

БОЙКО СЕРГІЙ

Національний університет «Запорізька політехніка»

e-mail: boyko_s_n@ukr.net

КАСАТКІНА ІРИНА

Криворізький національний університет

e-mail: ms.irina.vital@hmail.com

ЯНИЦЬКИЙ АНАТОЛІЙ

Кременчуцький льотний коледж Харківського національного університету внутрішніх справ

e-mail: bsn1987@i.ua

САМОХЛІБ ОЛЕКСАНДР

Кременчуцький льотний коледж Харківського національного університету внутрішніх справ

e-mail: boikosn2017@gmail.com

ДІАГНОСТИКА СИЛОВОЇ УСТАНОВКИ БЕЗПІЛОТНИХ ЛІТАЛЬНИХ АПАРАТІВ НА БАЗІ ЕЛЕКТРОПРИВОДУ

У статті розглянуто питання діагностики силової установки безпілотних літальних апаратів (БПЛА), що функціонують на базі електроприводу, з використанням інтелектуальних технологій аналізу даних. У контексті сучасних викликів, зумовлених збройною агресією проти України, актуальність дослідження зумовлена необхідністю забезпечення безпечної експлуатації автономних літальних систем у регіонах із пошкодженою інфраструктурою та обмеженою функціональністю телекомунікаційних мереж. Визначено, що ефективність функціонування БПЛА значною мірою залежить від стану його силової установки, а отже, від своєчасності та точності діагностування електроприводів у процесі польоту.

Основну увагу зосереджено на розробленні концептуальних засад побудови системи моніторингу технічного стану електроприводу на базі штучних нейронних мереж (ШНМ) із використанням середовища TensorFlow. Проведено аналіз сучасних тенденцій розвитку технологій глибокого навчання, які забезпечують можливість розпізнавання нелінійних залежностей між параметрами функціонування силової установки та її діагностичними показниками. Розкрито структуру типової нейронної мережі, що включає вхідний, прихований та вихідний шари, які реалізують нелінійні перетворення вхідних сигналів. Охарактеризовано алгоритми оптимізації навчального процесу (зокрема стохастичний градієнтний спуск, Adam, RMSProp) та функції втрат, що застосовуються для класифікації технічного стану електричних двигунів під час експлуатації.

У роботі обґрунтовано доцільність використання API TensorFlow Keras для побудови, тренування та тестування моделей діагностики силових установок БПЛА. Показано, що запропонований підхід дає змогу визначати з високою точністю рівень працездатності електродвигунів у реальному часі, виявляти відхилення від номінального режиму роботи та прогнозувати можливі відмови. Зокрема, експериментальні результати імітаційного моделювання підтвердили, що розроблена модель здатна класифікувати стан силової установки за трьома категоріями — справний, умовно працездатний і критичний — із точністю понад 95 %.

Детально розглянуто процес побудови нейронної моделі, включно з підбором гіперпараметрів, кількістю прихованих шарів і нейронів, вибором функцій активації (ReLU, Softmax) та параметрів навчання. Підкреслено роль апаратного прискорення (GPU, TPU) для забезпечення оперативної обробки великого обсягу діагностичних даних у режимі реального часу. Наведено результати аналізу впливу архітектури нейромережі на швидкість збіжності алгоритму навчання та стійкість до перенавчання.

Зроблено висновок, що запропонований метод діагностики силової установки на базі електроприводу з використанням штучних нейронних мереж є ефективним інструментом для підвищення надійності експлуатації безпілотних літальних апаратів, особливо в умовах обмеженої телекомунікаційної підтримки. Практичне впровадження таких систем дозволить забезпечити автономний контроль технічного стану силових елементів, своєчасне виявлення несправностей і запобігання аварійним ситуаціям під час польоту. Отримані результати мають значний науково-практичний потенціал для подальшої розробки комплексних систем технічної діагностики, інтегрованих у структуру систем авіапілота БПЛА.

Ключові слова: безпілотний літальний апарат, електропривід, силова установка, діагностика, штучна нейронна мережа, TensorFlow, глибоке навчання, технічний стан, моніторинг, безпека експлуатації.

