AI for SDGs

A TECHNICAL AND ILLUSTRATED TOUR

Laure Berti-Équille

edp sciences

AI for SDGs

A TECHNICAL AND ILLUSTRATED TOUR

Practical frameworks, real-world cases, and visual insights on harnessing AI to advance the UN 2030 Agenda

Laure Berti-Équille

How can Artificial Intelligence (AI) be effectively leveraged with the United Nations Sustainable Development Goals (SDGs)? Beyond the hype, AI has the potential to transform global challenges into opportunities—if applied responsibly and inclusively. Yet, connecting complex technical systems to urgent sustainability issues requires clarity, methodological rigor, and illustrative evidence.

This book offers a structured and visually engaging exploration of how AI can support each of the 17 SDGs. By combining technical depth with accessible illustrations, it bridges the gap between advanced AI concepts and their practical applications in domains such as poverty estimation, climate action, health, and education. Readers will encounter real-world case studies, annotated diagrams, and examples that highlight both the promises and the limitations of AI for sustainability.

Designed as both a reference and a guide, the book speaks to researchers, practitioners, policymakers, and students who want to understand not only the « what » and « why » but also the « how » of AI for sustainable development. By the end of this illustrated tour, readers will gain a clearer vision of where AI truly contributes, where caution is needed, and how innovation can be directed to serve the common good.

The author, Laure Berti-Equille is a Research Director in applied Data Science and AI at IRD, the French Research Institute for Sustainable Developement. She has extensive research and teaching experience with numerous published papers and student supervisions. This book and related online material are open-access.

ISBN : 978-7598-3883-7

Institut de Recherche pour le Développement FRANCE

edp sciences

AI FOR SDGs
A TECHNICAL & ILLUSTRATED TOUR

Laure Berti-Équille

edp sciences

Illustrator and cover designer: Claire Martha

ISBN : 978-7598-3883-7 (version papier) et 978-7598-3884-4 (version ebook)
DOI: https://doi.org/10.1051/978-2-7598-3883-7

AI FOR SDGs
A TECHNICAL & ILLUSTRATED TOUR

Laure Berti-Équille

CONTENTS

PREFACE

The world has already passed critical tipping points. Despite decades of progress, poverty, hunger, inequality, climate change, and environmental degradation continue to threaten global stability and human well-being. The United Nations' Sustainable Development Goals (SDGs) offer a bold blueprint to tackle these challenges by 2030—but time is running out. To meet these ambitious targets, we must turn to the most powerful tools available today.

Artificial Intelligence (AI) and deep learning can help us collect better data, make faster decisions, personalize interventions, and scale solutions in ways never before possible. But unlocking this potential requires clarity, rigor, and a deep understanding of how AI models can be aligned with human needs. Artificial Intelligence can be a powerful engine for solving today's most urgent challenges. Across every continent, AI models are being used to address the SDGs— from predicting poverty with satellite images, to fighting climate change, protecting biodiversity, and improving access to education and healthcare.

This book is a practical and engaging journey through some of the most impactful applications of deep learning and AI architectures in service of the SDGs. Whether you are a researcher, a student, or a decision-maker curious about how machine learning can serve people and the planet, this book will show you some real use cases, with concisely explained methods.

Each chapter includes:
- A technical breakdown of the method used, written clearly and concisely;
- An illustration of the architecture (such as CNN-LSTM, GAN, GNN, etc.);
- A balanced view of the pros and cons of the approach in that specific context;
- The state-of-the-art references, with DOI links for deeper reading;
- A link to a curated Awesome List of resources such as surveys, relevant papers, datasets, codes, benchmarks, and educational content to help you go further.

We've structured this book to be as hands-on and approachable as possible with real examples of AI solving real problems. If you care about the planet, people, and progress, and want to see how machine learning is being used for good, this book is for you.

Yet much remains to be done; advancing the SDGs demands a new generation of AI solutions that are not only powerful, but also ethical, transparent, frugal, resource-efficient, and deeply aligned with the needs of people and planet.

SDG#1 POVERTY ESTIMATION FROM SATELLITE IMAGES WITH TRANSFORMERS

Around 712 million people – 8.5% of the global population– live today on less than $2.15 per day, the extreme poverty line relevant for low-income countries. Three-quarters of all people in extreme poverty live in Sub-Saharan Africa or fragile and conflict-affected countries.

Estimating poverty with AI is especially useful in regions where ground data is scarce, outdated, or difficult to collect. AI models can infer poverty indicators from alternative data sources like satellite imagery, mobile phone usage, or night-time lights, offering timely and granular insights.

Source (Retrieved on March 25th, 2025): https://www.worldbank.org/en/topic/poverty/overview

SDG#1 POVERTY ESTIMATION FROM SATELLITE IMAGES WITH TRANSFORMERS

Transformers offer parallel computation, capture long-range dependencies, and enable contextual understanding.

Complexity, interpretability, and the resource-intensive nature of transformers pose challenges for real-world scenarios.

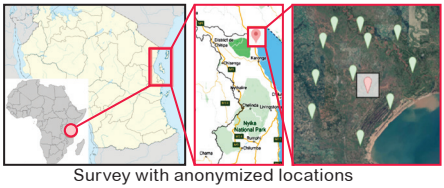
Transformers can analyze complex data patterns and can map socioeconomic indicators with satellite image features, enabling accurate poverty estimation. They utilize self-attention mechanisms to weigh input relevance and process entire sequences of diverse data sources simultaneously, enhancing parallelization.

METHOD

A transformer architecture can predict poverty estimates from satellite images by leveraging its ability to capture spatial patterns and contextual relationships within data. Through self-attention mechanisms, transformers encode and decode pixel-level features and their interactions across the image, allowing them to identify key indicators of poverty such as infrastructure, land use, and settlement patterns. By training on labeled datasets to learn the mapping between the ground truth poverty data as socioeconomic indicators surveyed on the ground and the satellite images, transformers can learn to correlate specific image features with poverty levels, enabling accurate predictions across different geographical regions. The architecture consists of encoder and decoder blocks with multi-head attention and feed-forward layers. Positional encoding is added to retain sequence order information. Their ability to model long-range dependencies has led to applications beyond NLP and vision and offer a scalable and effective solution for poverty estimation, facilitating targeted interventions and resource allocation in areas of need. However, challenges related to sparse data labeling, spatial uncertainty due to the anonymization of the poverty surveys, interpretability, and computational cost need to be addressed to ensure the applicability of transformer-based poverty prediction systems in real-world scenarios.

1. INPUT

Economic Wealth Indicators from surveys

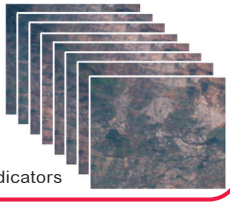


Survey with anonymized locations

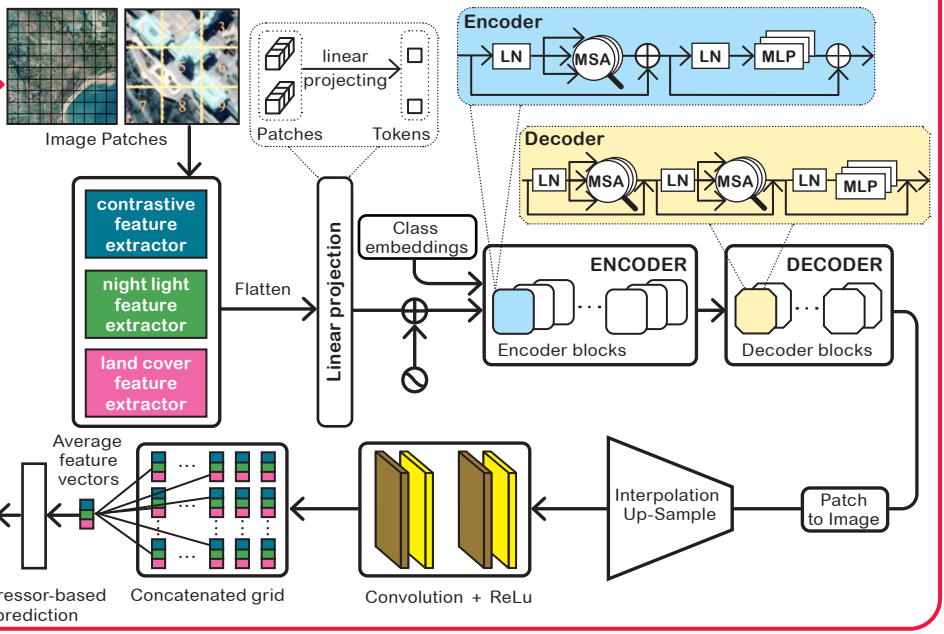
Latitude	Longitude	Consumption expenditures ($/day)	Name of the village
-9.7298955	33.859230	1.8056488929595411	Kaporo
-9.715316	33.88666	1.4157905044417405	Karonga district
-11.95341	33.367064	1.957231017237896	Mzimba disctrict

Ground truth values of poverty indicators

Satellite Images

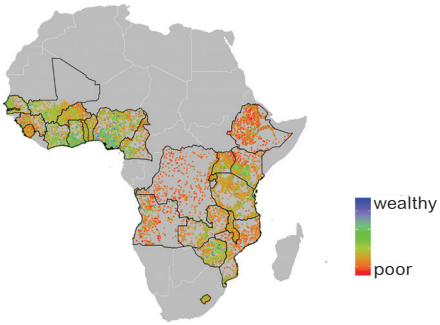


2. ARCHITECTURE



3. OUTPUT

Poverty value prediction



LEGEND

LN Layer Norm

MLP Multilayer Perceptron

MSA Multihead Self-Attention

⊕ Element-wise Addition

⊘ Position Embedding

FURTHER READING

O. Hall, et al., « A Review of Explainable AI in the Satellite Data, Deep machine Learning, and Human Poverty Domain », Patterns, vol. 3, Issue 10, no. 100600, 2022.
https://doi.org/10.1016/j.patter.2022.100600

R. Jarry, M. Chaumont, L. Berti-Equille, and G. Subsol, « Comparing Spatial and Spatio-Temporal Paradigms to Estimate the Evolution of Socio-Economic Indicators from Satellite Images », In 2023 IEEE Int. Geosci. Remote Sens. Symp., pp. 5790-5793, Jul. 2023.
https://doi.org/10.1109/IGARSS52108.2023.10282306

M. Kakooei and A. Daoud, « Increasing the Confidence of Predictive Uncertainty: Earth Observations and Deep Learning for Poverty Estimation », IEEE Transactions on Geoscience and Remote Sensing, vol. 62, pp. 1-13, no. 470461, 2024.
https://doi.org/10.1109/TGRS.2024.3392605

C. Yeh, et al., « SustainBench: Benchmarks for Monitoring the Sustainable Development Goals with Machine Learning », In 2021 NeurIPS International Conference, Datasets and Benchmarks Track, Dec. 2021.
https://openreview.net/forum?id=5HR3vCylqD

C. Yeh, et al., « Using Publicly Available Satellite Imagery and Deep Learning to Understand Economic Well-Being in Africa », Nat. Commun., vol. 11, no. 2583, 2020.
https://doi.org/10.1038/s41467-020-16185-w

Y. Yuan, et al., « SITS-Former: A Pre-trained Spatio-spectral-temporal Representation Model for Sentinel-2 Time Series Classification », Int. J. of Applied Earth Observation and Geoinformation, no. 106, 2022.
https://doi.org/10.1016/j.jag.2021.102651

Scan the QR code to access state-of-the-art research papers, datasets, codes, benchmarks, real-world use cases, and educational materials.

