt_output[tr], target_dep=True,
lag_in=10, n_splits=5)
      res, h = da.LSTM_crossvalidation(dt_input, dt_output[tr],
target_dep=False, n_splits=5, lag_in=240, neurons=300,
                                       epochs=400, patience=10)
      print("Max epoch", max(h))
      # res=da.LSTM_single_model(dt_input, dt_output[tr], target_dep=True,
lag_in=10, train_size=0.7, neurons=7, epochs=400, patience=10)
      res.columns = [n + ' ' + m for n in res]
      results[tr] = results[tr].join(res, how='outer')
```

```
        leg[tr_i].append(m)
        # plt.plot(dt_output[tr])
        plt.plot(res)
        plt.legend(leg[tr_i])

plt.show()

writer = pd.ExcelWriter('Results.xlsx')
for s, v in results.items():
    v.dropna(inplace=True, how='any')
    v.to_excel(writer, s)
writer.save()
```

hurricane_peak.py

```
import pandas as pd
from scipy.signal import find_peaks
# import os
import matplotlib.pyplot as plt
# import data_analysis
# from sklearn import preprocessing
import numpy as np

# tensorboard --logdir=logs

DS = pd.read_excel('Wind_on_Speed/DataSet.xlsx', 0, index_col=0)
DS.index = pd.to_datetime(DS.index)
print(DS.head())

for i, c in enumerate(DS.columns):
    x = DS[c]
    print(c)
    peaks, _ = find_peaks(x, width=20)
    plt.figure(i + 1)
    plt.title(c)
    plt.plot(x)
    plt.plot(x.iloc[peaks], "x")
    plt.savefig('Wind_on_Speed/' + c)
    x.iloc[peaks] = 1
    x[x.values != 1] = 0
```

```
# plt.plot(np.zeros_like(x), "--", color="gray")
plt.show()

writer = pd.ExcelWriter('Wind_on_Speed/Peaks.xlsx')
DS.to_excel(writer, "DataSet")
writer.save()
```

flood_load.py

```
import pandas as pd
from scipy.signal import find_peaks
# import os
import matplotlib.pyplot as plt
# import data_analysis
# from sklearn import preprocessing
import numpy as np

# tensorboard --logdir=logs

DS = pd.read_excel('Wind_on_Speed/DataSet.xlsx', 0, index_col=0)
DS.index = pd.to_datetime(DS.index)
print(DS.head())

for i, c in enumerate(DS.columns):
    x = DS[c]
    print(c)
    peaks, _ = find_peaks(x, width=20)
    plt.figure(i + 1)
    plt.title(c)
    plt.plot(x)
    plt.plot(x.iloc[peaks], "x")
    plt.savefig('Wind_on_Speed/' + c)
    x.iloc[peaks] = 1
    x[x.values != 1] = 0
# plt.plot(np.zeros_like(x), "--", color="gray")
plt.show()

writer = pd.ExcelWriter('Wind_on_Speed/Peaks.xlsx')
DS.to_excel(writer, "DataSet")
writer.save()
```

flood_corr_analys.py

```
import glob
import pandas as pd

folder = 'corr'


def lag_analisys(file):
    df = pd.read_excel(file, index_col=0)
    cor = df.max()
    lag = df.idxmax()
    f = file[file.find("/") + 1:file.find("_corr")]
    cor.name = lag.name = f
    return cor, lag


def lag_choice(folder, file_type):
    files_max = glob.glob(folder + "/*" + file_type + ".xlsx")
    df_c = pd.DataFrame()
    df_l = pd.DataFrame()
    for file in files_max:
        print(file)
        c, l = lag_analisys(file)
        df_c = df_c.join(c, how='outer')
        df_l = df_l.join(l, how='outer')
    return df_c, df_l


c, l = lag_choice(folder, 'delta')
writer = pd.ExcelWriter('DataSet_delta_lag.xlsx')
c.to_excel(writer, 'correlation')
l.to_excel(writer, 'lag')
writer.save()

c, l = lag_choice(folder, 'max')
writer = pd.ExcelWriter('DataSet_max_lag.xlsx')
c.to_excel(writer, 'correlation')
l.to_excel(writer, 'lag')
writer.save()
```

flood_peak.py