BOYKO SERHII

National University "Zaporizhzhya Polytechnic"

KASATKINA IRINA

Kryvyi Rih National University

YANITSKYI ANATOLI**SAMOHLYB OLEKSANDER**

Kremenchuk Flight College of Kharkiv National University of Internal Affairs

DIAGNOSIS OF THE POWER PLANT OF UNMANNED AERIAL VEHICLES BASED ON ELECTRIC DRIVE

The article addresses the issue of diagnosing the power plant of unmanned aerial vehicles (UAVs) based on electric drives using intelligent data analysis technologies. In the context of modern challenges caused by military aggression against Ukraine, the relevance of this research is determined by the need to ensure the safe operation of autonomous aerial systems in regions with damaged infrastructure and limited telecommunication functionality. It has been established that the efficiency of UAV operation largely depends on the condition of its power unit, and therefore, on the timeliness and accuracy of diagnosing electric drives during flight.

The main focus is placed on the development of conceptual foundations for constructing a monitoring system of the technical condition of the electric drive using artificial neural networks (ANN) within the TensorFlow environment. The study analyzes current trends in deep learning technologies that enable the recognition of nonlinear dependencies between the operating parameters of the power plant and its diagnostic indicators. The structure of a typical neural network, including input, hidden, and output layers that perform nonlinear transformations of input signals, is described. Optimization algorithms for the training process (including stochastic gradient descent, Adam, RMSProp) and loss functions applied for the classification of the technical condition of electric motors during operation are characterized.

The research substantiates the feasibility of using the TensorFlow Keras API for the construction, training, and testing of diagnostic models for UAV power plants. It is shown that the proposed approach allows for high-accuracy determination of the operational state of electric motors in real time, detection of deviations from nominal operating modes, and prediction of potential failures. Simulation results confirm that the developed model can classify the power unit condition into three categories—operational, conditionally operational, and critical—with an accuracy exceeding 95%.

The process of constructing the neural network model is described in detail, including the selection of hyperparameters, the number of hidden layers and neurons, activation functions (ReLU, Softmax), and training parameters. The importance of hardware acceleration (GPU, TPU) is emphasized for ensuring real-time processing of large volumes of diagnostic data. The results of the analysis of the influence of neural network architecture on the convergence rate of the training algorithm and resistance to overfitting are presented.

It is concluded that the proposed diagnostic method for power plants based on electric drives using artificial neural networks is an effective tool for enhancing the reliability of unmanned aerial vehicle operation, particularly under conditions of limited telecommunication support. The practical implementation of such systems enables autonomous monitoring of the technical state of power elements, timely detection of faults, and prevention of emergency situations during flight. The obtained results have significant scientific and practical potential for further development of integrated diagnostic systems incorporated into UAV autopilot architectures.

Keywords: *unmanned aerial vehicle, electric drive, power plant, diagnostics, artificial neural network, TensorFlow, deep learning, technical condition, monitoring, operational safety.*

Стаття надійшла до редакції / Received 18.10.2025

Прийнята до друку / Accepted 15.11.2025

Постановка проблеми

Зважаючи на реалії повномасштабного вторгнення, галузі економіки України поступово змінюються. Транспортна галузь стала однією з небагатьох галузей котрі зазнали логістичних, інфраструктурних та концептуальних змін подальшого функціонування. Особливо зміни торкнулися питань щодо впровадження сучасних систем диспетчерзації та керування. Між тим, зміни торкнулися і транспортного парку. Почали впроваджуватися у експлуатацію на теренах країни літальні апарати з можливістю повного керування без людини.

Впровадження зазначених транспортних засобів мають як позитивні аспекти так і деякі особливості впровадження. Між тим, у світовому контексті ці питання вирішуються шляхом удосконалення програмного забезпечення та модернізації засобів контролю простору навколо літального апарату [1, 2].

Одним із основних факторів експлуатації транспортних засобів є безпека їх функціонування. Тому велика кількість сучасних розробок у цьому напрямі спрямована на підвищення безпеки експлуатації транспортних засобів всіх типів.

Безумовно, враховуючи розвиток сучасних цифрових та телекомунікаційних технологій, транспортні засоби з автономним управлінням мають можливість безпечно функціонувати. Але одним із важливих умов експлуатації безпілотних літальних апаратів є їх технічний стан, що потребує регулярної, а іноді постійної діагностики. У зв'язку з чим є необхідність вдосконалення систем автопілоту безпілотних літальних апаратів для експлуатації в таких регіонах [3, 4].