SDG#2 CROP DISEASE DETECTION WITH PRETRAINED NETWORKS AND ENSEMBLE LEARNING

733 million people globally suffered from malnutrition in 2023, an increase of 152 million since 2019. An estimated 28.9 % of the global population – 2.33 billion people – were moderately or severely food insecure.

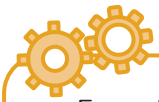
Automated early detection of diseases that can affect crops and livestock would cut costs for cultivators and farmers and help prevent major losses and low yield, impacting food security.

Source (Retrieved on March 25th, 2025): https://sdgs.un.org/goals/goal2#progress_and_info

SDG#2 CROP DISEASE DETECTION WITH PRETRAINED NETWORKS AND ENSEMBLE LEARNING

Pretrained networks facilitate the training of very deep networks by mitigating vanishing gradients.

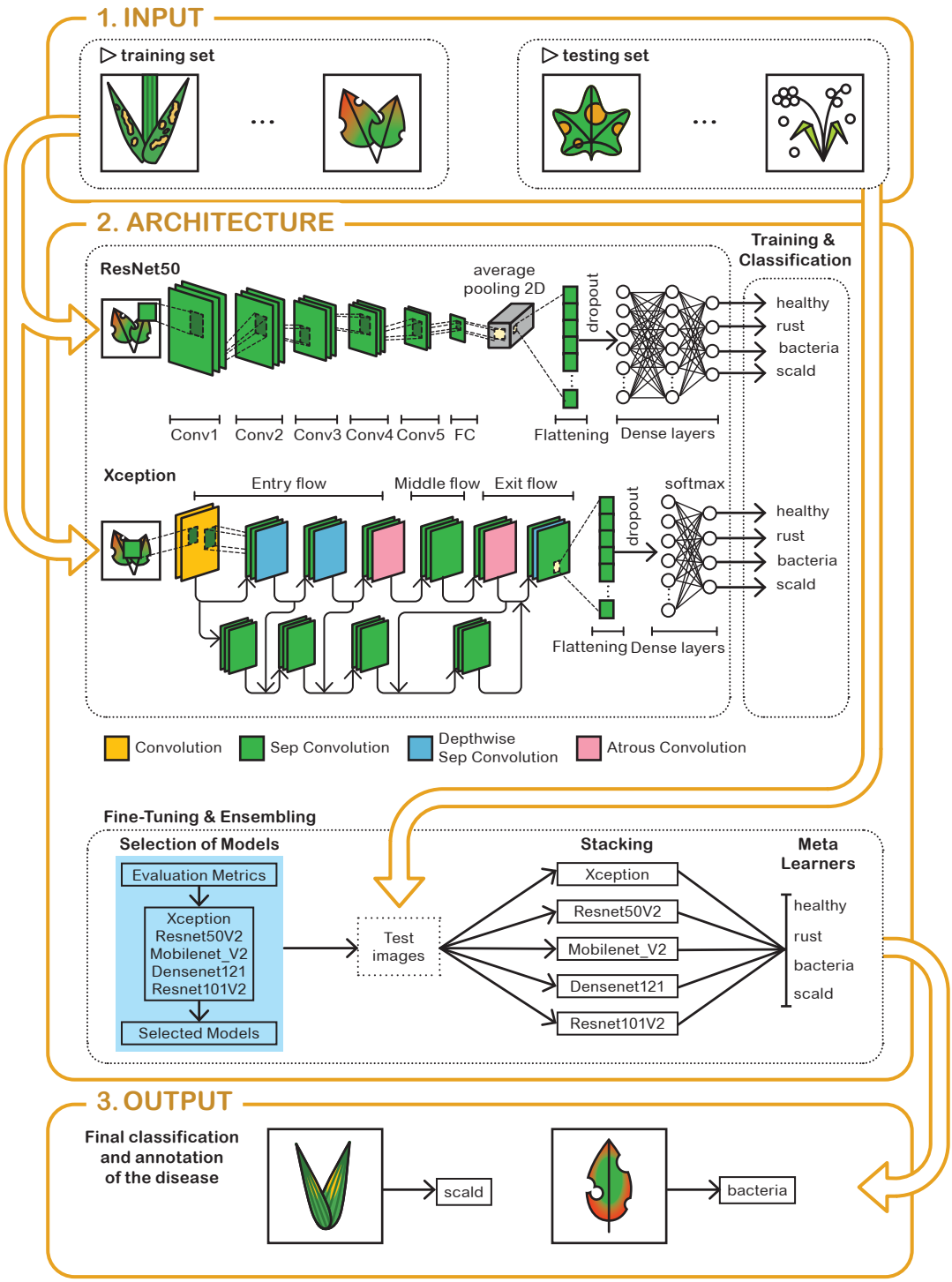
Combining multiple models improves predictive performance but increases complexity and training and inference times.



Ensemble learning such as bagging, boosting, and stacking allows the integration of diverse models to enhance generalization, effectively reducing variance and bias. ResNet introduces residual connections, allowing identity mappings. Xception utilizes depthwise separable convolutions for efficient computation. Both architectures have been influential in advancing deep learning and are widely used in image recognition tasks.

METHODS

ResNet addresses the degradation problem in deep networks by introducing residual learning. It uses shortcut connections to skip one or more layers, allowing gradients to flow directly through these connections during backpropagation. This approach enables the training of very deep networks with hundreds of layers. Xception builds upon the Inception architecture by replacing standard convolutions with depthwise separable convolutions. This factorizes convolutions into a depthwise convolution followed by a pointwise convolution, reducing computational cost while maintaining performance. Both architectures have set benchmarks in image classification tasks. Their designs have influenced subsequent neural network architectures. Ensemble learning combines predictions from these models to produce a more robust and accurate output. Bagging involves training multiple models independently on random subsets of data and averaging their predictions, reducing variance. Boosting sequentially trains models, each correcting errors of its predecessor, aiming to reduce bias. Stacking combines outputs of several models using a meta-model to improve predictive accuracy. This approach leverages the strengths of diverse models, mitigating individual weaknesses. Understanding ensemble strategies is valuable for building robust models that are carefully tuned and validated to avoid overfitting.


1. INPUT
training set
testing set
2. ARCHITECTURE
Training & Classification
ResNet50
average pooling 2D
dropout
healthy
rust
bacteria
scald
Conv1
Conv2
Conv3
Conv4
Conv5
FC
Flattening
Dense layers
Xception
Entry flow
Middle flow
Exit flow
softmax
Convolution
Sep Convolution
Depthwise Sep Convolution
Atrous Convolution
Fine-Tuning & Ensembling
Selection of Models
Evaluation Metrics
Xception
Resnet50V2
Mobilenet_V2
Densenet121
Resnet101V2
Selected Models
Test images
Stacking
Meta Learners
3. OUTPUT
Final classification and annotation of the disease
scald
bacteria

FURTHER READING

R. Kumar, et al., « Hybrid Approach of Cotton Disease Detection for Enhanced Crop Health and Yield », IEEE Access, vol. 12, pp. 132495-132507, 2024.
https://doi.org/10.1109/ACCESS.2024.3419906

L. Li, S. Zhang, and B. Wang, « Plant Disease Detection and Classification by Deep Learning—A Review », IEEE Access, vol. 9, pp. 56683-56698, 2021.
https://doi.org/10.1109/ACCESS.2021.3069646

A. Nader, M.H. Khafagy, and S.A. Hussien, « Grape Leaves Diseases Classification using Ensemble Learning and Transfer Learning », International Journal of Advanced Computer Science and Applications (IJACSA), vol. 13, no. 7, 2022.
http://dx.doi.org/10.14569/IJACSA.2022.0130767

H.N. Ngugi, et al., « Revolutionizing Crop Disease Detection with Computational Deep Learning: A Comprehensive Review », Environ. Monit. Assess., vol. 196, no. 302, 2024.
https://doi.org/10.1007/s10661-024-12454-z

R. Rashid, et al., « An Early and Smart Detection of Corn Plant Leaf Diseases Using IoT and Deep Learning Multi-Models », IEEE Access, vol. 12, pp. 23149-23162, 2024,
https://doi.org/10.1109/ACCESS.2024.3357099

K. Taji, et al., « An Ensemble Hybrid Framework: A Comparative Analysis of Metaheuristic Algorithms for Ensemble Hybrid CNN Features for Plants Disease Classification », IEEE Access, vol. 12, pp. 61886-61906, 2024,
https://doi.org/10.1109/ACCESS.2024.3389648

Scan the QR code to access state-of-the-art research papers, datasets, codes, benchmarks, real-world use cases, and educational materials.

SDG#3 AUGMENTATION OF HEALTHCARE DATA USING GAN

With the increased digitalization of health data and the market size of AI for healthcare expected to reach USD 45 billion by 2026, the role of synthetic data in the health information economy needs to be precisely delineated to develop fault-tolerant and patient-facing health systems.

In healthcare, patient privacy is governed by strict regulations like HIPAA and GDPR, which synthetic data helps to mitigate by mimicking real data without revealing personal identities. This is particularly helpful when real datasets are small, fragmented, or unavailable, such as in rare diseases or underrepresented populations.

Source (Retrieved on March 25th, 2025): https://www.marketsandmarkets.com/PressReleases/artificial-intelligence-healthcare.asp

SDG#3 AUGMENTATION OF HEALTHCARE DATA USING GAN

GANs can learn to mimic any data distribution effectively to generate a balanced, high-quality dataset

GANs require extensive training data, are computationally expensive, and may generate unrealistic samples.

Generative Adversarial Networks (GANs) can simulate realistic biomedical data and synthetic images, improving model training with data augmentation, preserving patient privacy, and thus enhancing disease diagnosis, treatment planning, and medical research. GANs consist of a generator and a discriminator for adversarial training. The generator creates data; the discriminator evaluates authenticity. Through competition, both networks improve.

METHOD

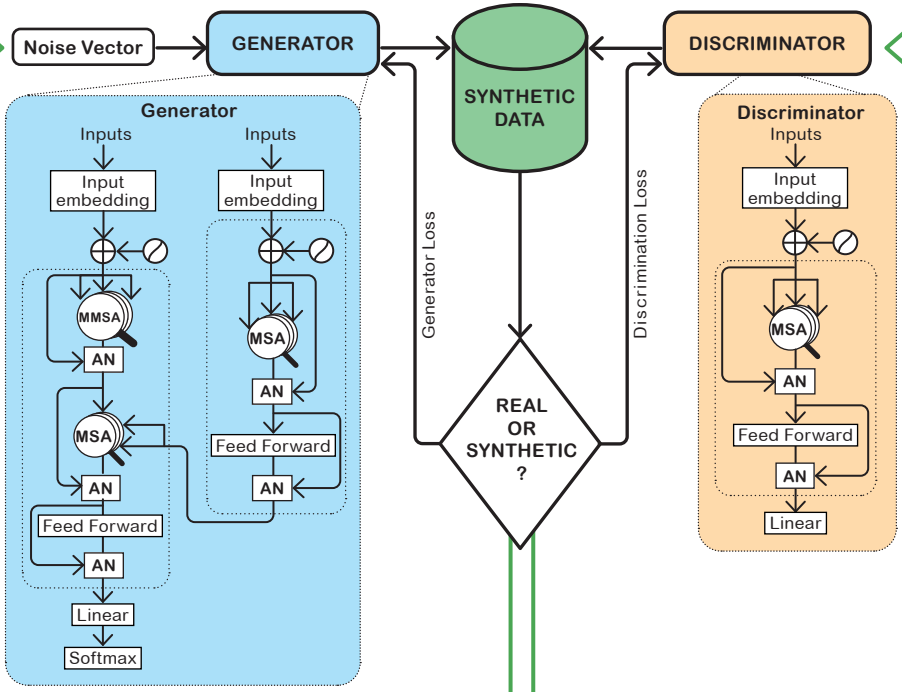
GANs consist of two neural networks: the generator and the discriminator. The generator aims to create realistic data samples, while the discriminator distinguishes between real and synthetic data. Over iterations, the generator learns to produce increasingly realistic outputs by minimizing the difference between its generated samples and real data. Trained with real data samples and those generated by the generator, the discriminator learns to differentiate. With opposing objectives, the generator seeks to minimize the log-probability that the discriminator correctly classifies synthetic data as synthetic, while the discriminator seeks to maximize this probability. Through iterative training, the generator improves its ability to generate realistic samples, while the discriminator becomes more adept at distinguishing between real and synthetic data. Ideally, this process leads to a state where the generator produces data that is indistinguishable from real data to augment the dataset to improve the performance of downstream tasks of classification or prediction. Applications include creating realistic images, enhancing image resolution, and data augmentation. Despite their potential, GANs are challenging to train due to issues like model collapse, where the generator produces limited varieties of data. Careful design of network architectures and training procedures is essential.