```
import pandas as pd
from scipy.signal import find_peaks
import matplotlib.pyplot as plt

xl = pd.ExcelFile('DataSet.xlsx')
writer = pd.ExcelWriter('Peaks.xlsx')

for sh in xl.sheet_names:  # see all sheet names
   if 'max' in sh:
      DS = pd.read_excel('DataSet.xlsx', sh, index_col=0)
      DS.index = pd.to_datetime(DS.index)
      for i, c in enumerate(DS.columns[:-1]):
         x = DS[c]
         # print(c)
         # peaks, _ = find_peaks(x, threshold= (np.max(x)-np.min(x))/10)
         peaks, _ = find_peaks(x)
         plt.figure(i + 1)
         plt.title(c)
         plt.plot(x)
         plt.plot(x.iloc[peaks], "x")
         # plt.savefig(c)
         x.iloc[peaks] = 1
         x[x.values != 1] = 0
      # plt.plot(np.zeros_like(x), "--", color="gray")
      plt.show()
      DS[DS.columns[-1]] = DS[DS.columns[-1]].apply(lambda x: 0 if x != 0
else 1)
      DS.to_excel(writer, sh)
writer.save()
```

flood_decision_tree.py

```
import pandas as pd
import matplotlib.pyplot as plt
import graphviz
# Classifiers
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.discriminant_analysis import QuadraticDiscriminantAnalysis
from sklearn.model_selection import cross_val_score
from sklearn.ensemble import VotingClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import plot_confusion_matrix
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn import tree

ex = pd.read_excel('Peaks_S2.xlsx', sheet_name=None)
df = pd.DataFrame()
for i, s in ex.items():
    df = df.append(pd.DataFrame(s))

df_short = df[['date', '> 10 MeV', '> 30 MeV', '38-53', '175-315', '310-580',
'PROTON DENSITY', 'BULK SPEED',
            'ION TEMPERATURE', '10.7 cm Radio Flux', 'precipitations',
            'days from the beginning of the flood']]

col = df_short.columns
x, y = df_short[col[1:-1]], df_short[col[-1]]

clf = LogisticRegression()
scores = cross_val_score(clf, x, y, scoring='accuracy', cv=5)
print("Accuracy: %0.2f (+/- %0.2f) [%s]" % (scores.mean(), scores.std(),
'Logistic Regression'))

clf1 = LogisticRegression(random_state=1)
clf2 = QuadraticDiscriminantAnalysis()
clf3 = GaussianNB()
clf4 = RandomForestClassifier(max_depth=5, n_estimators=10,
max_features=1)
clf5 = AdaBoostClassifier()
```

```
for clf, label in zip([clf1, clf2, clf3, clf4, clf5],
                      ['Logistic Regression', 'Quadratic Discriminant Analysis', 'naive
Bayes', 'Random Forest',
                       'Ada Bust']):
   scores = cross_val_score(clf, x, y, scoring='accuracy', cv=5)
   print("Accuracy: %0.2f (+/- %0.2f) [%s]" % (scores.mean(), scores.std(),
label))

clf = DecisionTreeClassifier()
scores = cross_val_score(clf, x, y, scoring='accuracy', cv=5)
print("Accuracy: %0.2f (+/- %0.2f) [%s]" % (scores.mean(), scores.std(),
'DecisionTreeClassifier'))
clf.fit(x, y)

dot_data = tree.export_graphviz(clf, feature_names=x.columns,
class_names=['No', 'Flood'], filled=True)
graph = graphviz.Source(dot_data, format="png")
graph.render("decision_tree_graphivz")

feat_importances = pd.Series(clf.feature_importances_, index=x.columns)
feat_importances.nlargest(10).plot(kind='barh')
plt.show()

print(feat_importances)
```