Основною метою дослідження є розробка концептуальних засад, котрі здатні забезпечити ефективне та безпечне функціонування безпілотних літальних апаратів в регіонах на території яких проходять збройні конфлікти унаслідок чого мережі сучасних цифрових та телекомунікаційних технологій мають неповну функціональність чи взагалі не функціонують. Метою дослідження є розробка та проведення експериментів на імітаційній моделі з подальшим аналізом ефективності використання інтелектуального програмного забезпечення, котре базується на штучних нейронних мережах та мають можливість врахування перешкод та сигналів із діагностичного обладнання силової установки безпілотних літальних апаратів. Особливістю запропонованих концептуальних засад є те, що вони дають можливість підвищити рівень безпечної експлуатації безпілотних літальних апаратів шляхом розпізнавання погіршення функціонування електричних двигунів під час польоту, котрі входять до складу силової установки безпілотних літальних апаратів [5, 6].

Аналіз останніх досліджень і публікацій

На сьогоднішній день, активно впроваджуються у різні галузі економіки програмні продукти штучного інтелекту, котрі мають високі функціональні можливості в порівнянні з існуючими підходами. Впровадження таких програмних продуктів дає можливість пришвидшити обробку інформації та побудувати інтелектуальні віртуальні системи здатні до аналізу даних, а відтак і процесів.

Безумовно, такий підхід має враховувати позитивні та негативні аспекти впровадження таких програмних продуктів.

Однак, слід зазначити, що врахування аспектів безпеки експлуатації літальних апаратів, у тому числі безпілотних, має стимулювати впровадження сучасних систем та підходів до побудови надійних та потенційно ефективних безпекових систем [1-5].

TensorFlow — це фреймворк другого покоління, розроблений командою Google Brain, який у 2015 році було випущено як open-source програмне забезпечення. Основною метою створення TensorFlow стала необхідність забезпечити більш високу продуктивність, модульність і масштабованість порівняно з

попередніми інструментами машинного навчання Google, такими як DistBelief. Завдяки оптимізованій архітектурі, платформа забезпечує спрощену інтеграцію у наукові, промислові та мобільні проєкти.

TensorFlow орієнтована на широке коло задач машинного навчання, охоплюючи як прості моделі для мобільних пристроїв, так і високонавантажені обчислювальні системи, що функціонують на тисячах обчислювальних вузлів у дата-центрах. Це дозволяє успішно застосовувати бібліотеку в задачах розпізнавання образів, обробки природної мови, генерації тексту, автоматизованих систем відповідей (наприклад, у сервісі Google Inbox), класифікації об'єктів, рекомендаційних системах тощо.

Однією з ключових переваг TensorFlow є його обчислювальна модель на основі графів потоків даних (dataflow graphs). У цій моделі вузли графа відповідають за виконання математичних операцій, а ребра описують тензори — багатовимірні масиви даних, які передаються між вузлами. Саме від терміна "тензор" походить назва платформи: Tensor — структура даних, що представляє багатовимірні масиви, і Flow — механізм передачі цих масивів через граф обчислень.

Висока гнучкість TensorFlow дозволяє досягати до п'ятикратного приросту у швидкості тренування нейронних мереж у порівнянні з попередніми платформами Google. Це забезпечує оперативне розгортання та експлуатацію складних моделей, зокрема у задачах комп'ютерного зору, прогнозування та рекомендацій. Платформа також підтримує розподілені обчислення, апаратне прискорення через GPU та TPU, а також автоматичне диференціювання, що робить її придатною для реалізації як простих моделей, так і глибоких нейронних мереж з багатьма шарами [6-7].

На сьогодні TensorFlow є еталонним середовищем для розробки та навчання нейронних мереж, яке підтримує виконання як на одиничних пристроях, так і у багатопроесорних конфігураціях. Середовище повністю сумісне з обчисленнями на графічних процесорах завдяки інтеграції з технологією NVIDIA CUDA, що забезпечує високопродуктивну обробку задач глибокого навчання. TensorFlow підтримується на 64-розрядних платформах Linux, Windows, macOS, а також має реалізації для мобільних операційних систем Android і iOS, що дозволяє реалізовувати повнофункціональні нейронні моделі навіть на вбудованих пристроях.