1. INPUT

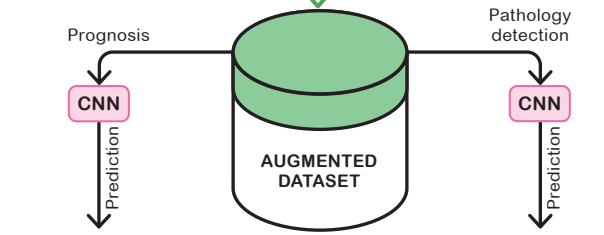
▷ Real patient data

X-ray, Blood reports, Medical reports, Medical test results, CT, MRI, Ultrasound output, ECG, EMG signals, Doctors and technicians' discussions and instructions

2. ARCHITECTURE



3. OUTPUT



LEGEND

FURTHER READING

R. J. Chen, et al., « Synthetic Data in Machine Learning for Medicine and Healthcare », Nat. Biomed. Eng., vol. 5, pp. 493-497, 2021.
https://doi.org/10.1038/s41551-021-00751-8

Y. Chen, et al., « Generative Adversarial Networks in Medical Image Augmentation: A Review », Computers in Biology and Medicine, 144, 2022.
https://doi.org/10.1016/j.compbiomed.2022.105382

K.K. Dixit, et al., « Data Augmentation with Generative Adversarial Networks for Deep Learning in Healthcare », In Proceedings of 2023 International Conference on Artificial Intelligence for Innovations in Healthcare Industries (ICAIIHI), Raipur, India, 2023, pp. 1-6.
https://doi.org/10.1109/ICAIIHI57871.2023.10489462

K. Rais, M. Amroune, and M.Y. Haouam, « Medical Image Generation Techniques for Data Augmentation: Disc-VAE versus GAN », In Proceedings of 2024 6th International Conference on Pattern Analysis and Intelligent Systems (PAIS), El Oued, Algeria, 2024, pp. 1-8.
https://doi.org/10.1109/PAIS62114.2024.10541221

A. Solanki and M. Naved (Eds), « GANs for Data Augmentation in Healthcare », Springer Cham Publisher, Nov. 2023.
https://doi.org/10.1007/978-3-031-43205-7

Z. Yang, Y. Li, and G. Zhou, « TS-GAN: Time-series GAN for Sensor-based Health Data Augmentation », ACM Trans. Comput. Healthcare, vol. 4, no. 2, Article 12, April 2023.
https://doi.org/10.1145/3583593

Scan the QR code to access state-of-the-art research papers, datasets, codes, benchmarks, real-world use cases, and educational materials.

SDG#3 PREDICTION OF AIR POLLUTION USING CNN-LSTM

Constant exposure to polluted air increases the risk of coronary and respiratory disease, stroke, diabetes and lung cancer. In 2017, air pollution was responsible for an estimated 5 million deaths globally, amounting to nearly 9% of the world's population.

With forecasting and predictive models, practitioners can better understand sources of pollution and provide warnings to the public ahead of peak pollution events.

Sources (Retrieved on March 25th, 2025): https://earth.org/10-facts-about-air-pollution/ https://ourworldindata.org/air-pollution

SDG#3 PREDICTION OF AIR POLLUTION USING CNN-LSTM

CNN-LSTM predicts accurately from multivariate time series due to its ability to capture spatial-temporal patterns efficiently.

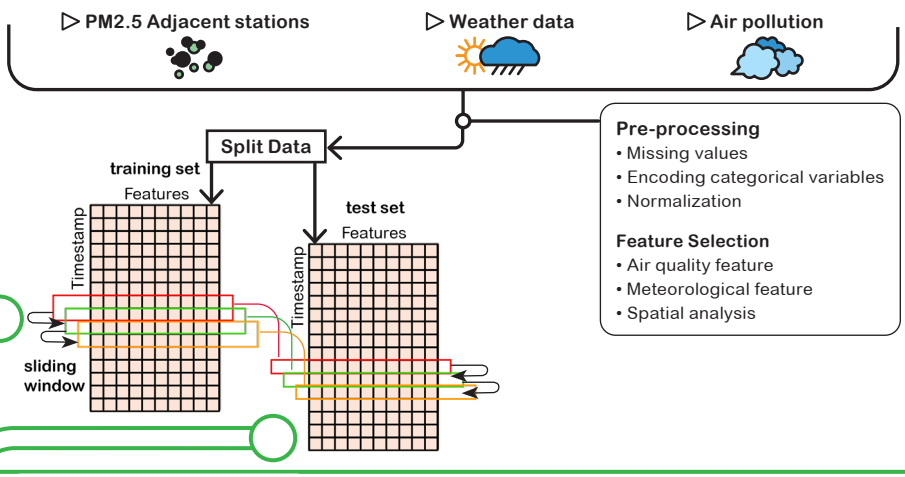
Tuning and training the CNN-LSTM is complex and computationally intensive. It is prone to overfitting and slow convergence.

A CNN-LSTM combines Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks. It offers a powerful framework for predicting air pollution from multivariate spatiotemporal time series. CNN handles spatial extraction; LSTM captures time. By leveraging both spatial and temporal information, it can capture intricate relationships between various weather parameters and past pollution levels.

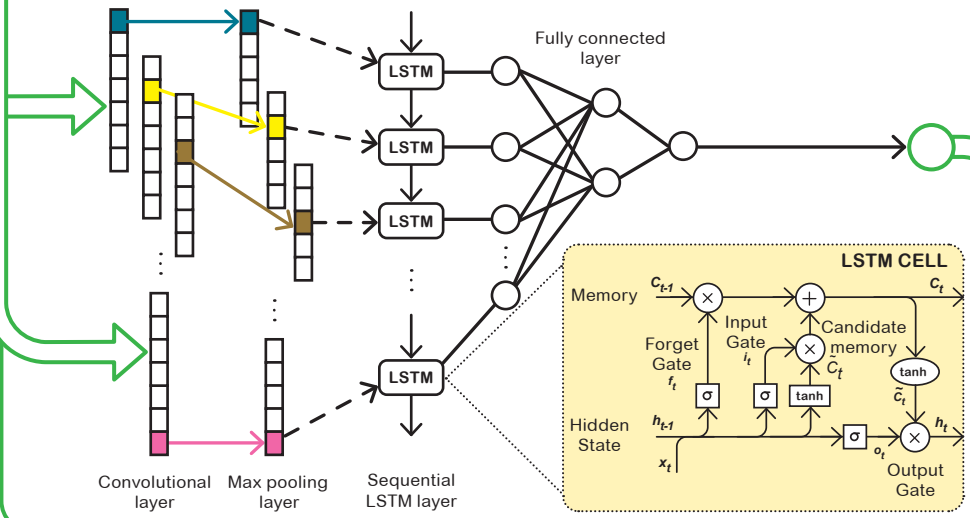
METHOD

A CNN-LSTM model combines the strengths of CNNs in capturing spatial patterns and LSTMs in modeling temporal dependencies. In predicting air pollution from multivariate spatiotemporal time series, such as weather data and previous PM2.5 concentrations, the model first processes input data through CNN layers. CNN layers analyze spatial relationships within data, extracting relevant features like temperature, humidity, and wind speed from weather data. Filters slide across the input grid, capturing patterns and creating feature maps. This process helps identify spatial correlations between different regions and weather variables. Next, each LSTM layer receives the CNN's output. LSTMs excel in capturing temporal dependencies by selectively retaining information over time through gates: forget, input, and output gates. This mechanism allows the model to remember long-term dependencies and ignore irrelevant information. The LSTM layer processes the sequential nature of time series data, such as historical PM2.5 concentrations. It learns patterns and trends in past pollution levels, capturing how they evolve. By incorporating previous PM2.5 concentrations, the model considers the pollutant's inertia and temporal dynamics. Temporal dependencies boost performance over static CNN and help with noisy or context-dependent sequences.

1. INPUT

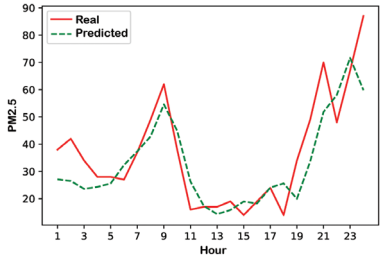


2. ARCHITECTURE



3. OUTPUT

Prediction of PM2.5 concentrations in time and space (h+1)



FURTHER READING

X. Bai, et al., « Prediction of PM2.5 Concentration Based on a CNN-LSTM Neural Network Algorithm », Peer J., vol. 12, no. e17811, Aug. 2024. https://doi.org/10.7717/peerj.17811

J. Duan, Y. Gong, J. Luo, et al. « Air-Quality Prediction Based on the ARIMA-CNN-LSTM Combination Model Optimized by Dung Beetle Optimizer », Sci. Rep. 13, 12127, 2023. https://doi.org/10.1038/s41598-023-36620-4

L. Jovova and K. Trivodaliev, « Air Pollution Forecasting Using CNN-LSTM Deep Learning Model », Proceedings of the 44th International Convention on Information, Communication and Electronic Technology (MIPRO), Opatija, Croatia, pp. 1091-1096, 2021. https://doi.org/10.23919/MIPRO52101.2021.9596860

J. Wang, et al., « An Air Quality Index Prediction Model Based on CNN-ILSTM », Sci. Rep., vol. 12, no. 8373, 2022. https://doi.org/10.1038/s41598-022-12355-6

Q. Zhang, et al., « Deep-AIR: A Hybrid CNN-LSTM Framework for Fine-Grained Air Pollution Forecast », arXiv: 2002.22957, 2020. http://arxiv.org/pdf/2001.11957.pdf

X. Zhu, et al., « Enhancing Air Quality Prediction with an Adaptive PSO-Optimized CNN-Bi-LSTM Model », Applied Sciences, vol. 14, issue 13, no. 5787, 2024. https://doi.org/10.3390/app14135787

Scan the QR code to access state-of-the-art research papers, datasets, codes, benchmarks, real-world use cases, and educational materials.

SDG#4 PERSONALIZATION OF A CONVERSATIONAL TUTORING SYSTEM WITH LLM

Only 58% of students worldwide achieved at least the minimum proficiency level in reading at the end of primary schooling in 2019. A large share of countries is moving backwards in learning outcomes at the end of lower secondary school.

Developing personalized education tools with AI can help overcome shortages of qualified teachers and limited educational resources by delivering adaptive, self-paced learning. It promotes equitable access to quality education, empowering students in underserved areas to improve their skills and future opportunities.

