

>>>>

SolarShield AI – система прогнозирования солнечных бурь

AI system



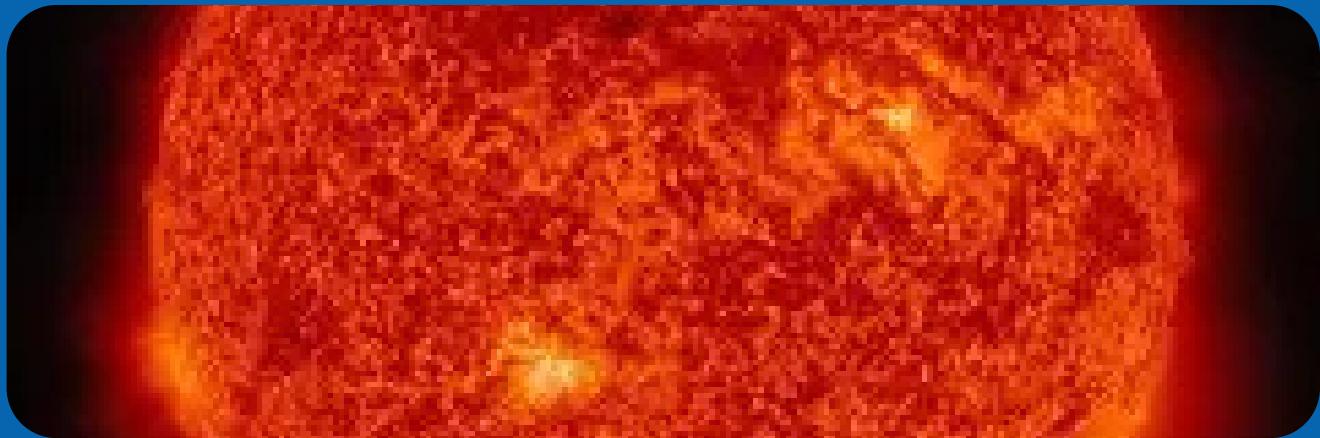
СОЛНЕЧНЫЕ ВСПЫШКИ И ГЕОМАГНИТНЫЕ БУРИ:

КАЖДЫЙ ГОД КОСМИЧЕСКАЯ
ИНФРАСТРУКТУРА СТАНОВИТСЯ БОЛЕЕ
УЯЗВИМОЙ.

- повреждают спутники
- нарушают GPS и связь
- создают радиационные риски



МАСШТАБ ПРОБЛЕМЫ



- Более 10 000 активных спутников на орбите
 - Рост коммерческих космических миссий
 - Ущерб от сильных бурь может достигать миллионов долларов
- ! Необходима система раннего предупреждения.

НАШЕ РЕШЕНИЕ

SolarShield AI
— это
система,
которая:

анализирует
данные
солнечной
активности

обрабатывает
магнитные
показатели

предсказывает
вероятность
солнечной
бури

Результат:

⚠ Предупреждение за несколько часов или дней



Supervised Machine Learning
Random Forest Model

Input:

- Solar Flux
- Magnetic Index
- Solar Wind Speed

Output:

- Probability of solar storm



КАК РАБОТАЕТ ИИ?



```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier

np.random.seed(42)

def make_data(n=600):
    solar_flux = np.random.normal(150, 50, n)
    magnetic = np.random.normal(5, 2, n)
    wind = np.random.normal(400, 100, n)

    storm = ((solar_flux > 180) & (magnetic > 6) & (wind > 450)).astype(int)

    return pd.DataFrame({
        "solar_flux": solar_flux,
        "magnetic_index": magnetic,
        "solar_wind_speed": wind,
        "storm": storm
    })

def train():
    df = make_data()
    X = df[["solar_flux", "magnetic_index", "solar_wind_speed"]]
    y = df["storm"]

    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    model = RandomForestClassifier(n_estimators=150, random_state=42)
    #model = RandomForestClassifier(n_estimators=150, random_state=42)
    model.fit(X_train, y_train)

    acc = model.score(X_test, y_test)
    print(f"Model accuracy: {acc:.2f}")
    return model

def ask_number(text):
    while True:
        try:
            return float(input(text).strip().replace(",", "."))
        except:
            print("Введите число, например 180 или 180.5")

def main():
    print("SolarShield AI - console MVP\n")
    model = train()

    while True:
        f = ask_number("Solar flux: ")
        m = ask_number("Magnetic index: ")
        w = ask_number("Solar wind speed: ")

        sample = pd.DataFrame({
            "solar_flux": [f],
            "magnetic_index": [m],
            "solar_wind_speed": [w]
        })
        risk = model.predict_proba(sample)[0][1] * 100
        print(f"\nRisk of solar storm: {risk:.1f}%")

        if risk >= 60:
            print("⚠ Warning: high risk!")
        else:
            print("✅ Low risk.")

        again = input("\nЕще раз? (y/n): ").lower().strip()
        if again not in ("y", "yes", "д", "да"):
            break
        print()

    if __name__ == "__main__":
        main()

...
*** SolarShield AI - console MVP

Model accuracy: 0.97
Solar flux: 300
Magnetic index: 10
Solar wind speed: 900

Risk of solar storm: 86.0%
⚠ Warning: high risk!

Еще раз? (y/n): 
```

MVP

MODEL ACCURACY: 97%
RISK PREDICTION: 78%
⚠ HIGH RISK DETECTED



DEMO—HOW IT WORKS

User enters data:

- Solar flux 300
- Magnetic index 10
- Solar wind speed 900

AI model analyzes parameters
System shows risk level (%)

Example:

Risk of solar storm: 86%

⚠ High risk

... SolarShield AI – console MVP

Model accuracy: 0.97

Solar flux: 300

Magnetic index: 10

Solar wind speed: 900

Risk of solar storm: 86.0%

⚠ Warning: high risk!

Еще раз? (y/n):

БИЗНЕС МОДЕЛЬ

Клиенты:

- спутниковые компании
- космические агентства
- страховые компании

Монетизация:

- подписка на API
- SaaS-платформа
- лицензирование технологии



НОВИЗНА И БУДУЩЕЕ



Новизна:

- AI-прогнозирование солнечных бурь
- Автоматическое предупреждение

Будущее:

- Реальные данные NASA
- Использование в лунных миссиях

SOLARSHIELD AI – ЭТО
ШАГ К БЕЗОПАСНОМУ
БУДУЩЕМУ КОСМОСА.

SOLARSHIELD AI
TEAM J3N
AEROO SPACE AI
COMPETITION