Порівняно з попередніми розробками компанії Google, TensorFlow є узагальненим середовищем машинного навчання, яке не обмежується лише задачами нейронних мереж, а дозволяє реалізовувати будь-які типи обчислень на основі графів потоків даних. Така гнучка архітектура істотно підвищує ефективність реалізації складних алгоритмів, включаючи адаптивне моделювання та ітераційне навчання.

У сучасній практиці глибокого навчання нейромережеві архітектури демонструють найвищу ефективність у вирішенні задач, що традиційно характеризуються високою складністю формалізації: розпізнавання образів, класифікація об'єктів, обробка природної мови, моделювання поведінки користувачів тощо. Це досягається завдяки здатності нейромережі самостійно визначати латентні закономірності між вхідними ознаками та цільовими мітками, що унеможливлено в класичних алгоритмічних системах.

Особливу увагу заслуговують високорівневі API та інструменти TensorFlow, зокрема Keras, які спрощують процес побудови, навчання та валідації моделей. Суттєвою перевагою є здатність нейромереж до самонавчання з мінімальним втручанням у структуру моделі — оптимізація вагових коефіцієнтів у процесі градієнтного спуску дозволяє мінімізувати людське втручання в процес налаштування.

Для забезпечення ефективного навчання великих моделей критично важливо використовувати апаратне прискорення — сучасні графічні процесори (GPU), тензорні процесори (TPU) та інші спеціалізовані архітектури. Саме завдяки цим засобам навчання нейронних мереж, що містять десятки мільйонів параметрів, стало можливим у прийнятні часові рамки.

Сама архітектура штучної нейронної мережі у TensorFlow базується на багатошаровій структурі, яка містить вхідний, один або кілька прихованих, та вихідний шари. У типовій реалізації використовуються щільні (Dense) або повнозв'язані (Fully Connected) шари, в яких кожен нейрон певного шару має з'єднання з усіма нейронами попереднього шару. Така архітектура забезпечує високу гнучкість моделі та дозволяє адаптуватися до різних типів задач — від регресії до багатокласової класифікації.

Одним із ключових етапів розробки моделі є вибір типу навчання (наприклад, з учителем, без учителя, або з підкріпленням) та архітектури нейромережі, яка найкраще підходить до поставленого завдання. Цей вибір ґрунтується не лише на аналітичному підході, а й на емпіричному досвіді з численних експериментів, що відображає загальний характер інженерії штучного інтелекту.

Концептуальні дослідження базуються на програмній реалізації розпізнавання образів, з метою визначення станів електричних двигунів під час функціонування та їх класифікація. Такий підхід дає можливість визначення не лише справності силової установки, а й аналізу функціонування силової установки безпілотного літального апарату у різних режимах польоту, що дасть можливість у тому числі ідентифікувати нетипові стани силової установки та запобігти виходу її з ладу, та вибрати оптимальний режим польоту безпілотного літального апарату.

Метою статті є аналіз підходів щодо діагностики функціонування силової установки під польоту безпілотних літальних апаратів на базі електроприводу.

Штучні нейронні мережі (ШНМ) являють собою ефективний інструмент обробки даних, що дозволяє знаходити складні, нелінійні залежності між вхідними ознаками та цільовими значеннями (мітками). В основі архітектури ШНМ лежить ієрархічна структура, яка моделюється у вигляді направленного ациклічного графа, де вузли відповідають за математичні операції, а ребра — за передачу тензорів між ними.

Базовими елементами такої архітектури є шари нейронів, серед яких виділяють вхідний, один або декілька прихованих шарів та вихідний. Приховані шари містять набір нейронів, що здійснюють афінні перетворення вхідних сигналів з подальшим застосуванням нелінійних активаційних функцій. Саме ці шари і є джерелом здатності мережі моделювати складні функціональні залежності.

Однією з найбільш поширених реалізацій є щільна (dense) або повнозв'язна (fully connected) нейронна мережа, в якій кожен нейрон певного шару має з'єднання з усіма нейронами попереднього шару. Така повна зв'язність дозволяє максимально використати інформацію з попередніх рівнів обробки та забезпечує гнучкість моделі у контексті навчання.

Застосування повнозв'язних мереж доцільне для задач, у яких структурна залежність вхідних даних є неочевидною або розподіленою по багатьох вимірах. Це, зокрема, задачі класифікації, регресійного аналізу, обробки сигналів та передбачення часових рядів.