Source (Retrieved on March 25th, 2025): https://sdgs.un.org/goals/goal4#progress_and_info

SDG#4 PERSONALIZATION OF A CONVERSATIONAL TUTORING SYSTEM WITH LLM

LLMs excel at zero-shot and few-shot learning across various application domains and tasks.

LLMs are resource-intensive and prone to hallucination and stereotype amplification due to bias and low-quality training data.

LLMs are massive transformer-based architectures with billions of parameters. They are pretrained on diverse, very large-scale corpora and fine-tuned for specific downstream tasks. They are capable of reasoning, generation, translation and can be used in chatbots, summarization, question-answering, which makes them adequate for backing novel solutions in Education.

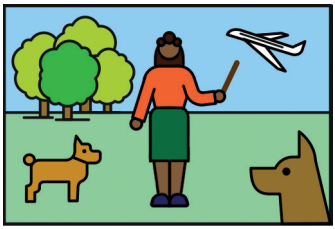
METHOD

A Large Language Model (LLM) is based on a stack of transformer blocks, each containing self-attention and feedforward layers. The model begins with an embedding layer that converts input tokens (words or subwords) into dense vector representations. These embeddings are combined with positional encodings to provide information about token order. Each transformer block applies multi-head self-attention to allow the model to weigh relationships between tokens. The attention mechanism computes attention scores that determine how much focus to give to other tokens. Then, the result is passed through a feedforward neural network with non-linear activations. Layer normalization and residual connections help stabilize training and allow deep architectures. The outputs from each transformer layer are passed on to the next block, building increasingly abstract representations. The final transformer layer produces context-aware embeddings for each token. These are passed to a linear projection layer followed by a softmax to produce a probability distribution over the vocabulary. The model is trained to predict the next token in a sequence (causal language modeling) or to fill in blanks (masked language modeling). Some LLMs use decoder-only models (e.g., GPT), while others use encoder-decoder formats (e.g., T5). To handle long texts, techniques like attention masking and sliding windows are used.

1. INPUT

▷ **A collection of pedagogical materials and a specific task**

Lesson: How to describe a picture



Scene: In a park

Person: One woman, two dogs

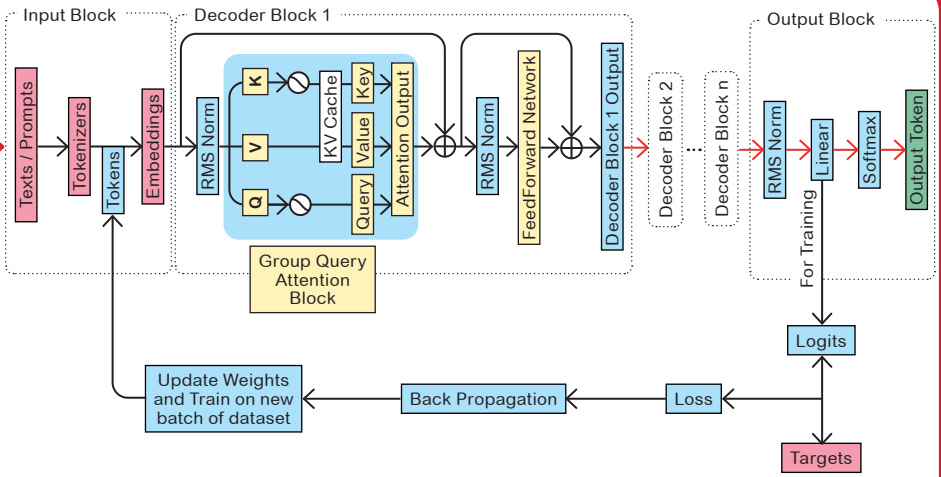
Objects: Wooden stick, tree, plane, etc.

Activities: A woman is carrying a wooden stick, playing with her dog in a park.

▷ **A series of text prompts**

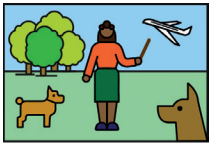
[Role & Task Definition] You are a primary school language teacher. You teach the student to describe the picture.
[Pedagogical Instruction] You are using the knowledge construction approach to help me describe the picture […]
[Behavior Constraint] Ask me only one question at a time. […]

2. ARCHITECTURE



3. OUTPUT

▷ **Conversations and learning experiences through interactions based on natural language**



Scene : In a park

Person : One woman, two dogs

Objects : Wooden stick, tree, plane, etc.

Activities : A woman is carrying a wooden stick, playing with her dog in a park.

Low Consciousness Low Extraversion

- Let's start by looking at the picture. Can you tell me, what do you see in the sky?
- … (looks away)
- That's okay. Take your time. Can you see what the woman in the picture is doing?
- She is … she is standing …

High Consciousness High Extraversion

- Let's start by looking at the picture. Can you tell me, what do you see in the park?
- Oh there's a woman. She is playing with dogs.
- Well done! Can you describe how the dogs are reacting?
- They are happy, dogs love wooden sticks.

LEGEND

→ Training → Inference ⊕ Element-wise Addition ⊘ Rotary Positional Encoding

FURTHER READING

Z. Cai, et al., « Advancing Knowledge Together: Integrating Large Language Model-based Conversational AI in Small Group Collaborative Learning », In Extended Abstracts of the CHI Conference on Human Factors in Computing Systems (CHI EA '24), New York, NY, USA, no. 37, pp. 1-9.
https://doi.org/10.1145/3613905.3650868

W. Gan, et al., « Large Language Models in Education: Vision and Opportunities », In Proceedings of 2023 IEEE International Conference on Big Data (BigData), pp. 4776-4785, 2023.
https://doi.org/10.48550/arxiv.2311.13160

S. Laato, et al., « AI-Assisted Learning with ChatGPT and Large Language Models: Implications for Higher Education », In Proceedings of 2023 IEEE International Conference on Advanced Learning Technologies (ICALT), Orem, UT, USA, pp. 226-230, 2023.
https://doi.org/10.1109/ICALT58122.2023.00072

Z. Liu, S.X. Yin, and N.F. Chen., « Optimizing Code-Switching in Conversational Tutoring Systems: A Pedagogical Framework and Evaluation », In Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pp. 500–515, Sept. 2024.
https://aclanthology.org/2024.sigdial-1.43.pdf

N. Rane, « Enhancing the Quality of Teaching and Learning through ChatGPT and Similar Large Language Models: Challenges, Future Prospects, and Ethical Considerations in Education », Social Science Research Network, (SSRN), Sept. 2023.
https://doi.org/10.2139/ssrn.4599104

L. Yan, et al., « Practical and Ethical Challenges of Large Language Models in Education: A Systematic Scoping Review », British Journal of Educational Technology, 2023.
https://doi.org/10.1111/bjet.13370

Scan the QR code to access state-of-the-art research papers, datasets, codes, benchmarks, real-world use cases, and educational materials.

SDG#5 GENDER-BASED VIOLENCE CLASSIFICATION FROM TWEETS WITH ATTENTION-BASED BI-GRU

Over 230 million girls and women worldwide are estimated to have undergone female genital mutilation as of 2024, and globally, around 640 million girls and women were married before age 18, with India accounting for one-third.

Using AI to detect gender-based violence (GBV) online is crucial for identifying harmful patterns and content at scale, which would be impossible through manual moderation alone. It helps protect vulnerable individuals by flagging threats, harassment, and abuse in real time, enabling faster intervention and making digital spaces safer and more equitable for all genders.

Source (Retrieved on March 25th, 2025): https://sdgs.un.org/goals/goal5#progress_and_info

SDG#5 GENDER-BASED VIOLENCE CLASSIFICATION FROM TWEETS WITH ATTENTION-BASED BI-GRU

GRUs offer a good trade-off between complexity and performance in sequential modeling tasks.

The performance of Bi-GRU with attention can be very sensitive to hyperparameters and needs large memory.

A GRU is a type of recurrent neural network (RNN) designed to handle sequence data effectively. It was introduced to address the vanishing gradient problem found in traditional RNNs. GRUs have gating mechanisms that control the flow of information, making them more efficient than standard RNNs. Attention-based GRUs are lighter and faster than transformer models but still offer significant gains over plain RNNs or GRUs.

METHOD

A Bi-GRU (Bidirectional GRU) architecture is designed to capture information from both the past and future of a sequence. It consists of two types of GRU layers: one processes the input sequence from left to right (forward direction), and the other from right to left (backward direction). Each GRU layer outputs a hidden state at every time step, creating two distinct sequences of hidden states. The outputs from both GRUs are typically concatenated at each time step to provide a richer, bidirectional representation. This bidirectional nature allows the model to learn dependencies in both directions, improving performance on tasks with context-sensitive information. The sequence of concatenated hidden states is then passed to an attention mechanism for further refinement. The attention mechanism computes a context vector by assigning different attention weights to different time steps. These attention weights represent how important each hidden state is for the final prediction at a given time step. The context vector effectively provides a dynamic, focused summary of the entire input sequence at each step. The Bi-GRU with attention is particularly effective in handling long-range dependencies and long input sequences. Overall, the combination of bidirectional context from the GRUs and dynamic focus via attention provides a powerful architecture for sequence modeling.

1. INPUT

Collection of tweets with 5 types of labels for training: emotional, sexual, economic, physical violence and harmful practice

[redacted] 15 Jun 2020
Replying to @someone
You are a #&$@! as far as I've seen
3

[redacted] 15 Jun 2020
Replying to @someone
As a "nigger" and concerned citizen: you're a @¡$# !

[redacted] 15 Jun 2020
Replying to @someone
Ugly and fat with %!@ hair. Sad to be you :(

[redacted] 15 Jun 2020
Replying to @someone
4chan wants you. good luck ¡#@%
1

2. ARCHITECTURE

Preprocessing

replace #hashtag
replace @username
remove missing values
remove punctuations
remove stop words
remove numbers
remove retweet
eliminate special characters
lowering
lemmatization
emoji's handling
replace url

Input layer

Word Embedding

DxK Representation Text

GloVe

Spell Check

Point of Speech

Concatenation Pooling Convolution

Bi-GRU
Bi-GRU
Bi-GRU

W

Attention weights

Normalization
Merge layer
Output layer

3. OUTPUT

Tweet classification

[redacted] 20 Jun 2022
Replying to @someone
You are a !%¡&! @#!! monster to kill
3

physical_violence

[redacted] 20 Jun 2022
Replying to @someone
As a "nigger" and concerned citizen: you're a ¡#@%

economic_violence

FURTHER READING

M.A.B. Abbass and H.-S. Kang, « Violence Detection Enhancement by Involving Convolutional Block Attention Modules Into Various Deep Learning Architectures: Comprehensive Case Study for UBI-Fights Dataset », in IEEE Access, vol. 11, pp. 37096-37107, 2023. https://doi.org/10.1109/ACCESS.2023.3267409

G. Abercrombie, et al., « Resources for Automated Identification of Online Gender-Based Violence: A Systematic Review », In Proceedings of the 7th Workshop on Online Abuse and Harms (WOAH), pp. 170-186, Toronto, Canada, 2023. https://aclanthology.org/2023.woah-1.17/

C.M. Castorena, et al., « Deep Neural Network for Gender-Based Violence Detection on Twitter Messages », Mathematics, vol. 9, issue 8, no. 807, 2021. https://doi.org/10.3390/MATH9080807

G. Miranda, et al., « Deep Neural Network to Detect Gender Violence on Mexican Tweets », Artificial Intelligence and Pattern Recognition, pp. 24-32, 2021. https://doi.org/10.1007/978-3-030-89691-1_3

C. Suman, et al., « An Attention Based Multi-Modal Gender Identification System for Social Media Users », Multimedia Tools and Applications, pp. 1-23, 2021. https://doi.org/10.1007/S11042-021-11256-6

M.Z. Ur Rehman, et al., « A Context-Aware Attention and Graph Neural Network-Based Multimodal Framework for Misogyny Detection », Inf. Process. Manage., vol. 62, no. 1, Jan. 2025. https://doi.org/10.1016/j.ipm.2024.103895

Scan the QR code to access state-of-the-art research papers, datasets, codes, benchmarks, real-world use cases, and educational materials.