Графічне представлення такої мережі відповідає концепції послідовної трансформації вхідного вектору через низку нелінійних функцій, кожна з яких виконується шаром нейронів. Кількість нейронів у кожному шарі та кількість прихованих шарів визначається шляхом емпіричного налаштування або гіперпараметричної оптимізації з урахуванням складності задачі [8-9].

Коли модель навчена і подає немаркований приклад, вона дає три прогнози: вірогідність того, що визначена форма та її колір – це рівень стану силової установки. Для цього прикладу сума вихідних прогнозів становить 1,0. В реалії цей прогноз розбивається на такі складові у випадку попереднього діагностичного контролю силової установки: 0,03 для умовно робочого стану придатного до функціонування у різних режимах польоту, 0,95 для повної справності силової установки та 0,02 для критичних режимів функціонування силової установки. Це означає, що модель передбачає, з 95% ймовірністю стану силової установки.

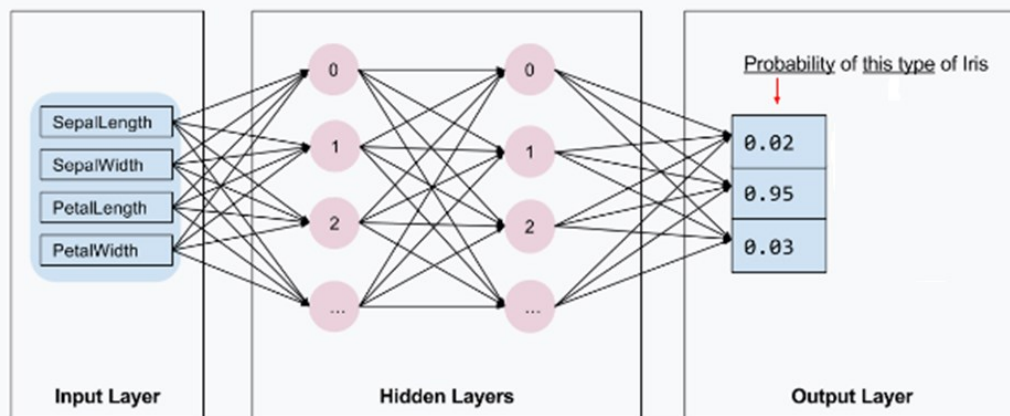


Рис.1. Нейронна мережа

API TensorFlow tf.keras – це найкращий спосіб створити моделі та шари. Це дозволяє легко створювати моделі та експериментувати, а Keras обробляє складність об'єднання всього. Tf.keras.Sequential модель – лінійний стек шару. Його конструктор приймає список шарів-екземплярів, в цьому випадку два щільні шари з кожними 10 вузлами і вихідний шар з 3 вузлами, що представляють передбачувані мітки. Параметр input_shape першого шару відповідає кількості функцій з набору даних і є обов'язковим.

Функція активації визначає вихід одного нейрона на наступний шар. Це вільно залежить від того, як пов'язані нейронні слої. Є багато доступних активацій, але ReLU є загальним для прихованих шарів [5].

Підбір оптимальної архітектури штучної нейронної мережі, включаючи кількість прихованих шарів та кількість нейронів у кожному з них, є критично важливим аспектом моделювання в задачах глибинного навчання. Ідеальна конфігурація значною мірою залежить від характеру проблеми, складності вхідних даних та кількості доступної навчальної інформації. Відомо, що збільшення кількості прихованих шарів здатне підвищити виражальну здатність моделі. Проте це також збільшує ймовірність перенавчання (overfitting), особливо при обмеженому обсязі навчальної вибірки.

Перенавчання – це ефект, коли модель занадто точно відтворює закономірності навчального набору, втрачаючи здатність узагальнювати на нові, раніше не бачені дані. У результаті, така модель демонструє високу точність на тренувальному наборі, проте її ефективність різко знижується на тестовому наборі або в реальних умовах експлуатації.

Навчання нейронної мережі передбачає поступову оптимізацію вагових коефіцієнтів з метою зведення до мінімуму функції втрат. Цей процес реалізується через метод зворотного поширення помилки з використанням стохастичного градієнтного спуску або його модифікацій (наприклад, Adam, RMSProp).