SDG#6 WATER SANITATION PREDICTION USING LS-SVM

More than 1,000 children under 5 die every day from diseases related to lack of clean water, sanitation, and hygiene and 1.69 billion people live without access to adequate sanitation.

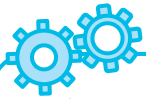
Developing AI tools for water pollution prediction enables the early detection of contamination risks by analyzing complex environmental data in real time. This supports timely intervention, protects public health, and ensures more effective management of water resources.

Source (Retrieved on March 25th, 2025): https://www.worldvision.org/clean-water-news-stories/global-water-crisis-facts

SDG#6 WATER SANITATION PREDICTION USING LS-SVM

LS-SVMs are able to capture non-linear relationships in data, providing accurate forecasts aiding in proactive sanitation measures.

LS-SVMs are computationally intensive and sensitive to the choice of kernel parameters.

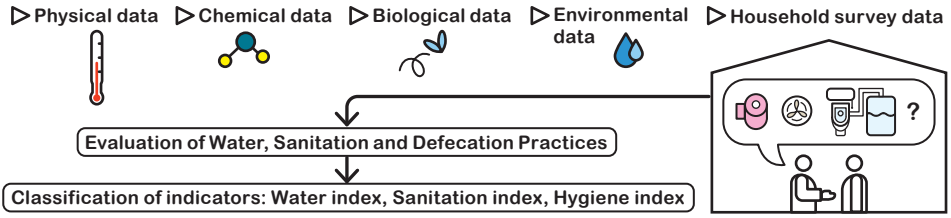


Least Squares Support Vector Machines (LS-SVMs) can predict water quality by learning from historical data to classify or regress water parameters. It minimizes simultaneously the margin and the sum of square errors (SSEs) on training samples to make accurate predictions in water pollution monitoring and contaminant identification.

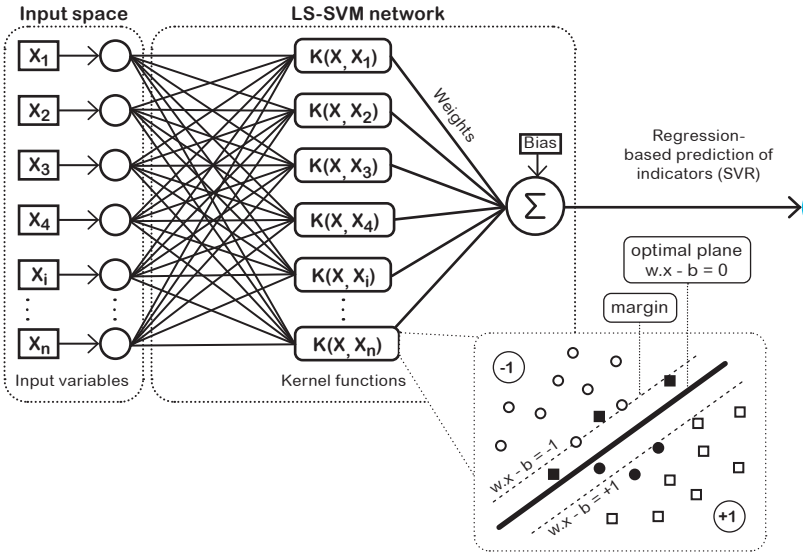
METHOD

LS-SVM (Least Squares Support Vector Machine) modifies the standard SVM formulation by replacing the quadratic loss function with a least squares loss function. This change leads to a linear system of equations instead of the typical convex quadratic optimization problem. The idea behind SVM is to find a hyperplane that separates classes with a maximum margin. SVR is an extension of SVM used from predicting numerical values using regression. In LS-SVM, the objective is to minimize the squared error between the predicted values and the true values, while maintaining a large margin for separation. Like traditional SVM, LS-SVM uses a kernel function to map the input features into a higher-dimensional space where linear separation is easier. The kernel trick allows LS-SVM to handle non-linear relationships by computing the inner products in the higher-dimensional space without explicitly mapping the data. The architecture consists of two main parts: the feature transformation (via the kernel) and the model learning phase. In the model learning phase, LS-SVM seeks to minimize a loss function that combines both the squared error and a regularization term for margin maximization. The regularization term ensures that the solution does not overfit the data. The choice of kernel (e.g., linear, polynomial, Gaussian) significantly affects the model's ability to generalize to different types of data.

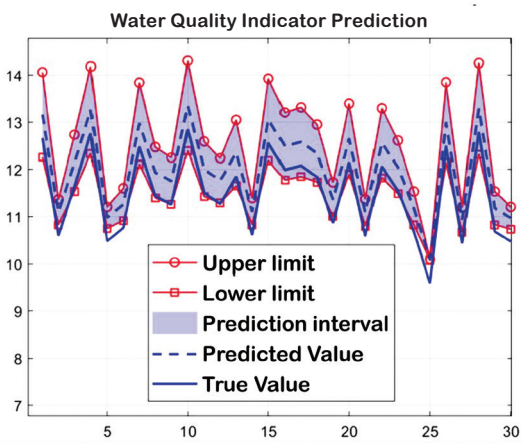
1. INPUT

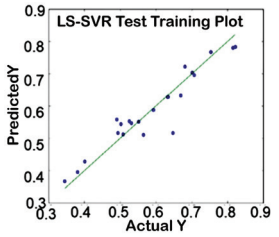


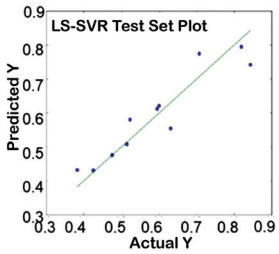
2. ARCHITECTURE



3. OUTPUT







FURTHER READING

A.P. Dadhich, P.N. Dadhich, and R. Goyal, « Synthesis of Water, Sanitation, and Hygiene (WaSH) Spatial Pattern in Rural India: an Integrated Interpretation of WaSH Practices », Environ. Sci. Pollut. Res., vol. 29, pp. 86873–86886, 2022.
https://doi.org/10.1007/s11356-022-21918-z

W.C. Leong, et al., « Prediction of Water Quality Index (WQI) Using Support Vector Machine (SVM) and Least Square-Support Vector Machine (LS-SVM) », International Journal of River Basin Management, vol. 19, no. 2, pp. 149-156, 2019.
https://doi.org/10.1080/15715124.2019.1628030

J. Ruan, et al., « A Novel RF-CEEMD-LSTM Model for Predicting Water Pollution », Sci. Rep., vol. 13, no. 20901, 2023.
https://doi.org/10.1038/s41598-023-48409-6

M.Y. Shams, et al., « Water Quality Prediction Using Machine Learning Models Based on Grid Search Method », Multimedia Tools and Applications, vol. 83, pp. 35307-35334, 2024.
https://doi.org/10.1007/s11042-023-16737-4

K.P. Wai, et al., « Applications of Deep Learning in Water Quality Management: A State-Of-The-Art Review », Journal of Hydrology, vol. 613, Part A, 2022.
https://doi.org/10.1016/j.jhydrol.2022.128332

Y. Xiang and L. Jiang, « Water Quality Prediction Using LS-SVM and Particle Swarm Optimization », In Proceedings of 2009 Second International Workshop on Knowledge Discovery and Data Mining, Moscow, Russia, pp. 900-904, 2009.
https://doi.org/10.1109/WKDD.2009.217

Scan the QR code to access state-of-the-art research papers, datasets, codes, benchmarks, real-world use cases, and educational materials.

SDG#7 CONTROLLING & SCHEDULING ENERGY WITH DEEP REINFORCEMENT LEARNING

Almost 800 million people globally have no electricity, and about 2.6 billion, a third of the world's population, have no access to clean cooking fuels. The lack of clean energy not only harms the environment but also kills 1.6 million people in the world every year from fumes from burning fuels like charcoal to cook food.

Developing AI tools for energy control and scheduling enables smarter, real-time management of energy resources, improving efficiency and reducing costs. These tools can help balance supply and demand, integrate renewable energy, and optimize usage across grids, buildings, and devices.

Source (Retrieved on March 25th, 2025): https://www.un.org/en/climatechange/damilola-ogunbiyi-ending-energy-poverty

SDG#7 CONTROLLING & SCHEDULING ENERGY WITH DEEP REINFOREMENT LEARNING

DRL ensures efficient decision-making, adapts to dynamic environments, with continuous improvement.

DRL can be unstable, sensitive to hyperparameter settings, and requires significant computational resources and training data.

Deep Reinforcement Learning (DRL) can optimize energy management by learning to make decisions that maximize efficiency and reduce costs, adapting to dynamic environments like power systems, and improving over time with continuous feedback and adjustments. DRL algorithms typically rely on exploration and exploitation strategies, like epsilon-greedy (which balances random actions and learned actions) or entropy maximization.

METHOD

DRL involves training agents using neural networks to optimize decisions from collected observations of an environment. The agent learns from interactions by receiving rewards for each of its actions, aiming to maximize the reward signal over time. The agent takes actions in the environment, observes the resulting states, and receives feedback in the form of a reward. The agent's goal is to learn a policy that maps states to actions to maximize its cumulative reward. Deep learning models, particularly deep neural networks, are used to approximate the value function or policy due to the high-dimensionality of state spaces. The value function, typically denoted as $V(s)$, estimates the expected future reward from a given state s, while the policy $\pi(s)$ defines the action to take at each state. Deep Q-Network (DQN) uses a neural network to approximate the Q-value function, $Q(s,a)$, where Q represents the expected future reward for state s and action a. DQN uses discrete action spaces and is based on a value function. It estimates the maximum possible reward attainable from a given state using a Q-value function updated via the Bellman equation. DDPG (Deep Deterministic Policy Gradient), suitable for continuous action spaces, operates on a policy gradient method where it directly learns the optimal policy that maximizes reward, unlike DQN, which selects actions based on Q-values. DDPG utilizes actor-critic architecture to stabilize learning in such environments.