У випадку задач класифікації (зокрема, класифікації станів силових установок), застосовується наглядове навчання (supervised learning). У цьому випадку модель навчається на основі прикладів, що містять відомі мітки класів. На відміну від цього, у безнаглядному навчанні (unsupervised learning), яке також активно застосовується в сучасних дослідженнях, мітки відсутні, і завдання моделі полягає у виявленні внутрішніх закономірностей чи кластерів у наборі вхідних даних.

Під час навчання необхідно обчислювати функцію втрат (loss function) – кількісну міру відхилення прогнозованих значень від реальних міток. У TensorFlow для задач багатокласової класифікації з категоріальними мітками часто застосовується функція:

`tf.keras.losses.SparseCategoricalCrossentropy()`

Ця функція оцінює відмінність між імовірнісним розподілом, отриманим на виході мережі (зазвичай з використанням `softmax`), та фактичною категоріальною міткою у вигляді цілого числа.

Модель обчислює свою втрату за допомогою функції `tf.losses.sparse_softmax_cross_entropy`, яка приймає передбачення моделі та потрібну мітку. Значення повернутої втрати поступово збільшується, оскільки прогноз погіршується.

TensorFlow має багато алгоритмів оптимізації, доступних для навчання. Ця модель використовує `tf.train.GradientDescentOptimizer`, який реалізує алгоритм стохастичного градієнтного спуску (SGD). `Learning_rate` визначає розмір кроку для кожної ітерації вниз по пагорбу. Це гіперпараметр, який зазвичай коригується, щоб досягти кращих результатів.

```
>>> train_loss_results = []
>>> train_accuracy_results = []
>>> num_epochs = 201
>>> for epoch in range(num_epochs):
>>>     epoch_loss_avg = tfe.metrics.Mean()
>>>     epoch_accuracy = tfe.metrics.Accuracy()
>>>     for x, y in train_dataset:
>>>         grads = grad(model, x, y)
>>>         optimizer.apply_gradients(zip(grads, model.variables), global_step=tf.train.get_or_create_global_step())

>>> epoch_loss_avg(loss(model, x, y))
<tf.Tensor: id=73432, shape=(), dtype=float32, numpy=0.18478529>
>>> epoch_accuracy(tf.argmax(model(x), axis=1, output_type=tf.int32), y)
<tf.Tensor: id=73449, shape=(24,), dtype=int32, numpy=
array([2, 2, 2, 2, 0, 0, 1, 1, 2, 2, 0, 1, 1, 2, 2, 2, 0, 1, 0, 0, 1,
       0, 2])>, <tf.Tensor: id=73286, shape=(24,), dtype=int32, numpy=
array([2, 2, 1, 2, 1, 0, 0, 1, 1, 2, 2, 0, 1, 1, 2, 2, 0, 1, 0, 0, 1,
       0, 2])>>
>>> train_loss_results.append(epoch_loss_avg.result())
>>> train_accuracy_results.append(epoch_accuracy.result())
>>> if epoch % 50 == 0:
>>>     print("Epoch {:03d}: Loss: {:.3f}, Accuracy: {:.3%}".format(epoch, epoch_loss_avg.result(), epoch_accuracy.result()))
Epoch 200: Loss: 0.051, Accuracy: 98.333%
```

Рис.2. Тренування мережі

Цикл навчання надає приклади у модель дані, щоб допомогти зробити прогнози кращими.