1. INPUT

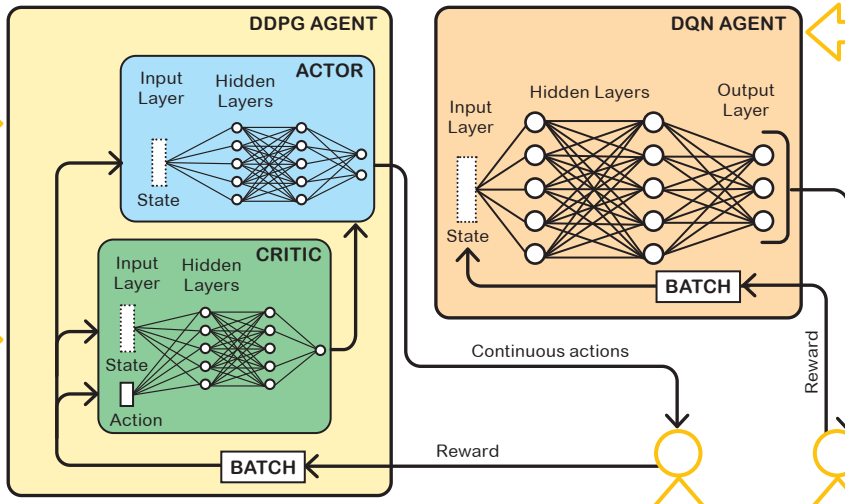
▷ Historical Databases

Industrial consumers | Electrified vehicles | Trains | Solar panels | Data centers | Wind farms | Residential consumers

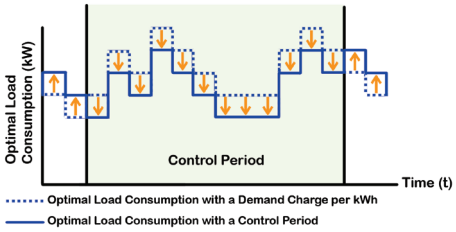
electricity price | solar radiation | living habits

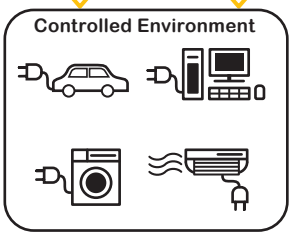
Observations

2. ARCHITECTURE



3. OUTPUT





BELLMAN EXPECTION EQUATION

policy to select the agent's actions for all time steps

$$Q_{\pi}(s,a) = E_{\pi}\,[R_{t+1} + \gamma * max(Q_{\pi}(s_{t+1}, a_{t+1})]$$

- $Q_{\pi}(s,a)$: Value of taking action '*a*' in state '*s*'
- E_{π}: Expected value of immediate reward
- γ: discount factor determining the importance of future reward
- $max(Q_{\pi}(s_{t+1}, a_{t+1}))$: maximum value among all possible actions in the next state

FURTHER READING

P.L. Donti and J. Z. Kolter, « Machine Learning for Sustainable Energy Systems », Annual Review of Environment and Resources, vol. 46, pp. 719-747, 2021.
https://doi.org/10.1146/annurev-environ-020220-061831

A. Jayanetti, S. Halgamuge, and R. Buyya, « Deep Reinforcement Learning for Energy and Time Optimized Scheduling of Precedence-constrained Tasks in Edge-cloud Computing Environments », Future Generation Computer Systems, vol. 137, pp. 14-30, 2022.
https://doi.org/10.1016/j.future.2022.06.012

S. Jittanon, Y. Mensin, and C. Termritthikun, « Intelligent Forecasting of Energy Consumption using Temporal Fusion Transformer model », In Proceedings of 2023 IEEE International Conference on Cybernetics and Innovations (ICCI), Thailand, pp. 1-5, 2023.
https://doi.org/10.1109/ICCI57424.2023.10112297

S. Keren, et al., « Multi-Agent Reinforcement Learning for Energy Networks: Computational Challenges, Progress and Open Problems », arXiv:2404.15583v1, 2024.
https://arxiv.org/abs/2404.15583v1

K. Ponse, et al., « Reinforcement Learning for Sustainable Energy: A Survey », arXiv:2407.18597v1, 2024.
https://arxiv.org/abs/2407.18597v1

Z. Yao, et al., « Machine Learning for a Sustainable Energy Future ». Nat. Rev. Mater., vol. 8, pp. 202–215, 2023.
https://doi.org/10.1038/s41578-022-00490-5

C. Yeh, et al., « SustainGym: A Benchmark Suite of Reinforcement Learning for Sustainability Applications ». In Proceedings of the 2023 International Conf. on Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track, New Orleans, LA, USA, Dec. 2023.
https://openreview.net/forum?id=vZ9tA3o3hr

Scan the QR code to access state-of-the-art research papers, datasets, codes, benchmarks, real-world use cases, and educational materials.

SDG#8 PREDICTION OF GHG EMISSIONS WITH TABULAR BACKBONES

The 'average' person produces 6.28 tonnes of GHG emissions annually. But this number varies widely by country and income level. Wealthier, higher-consuming populations may emit up to 110 tonnes of CO_2 equivalent (CO_2eq) per year. Among lower-income groups, emissions can be as low as 1.6 tonnes of CO_2eq per year.

Predicting GHG emissions with AI can enable accurate, real-time forecasting based on complex and dynamic data from various sectors to help policymakers and industries monitor progress, design effective mitigation strategies, and meet climate targets more efficiently.

Source (Retrieved on June 18th, 2025): https://www.wri.org/insights/climate-impact-behavior-shifts

SDG#8 PREDICTION OF GHG EMISSIONS WITH TABULAR BACKBONES

Tabular foundation models support downstream tasks with few labeled samples using transferability.

Tabular foundation models require large-scale pretraining datasets and are computationally expensive to fine-tune.

Tabular foundation models (TFM) are trained on large tabular datasets to capture generalized representations. They use attention or transformer-style architectures tailored for table formats and they can learn representations of column types, distributions, and cross-feature relations outperforming XGBoost and traditional MLPs on structured data. They reduce the need for manual feature engineering and domain-specific preprocessing.

METHOD

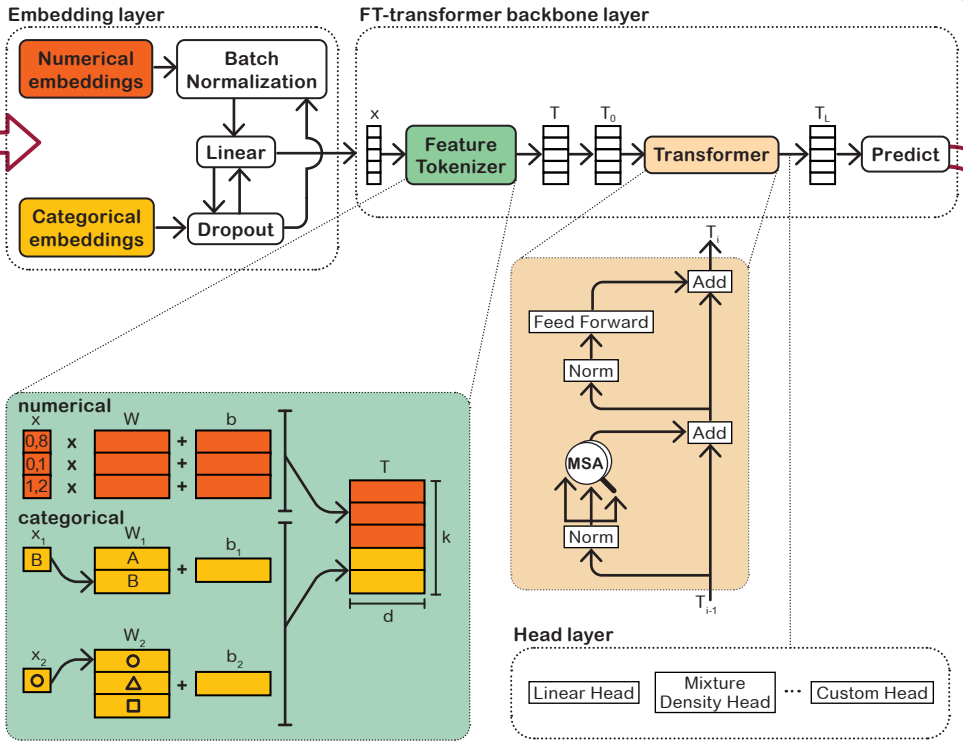
The TFM architecture typically includes an embedding layer to handle categorical features by converting them into dense vector representations. Continuous features (numeric variables) are often passed directly into the model, possibly undergoing normalization or scaling. These embeddings and numeric features are then concatenated to form a unified input vector for the model. The model often uses feedforward neural networks (i.e., fully connected layers) to process the data, with one or more hidden layers. Each hidden layer applies linear transformations followed by non-linear activation functions like ReLU (Rectified Linear Unit) to introduce complexity. Dropout or batch normalization is typically applied between layers to reduce overfitting and stabilize training. The final layer of the network produces a single scalar value for regression tasks or a probability distribution for classification tasks. To handle feature interactions effectively, more sophisticated techniques like attention mechanisms or cross-product transformations are used. Unlike traditional models (like decision trees or logistic regression), TFMs leverage deep learning to automatically capture complex relationships between features. The model is trained using backpropagation and gradient-based optimization methods (e.g., Adam or SGD) to minimize a loss function.

1. INPUT

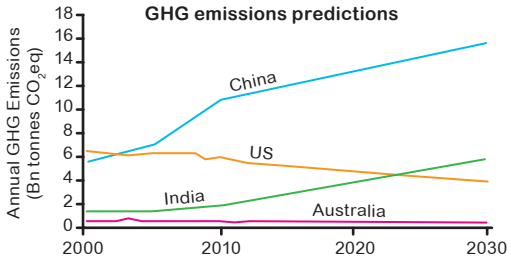
▷ Collection of tabular economic data

Region | Sector | Value Added (M. EUR) | Employment (1000 p.) | GHG emissions (kg CO_2eq) | Energy carrier Net Total (TJ) | Year

2. ARCHITECTURE



3. OUTPUT



LEGEND

MSA — Multihead Self-Attention

FURTHER READING

Y. Gorishniy, I. Rubachev, and A. Babenko, « On Embeddings for Numerical Features in Tabular Deep Learning », arXiv:2203.05556, 2022.
https://arxiv.org/abs/2203.05556

D. Singh, et al., « Machine-Learning- and Deep-Learning-Based Streamflow Prediction in a Hilly Catchment for Future Scenarios Using CMIP6 GCM Data », Hydrol. Earth Syst. Sci., vol. 27, pp. 1047-1075, 2023.
https://doi.org/10.5194/hess-27-1047-2023

A. Verma, et al., « Performance Comparison of Deep Learning Models for CO_2 Prediction: Analyzing Carbon Footprint with Advanced Trackers », In Proceedings of 2024 IEEE International Conference on Big Data (BigData), Washington, DC, USA, pp. 4429-4437, 2024.
https://doi.org/10.1109/BigData62323.2024.10825767

Z. Wang, et al., « AnyPredict: Foundation Model for Tabular Prediction », arXiv:2305.12081, 2023.
https://doi.org/10.48550/arXiv.2305.12081

X. Wu , et al., « Carbon Emissions Forecasting Based on Temporal Graph Transformer-Based Attentional Neural Network », Journal of Computational Methods in Sciences and Engineering. vol. 24, no. 3, pp. 1405-1421, 2024.
https://doi.org/10.3233/JCM-247139

T. Zhang, et al., « Generative Table Pre-training Empowers Models for Tabular Prediction », arXiv:2305.09696, 2023.
https://arxiv.org/abs/2305.09696

Scan the QR code to access state-of-the-art research papers, datasets, codes, benchmarks, real-world use cases, and educational materials.

SDG#9 IOT ANOMALY DETECTION WITH MLP

By 2025, there will be approximately 13.1 billion connected devices, and the number of installed IoT devices is projected to reach 42.62 billion. As IoT adoption accelerates, particularly in industrial settings, the scale and complexity of connected systems make them increasingly vulnerable to faults, cyberattacks, and system failures.

AI can detect anomalies and threats in real time, helping to prevent disruptions, data theft, and cascading failures across connected systems.

Source (Retrieved on June 18th, 2025): https://techjury.net/industry-analysis/iot/

SDG#9 IOT ANOMALY DETECTION WITH MLP

MLP is a simple architecture, fast to train with basic hardware. It doesn't require explicit feature engineering.

MLPs are sensitive to hyperparameters tuning and can be prone to overfitting.

MLPs are versatile models suitable for various tasks, including classification, regression, and function approximation, due to their simple yet powerful architecture. They can handle a variety of input types, including tabular data, images (when pre-processed), and other structured data. They can capture complex, non-linear relationships between inputs and outputs.

METHOD

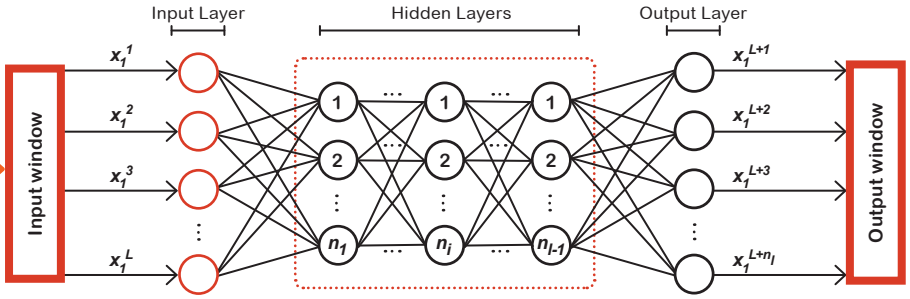
A Multilayer Perceptron (MLP) consists of at least three types of layers: an input layer, one or more hidden layers, and an output layer. The input layer receives the raw data, such as features or images, and passes it to the next layer. Each neuron in the input layer corresponds to a feature in the input data, and these inputs are forwarded to the first hidden layer. The hidden layers are fully connected layers, meaning every neuron in one layer is connected to every neuron in the next layer. Each connection has an associated weight that is learned during training, which helps in adjusting the strength of the signal passed between neurons. The output of each neuron is computed as a weighted sum of the inputs followed by a non-linear activation function, such as ReLU (Rectified Linear Unit) or sigmoid. The choice of activation function introduces non-linearity, enabling the MLP to model complex relationships between inputs and outputs. The MLP can have multiple hidden layers, allowing it to learn hierarchical feature representations and increasing its capacity to model intricate patterns. The output layer is responsible for producing the final prediction, with the number of neurons typically matching the number of output classes for classification or a single neuron for regression tasks. During training, the MLP is optimized by minimizing a loss function through backpropagation.

1. INPUT

▷ Collection of IoT telemetry data for intrusion detection and threat analysis

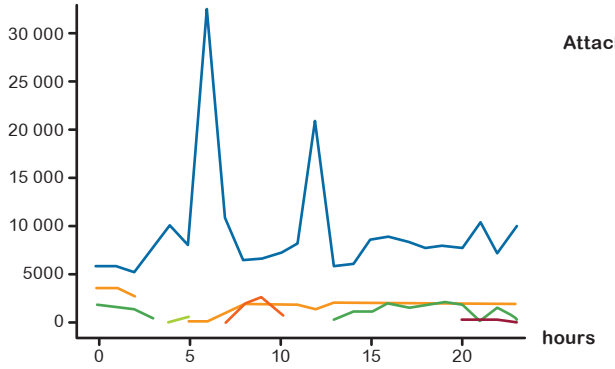
window		date	time	FC1_Read_Input_Register	FC2_Read_Discrete_Value	FC3_Read_Holding_Register	FC4_Read_Coil	label
	0	31-Mar-19	12:36:55	53287	1463	33518	23014	0
	1	31-Mar-19	12:36:58	41029	55891	26004	50645	0
	2	31-Mar-19	12:36:58	41029	55891	26004	50645	0
	3	31-Mar-19	12:37:00	64661	40232	33460	44046	0

2. ARCHITECTURE



3. OUTPUT

Detected anomalies



Attack types distribution

- normal
- backdoor
- password
- injection
- scanning
- xss

FURTHER READING

B.D. Ananya, K.S. Mahalakshmi, and P. Joshi, « Advancing IoT Security: A Stacked Hybrid AI Approach for Anomaly Detection », 2024 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), Bangalore, India, 2024, pp. 1-6, 2024. https://doi.org/10.1109/CONECCT62155.2024.10677130

R. Ahmad, I. Alsmadi, « Machine Learning Approaches to IoT Security: A Systematic Literature Review », Internet of Things, vol. 14, 2021. https://doi.org/10.1016/j.iot.2021.100365

G. Raman, N. Somu, and A.P. Mathur, « A Multilayer Perceptron Model for Anomaly Detection in Water Treatment Plants », International Journal of Critical Infrastructure Protection, vol. 31, 2020. https://doi.org/10.1016/j.ijcip.2020.100393

S. Tsimenidis, T. Lagkas, and K. Rantos, « Deep Learning in IoT Intrusion Detection », J. Netw. Syst. Manage., vol. 30, no. 8, 2022. https://doi.org/10.1007/s10922-021-09621-9

G. Sivapalan, K.K. Nundy, S. Dev, B. Cardiff, and D. John, « ANNet: A Lightweight Neural Network for ECG Anomaly Detection in IoT Edge Sensors », in IEEE Transactions on Biomedical Circuits and Systems, vol. 16, no. 1, pp. 24-35, Feb. 2022. https://doi.org/10.1109/TBCAS.2021.3137646

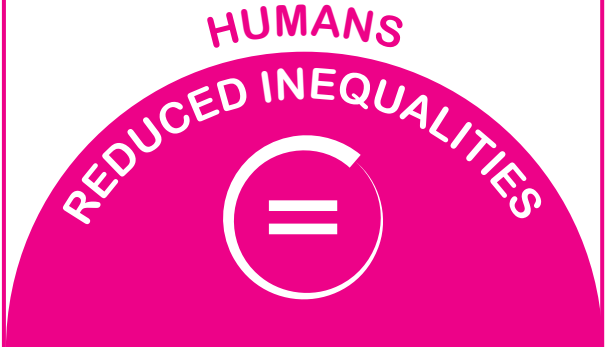
L. Van Efferen and A.M.T. Ali-Eldin, « A Multi-Layer Perceptron Approach for Flow-Based Anomaly Detection », In Proceedings of 2017 International Symposium on Networks, Computers and Communications (ISNCC), Marrakech, Morocco, pp. 1-6, 2017. https://doi.org/10.1109/ISNCC.2017.8072036

Scan the QR code to access state-of-the-art research papers, datasets, codes, benchmarks, real-world use cases, and educational materials.

SDG#10 COMBATING HUMAN-TRAFFICKING WITH SWIN TRANSFORMER

With nearly half of the world's population living on less than $6.85 per person per day, or with at least three billion people worldwide living in areas severely affected by climate change and non-climatic environmental degradation, millions of individuals have become vulnerable to exploitation.

AI can combat human trafficking by analyzing patterns in online ads at scale, financial transactions, and travel data to detect suspicious activity and identify trafficking networks. It enables law enforcement and NGOs to act faster and more precisely, improving victim rescue efforts and disrupting criminal operations.



Source (Retrieved on June 18th, 2025): https://www.unodc.org/unodc/frontpage/2024/May/8-facts-you-need-to-know-about-human-trafficking-in-the-21st-century.html

SDG#10 COMBATING HUMAN-TRAFFICKING WITH SWIN TRANSFORMER

Swin transformers can efficiently model local and global visual features in images with high scalability.

Swin transformers require significant computational resources and high memory consumption during training.

Swin Transformer (Shifted Window Transformer) is a state-of-the-art architecture designed to efficiently handle high-resolution images while maintaining strong performance in computer vision tasks. Unlike traditional transformer architectures which apply global attention over the entire image, Swin Transformer uses local attention within non-overlapping windows to reduce computational complexity.

METHOD

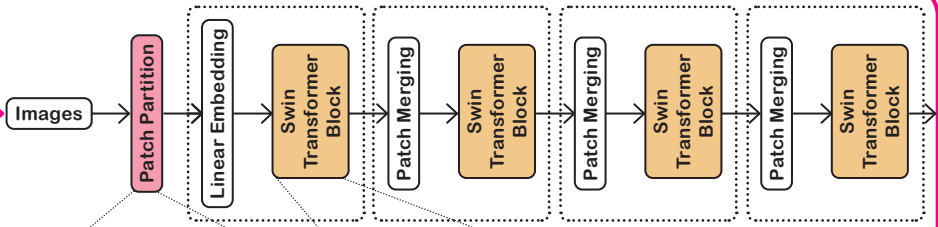
The key idea of Swin Transformer is the use of a shifted window approach, where the windows used for attention are shifted between successive layers, enabling the model to capture both local and global context across different layers. The input image is first divided into patches, and each patch is embedded into a fixed-size token vector using a patch embedding layer. These tokenized patches are processed through several Swin Transformer blocks, where each block consists of a series of multi-head self-attention layers applied within the local windows. The attention mechanism operates within these windows to capture local dependencies, but shifting the windows at each layer helps the model learn global dependencies as well. The attention computation is performed in linear time, more computationally efficient compared to standard Transformers, which operate in quadratic time. In addition to the window-based attention, Swin Transformers also incorporate a shifted window partitioning strategy where the windows overlap in successive layers, ensuring the model can learn from various spatial configurations. Each Swin Transformer block consists of a window-based multi-head self-attention layer and an MLP-based feedforward network, followed by normalization layers (LN). The architecture is hierarchical, i.e., the resolution of the input representation progressively decreases through the network, while the number of channels (features) increases.

1. INPUT

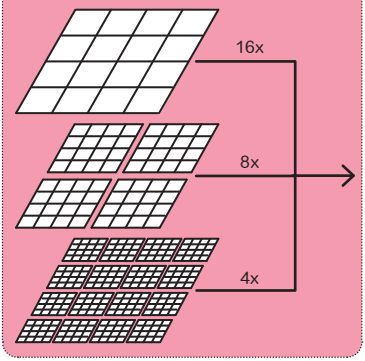
▷ A queried image to identify the hotel

▷ A collection of images of hotel rooms (e.g., from TraffickCam-Hotels-50K dataset)

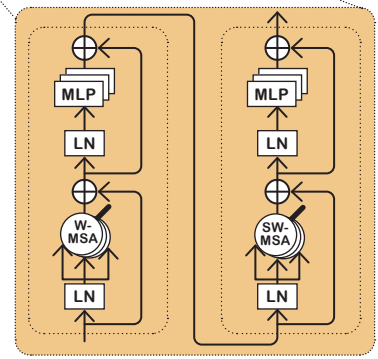
2. ARCHITECTURE



Hierarchical representation with non-overlapping patches

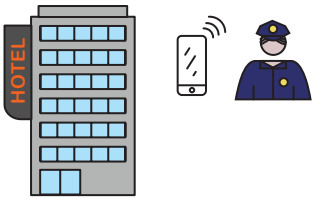


Two successive Swin Transformer blocks



3. OUTPUT

Recognition and identification of the hotel from the input image



LEGEND

LN Layer Norm

MLP Multilayer Perceptron

W-MSA Regular windowing Multihead Self-Attention

SW-MSA Shifted windowing Multihead Self-Attention

⊕ Element-wise Addition

FURTHER READING

S.S. Esfahani, et al., « Context-specific Language Modeling for Human Trafficking Detection from Online Advertisements », In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 1180-1184, Florence, Italy, 2019.
https://aclanthology.org/P19-1114/

A. Hatamizadeh, et al., « Swin UNETR: Swin Transformers for Semantic Segmentation of Brain Tumors in MRI Images », Lecture Notes in Computer Science, vol. 12962, Springer, Cham, 2022.
https://doi.org/10.1007/978-3-031-08999-2_22

A.P. Joshi, et al., « HotelWatch: A Hotel Identification System to Combat Human Trafficking, » 2024 IEEE International Conference on Big Data (BigData), Washington, DC, USA, pp. 2801-2810, 2024.
https://doi.org/10.1109/BigData62323.2024.10825725

V.K. Saxena, et al., « MATCHED: Multimodal Authorship-Attribution To Combat Human Trafficking in Escort-Advertisement Data », arXiv:2412.13794, 2024.
https://arxiv.org/abs/2412.13794

A. Stylianou, et al., « Hotels-50K: A Global Hotel Recognition Dataset », In Proceedings of AAAI Conference on Artificial Intelligence, 2019.
https://doi.org/10.1609/aaai.v33i01.3301726

Y. Tang, et al., « Self-Supervised Pre-Training of Swin Transformers for 3D Medical Image Analysis », In Proceedings of 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, pp. 20698-20708, 2022.
https://doi.org/10.1109/CVPR52688.2022.02007

Scan the QR code to access state-of-the-art research papers, datasets, codes, benchmarks, real-world use cases, and educational materials.