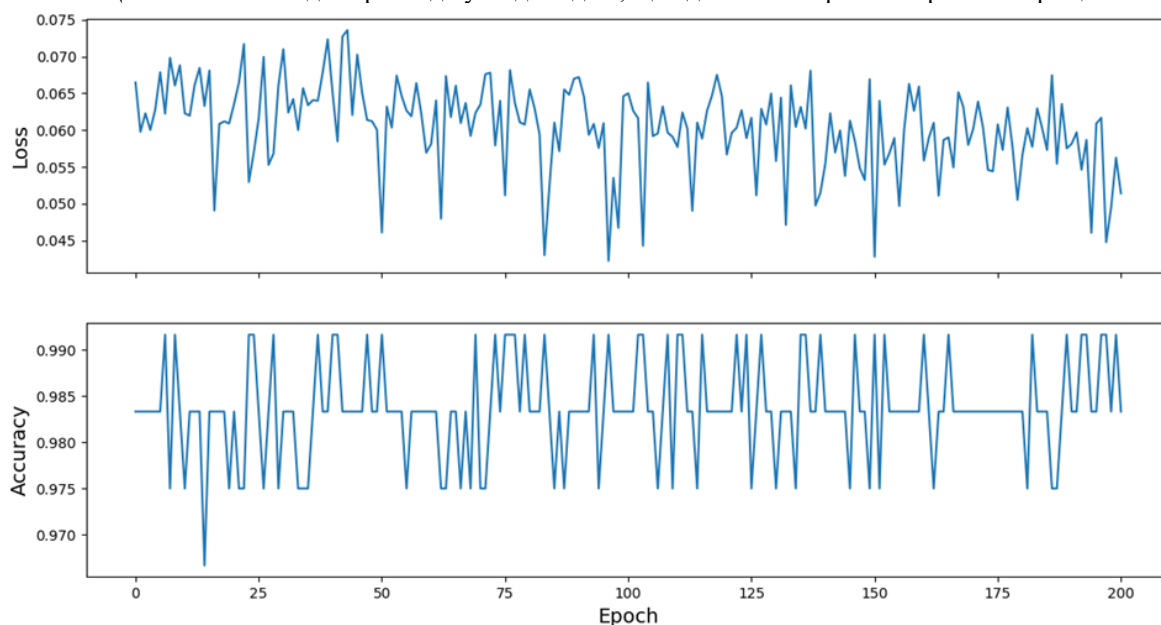


Рис.3. Графічні дані результату

Налаштування відповідного тесту Dataset здійснюється подібно для даних навчання, поданих на рисунку 4.

Після завершення етапу тренування (навчання), модель машинного навчання повинна пройти фазу оцінювання, в якій перевіряється її здатність узагальнювати знання на нових, не бачених раніше даних. На відміну від фази навчання, де модель проходить через декілька епох (ітерацій) на навчальній вибірці – тобто обробляє кожен приклад з тестового набору без коригування своїх параметрів.

```
>>> test_url = "http://download.tensorflow.org/data/iris_test.csv"
>>> test_fp = tf.keras.utils.get_file(fname=os.path.basename(test_url), origin=test_url)
Downloading data from http://download.tensorflow.org/data/iris_test.csv
8192/573 [=====] - 0s 0us/step
>>> test_dataset = tf.data.TextLineDataset(test_fp)
>>> test_dataset = test_dataset.skip(1)
>>> test_dataset = test_dataset.map(parse_csv)
>>> test_dataset = test_dataset.shuffle(1000)
>>> test_dataset = test_dataset.batch(32)
```

Рис.4. Приклад тесту Dataset для налаштування

Це дозволяє не лише зафіксувати стабільність та якість узагальнення, а й виміряти ключові показники ефективності моделі, зокрема точність (accuracy), повноту (recall), F1-метрику, втрати (loss), і т. ін., залежно від типу завдання.

```
>>> predict_dataset = tf.convert_to_tensor([
    [5.1, 3.3, 1.7, 0.5,],
    [5.9, 3.0, 4.2, 1.5,],
    [6.9, 3.1, 5.4, 2.1]
])
>>> predictions = model(predict_dataset)
>>> for i, logits in enumerate(predictions):
    class_idx = tf.argmax(logits).numpy()
    name = class_ids[class_idx]
    print("Example {} prediction: {}".format(i, name))

Example 0 prediction: Iris setosa
Example 1 prediction: Iris versicolor
Example 2 prediction: Iris virginica
```

Рис.5. Приклад тестування

Таким чином, початковим етапом розпізнавання станів силової установки безпілотного літального апарату є його визначення та класифікація з метою ідентифікації та подальшого визначення значення для подальшого вибору оптимального режиму польоту з метою виконання завдання.

Запропонований варіант програмної реалізації інтелектуальної системи та концепція побудови принципів визначення та ідентифікації несправностей силової установки безпілотного літального апарату.

Висновки

1. У роботі розглянуто концептуальну можливість використання сучасних програмних продуктів у контексті реалізації функціонування безпілотних літальних апаратів в умовах обмеженої транспортної та інформаційної інфраструктури спрямованої на оперативне реагування вихід із ладу силової установки.

2. Розглянуто приклад класифікації сигналів діагностичної системи на прикладі трьох станів функціонування електроприводів силової установки при використанні програмного продукту TensorFlow. Визначення оптимальної архітектури нейронної мережі є критично важливим завданням при її застосуванні до конкретної прикладної задачі. Як і у більшості процесів машинного навчання, вибір кількості прихованих шарів та кількості нейронів у кожному з них не має універсального рецепта.

3. У контексті розпізнавання технічного стану силових електродвигунів або інших агрегатів енергетичного обладнання, основна концепція зводиться до отримання вектору ймовірностей, які характеризують відповідність поточних вхідних даних, певному стану або аномалії.

Література

1. Григоров О.М. Міжнародно-правові засади регулювання відносин у сфері цивільної авіації: становлення та розвиток: Монографія. Одеса: Фенікс. 2020. С.280–281.
2. Харченко В.П. Авіоніка безпілотних літальних апаратів / В.П. Харченко, В.І. Чепіженко, А.А. Тунік, С.В. Павлова/; за ред. В.П. Харченка. К.: ТОВ «Абрис-принт», 2012. 464с.
3. T. Shmelova, S. Boiko, O. Kotov, O. Burlaka, M. Nozhnova, Yu. Bershadska, L. Chyzhova, D. Hinosian, V. Zhurid, V. Yemets, Yu. Oliinyk, B. Moskaluk Modern aspects of application and development of Unmanned Aerial Vehicles. Warsaw: iScience Sp. z.o.o. – 2021. – 139 p.
4. Handmann, U., Kalinke, T., Tzomakas, C., Werner, M., & von Seelen, W. Computer vision for driver assistance systems. In Proceedings of SPIE - The International Society for Optical Engineering, 3364, 136-147 (1999). <https://doi.org/10.1117/12.317463>
5. Lee, H., Kim, H., & Choi, S. Driving skill modeling using neural networks for performance-based haptic assistance. IEEE Transactions on Human-Machine Systems, 51(3), 198-210 (2021). <https://doi.org/10.1109/THMS.2021.3061409>
6. Abadi, M., et al. (2016). TensorFlow: A System for Large-Scale Machine Learning. OSDI.

7. Martín Abadi et al., Google Research White Paper, TensorFlow, 2015.
- Kalra, N., & Paddock, S. M. How many miles of driving would it take to demonstrate autonomous vehicle reliability? RAND Corporation (2016). <https://doi.org/10.7249/RR1478>
8. Goodfellow, I., Bengio, Y., Courville, A. *Deep Learning*. MIT Press, 2016.
9. Geron, A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, O'Reilly, 2022.
10. Chengula, T. J., Mwakalonge, J., Comert, G., Sulle, M., & Siuhi, S. Enhancing advanced driver assistance systems through explainable artificial intelligence for driver anomaly detection. *Machine Learning with Applications*, 17, Article 100580 (2024). <https://doi.org/10.1016/j.mlwa.2024.100580>
11. Ren, K., Wang, Q., Wang, C., Qin, Z., & Lin, X. The security of autonomous driving: Threats, defenses, and future directions. *IEEE Transactions on Vehicular Technology*, 69(8), 8743-8756 (2020). <https://doi.org/10.1109/JPROC.2019.2948775>
12. Wang, J., Zhang, L., Huang, Y., & Zhao, J. Safety of autonomous vehicles. *Journal of Advanced Transportation*, 2020, Article ID 8867757. <https://doi.org/10.1155/2020/8867757>
13. Zhang, B., Chen, W., Xu, C., Qiu, J., & Chen, S. Autonomous vehicles traversability mapping fusing semantic–geometric in off-road navigation. *Drones*, 8(496), (2024). <https://doi.org/10.3390/drones8090496>
14. Fernandes, N. O., Thürer, M., Rodrigues, F., Ferreira, L. P., Silva, F. J., & Avila, P. Worker assignment in dual resource constrained systems subject to machine failures: a simulation study. *International Journal of Industrial Engineering and Management*, 13(2), 110-118 (2022). <https://doi.org/10.24867/IJIE-2022-2-305>
15. Kupin, A., Zubov, D. A., Qureshi, M. S., Köse, U. Prototyping smart home for immobilized people: EEG/MQTT-based brain-to-thing communication No. 2: Radio Electronics, Computer Science, Control. P.90-100 (2022). <https://doi.org/10.15588/1607-3274-2022-2-9>