SDG#11 PREDICTING SEA LEVEL CHANGE USING LSTM

Chronic water scarcity affecting more than 40% of the global population, hydrological uncertainty, and extreme weather events (floods and droughts) are perceived as some of the biggest threats to global prosperity and stability. Water-related disasters account for 70% of all deaths related to natural disasters. Flood damages are estimated at around USD 120 billion per year (only from property damage).

Accurately predicting extreme events and flood damages is critical for warning the populations and prioritizing disaster response.

Source (Retrieved on March 25th, 2025): https://www.who.int/health-topics/floods#tab=tab_1 https://www.worldbank.org/en/topic/waterresourcesmanagement

SDG#11 PREDICTING SEA LEVEL CHANGE USING LSTM

LSTMs can capture long-term dependencies and patterns over longer sequences compared to traditional RNNs.

LSTMs require large amounts of data for effective but they are computationally expensive to train and may overfit on small datasets.

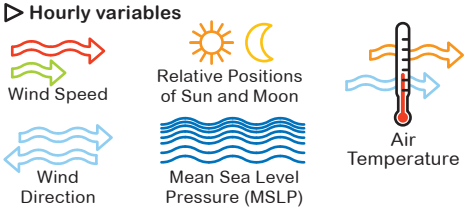
Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) designed to overcome the vanishing gradient problem and capture long-term dependencies. LSTM cells have memory units that can store and retrieve information over extended periods. LSTMs are suitable for modeling complex patterns. They learn from past sea level variations to predict future changes, aiding in understanding and mitigating the impacts of climate change.

METHOD

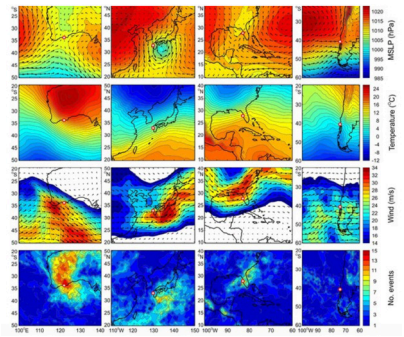
LSTM (Long Short-Term Memory) is a type of Recurrent Neural Network (RNN) designed to address the limitations of traditional RNNs, specifically the vanishing and exploding gradient problems. LSTM models are particularly effective for processing and predicting sequences of data, where temporal dependencies are important. Unlike RNNs, LSTMs have specialized units called memory cells that help retain information over long time periods. LSTM consists of three primary gates: the input gate, the forget gate, and the output gate. The input gate controls how much new information from the current time step should be stored in the memory cell. The forget gate decides what portion of the past information should be discarded from the memory, allowing the model to « forget » irrelevant data. The output gate determines what part of the memory cell should be output at the current time step, which influences the hidden state passed to the next time step. The cell state is a key feature of LSTM carrying information across time steps with minimal changes. During training, the model learns how to update the cell state and the gates, allowing it to capture long-term dependencies in the data. LSTMs are typically trained using backpropagation through time, where gradients are computed at each time step and propagated back through the network to update the weights.

1. INPUT

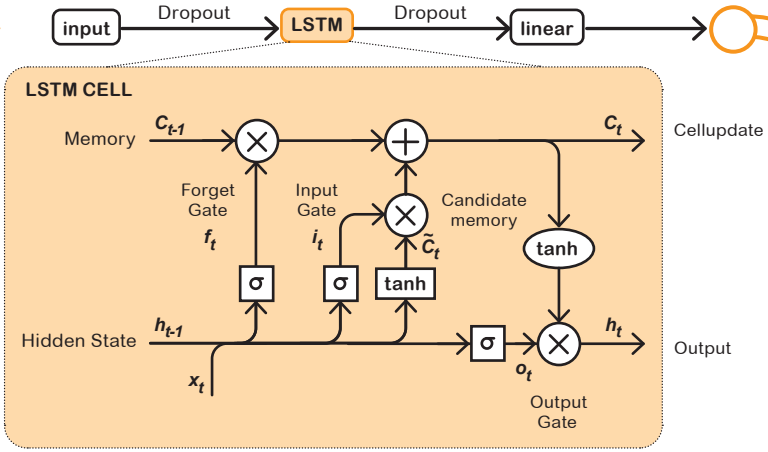
▷ Hourly variables



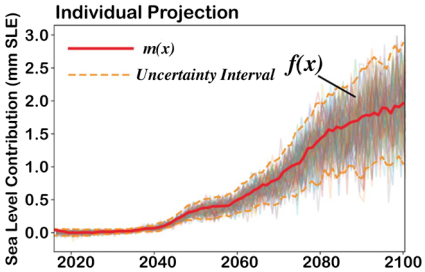
▷ Annual Global Air Temperature

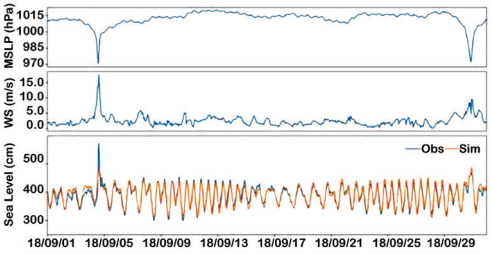


2. ARCHITECTURE



3. OUTPUT





NOTATIONS

Inputgate: $i_t = \sigma\,(W^{(ii)}\overline{x}_t + W^{(ii)}h_{t-1})$ **Forgetgate:** $f_t = \sigma\,(W^{(if)}\overline{x}_t + W^{(hf)}h_{t-1})$ **Outputgate:** $i_t = \sigma\,(W^{(io)}\overline{x}_t + W^{(ho)}\,h_{t-1})$

ProcessInput: $C_t = tanh\,(W^{(ic)}\overline{x}_t + W^{(hc)}\,h_{t-1})$ **Cellupdate:** $C_t = f_t \,.\, C_{t-1} + i_t \,.\, \tilde{C}_t$ **Output:** $y_t = h_t = O_t + i_t * tanh\,(C_t)$

FURTHER READING

N.A.A.B.S. Bahari, et al., « Predicting Sea Level Rise Using Artificial Intelligence: A Review », Arch. Computat. Methods Eng. vol. 30, pp. 4045-4062, 2023.
https://doi.org/10.1007/s11831-023-09934-9

A.L. Balogun and N. Adebisi, « Sea Level Prediction Using ARIMA, SVR and LSTM Neural Network: Assessing the Impact of Ensemble Ocean-Atmospheric Processes on Models' Accuracy », Geomatics, Natural Hazards and Risk, vol. 12, no. 1, pp. 653-674, 2021.
https://doi.org/10.1080/19475705.2021.1887372

K. Ishida, et al., « Hourly-Scale Coastal Sea Level Modeling in a Changing Climate Using Long Short-Term Memory Neural Network », Science of The Total Environment, vol. 720, 2020.
https://doi.org/10.1016/j.scitotenv.2020.137613

W. Li, A. Kiaghadi, and C. Dawson. « High Temporal Resolution Rainfall–Runoff Modeling Using Long-Short-Term-Memory (LSTM) Networks », Neural Comput. Appl. 33, pp. 1261-1278, 2021.
https://doi.org/10.1007/s00521-020-05010-6

O.M. Sorkhabi, B. Shadmanfar, M.M. Al-Amidi, « Deep Learning of Sea-Level Variability and Flood for Coastal City Resilience », City and Environment Interactions, vol. 17, 2023.
https://doi.org/10.1016/j.cacint.2022.100098

P. Van Katwyk, et al., « A Variational LSTM Emulator of Sea Level Contribution from the Antarctic Ice Sheet », Journal of Advances in Modeling Earth Systems, vol. 15, no. e2023MS003899, 2023.
https://doi.org/10.1029/2023MS003899

Scan the QR code to access state-of-the-art research papers, datasets, codes, benchmarks, real-world use cases, and educational materials.

SDG#12 WASTE CLASSIFICATION WITH ZERO-SHOT LEARNING WITH CLIP

In 2022, 19% of global food was wasted, totalling 1.05 billion tonnes, with household waste accounting for 60%. This waste generates significant greenhouse gas emissions, costing over $1 trillion annually, while 783 million people suffer from hunger.

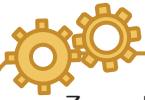
Classifying waste using image detection enables efficient and accurate sorting, which improves recycling rates. It helps automate the process, reducing the need for manual labor for more sustainable waste management practices.

Source (Retrieved on March 25th, 2025): https://sdgs.un.org/goals/goal12#progress_and_info

SDG#12 WASTE CLASSIFICATION WITH ZERO-SHOT LEARNING WITH CLIP

Zero-shot learning with CLIP enables models to generalize across a wide range of tasks without task-specific training.

CLIP's performance relies heavily on the alignment between language and visual information in the training data.



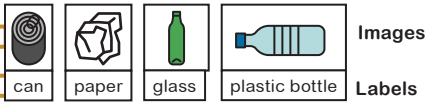
Zero-shot learning with CLIP (Contrastive Language-Image Pretraining) allows models to perform tasks like image classification, object detection without task-specific training by leveraging large-scale image-text pair datasets. By aligning images and textual descriptions in a shared embedding space, CLIP can generalize to a wide variety of tasks using natural language prompts, making it versatile and efficient across many domains.

METHOD

Zero-shot learning with CLIP uses a unified architecture that bridges vision and language by aligning images and text in a shared embedding space. The architecture consists of two main components: an image encoder and a text encoder. The image encoder can be a Vision Transformer (ViT) or a ResNet network, which processes images and converts them into feature vectors. The text encoder is a Transformer-based model (like GPT or BERT) that processes textual descriptions and converts them into corresponding feature vectors. Both encoders are trained simultaneously in a contrastive learning framework, where the goal is to bring the feature vectors of matching image-text pairs closer together in the shared embedding space. During training, the model is fed with a large dataset of paired image-text data, learning to understand. The model uses a contrastive loss function, such as InfoNCE loss, to maximize the similarity of positive pairs (correct image-text pairs) and minimize the similarity of negative pairs (incorrect image-text pairs). This results in the image and text encoders learning to map both modalities into a shared vector space, where similar images and descriptions are closer together. During inference, the model can take a textual description as input and retrieve the most relevant image by comparing the text's feature vector with the image feature vectors.

1. INPUT

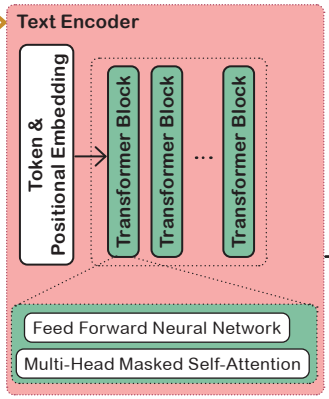
▷ Collection of images of identified waste

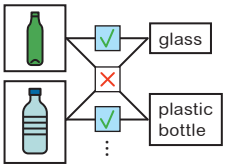


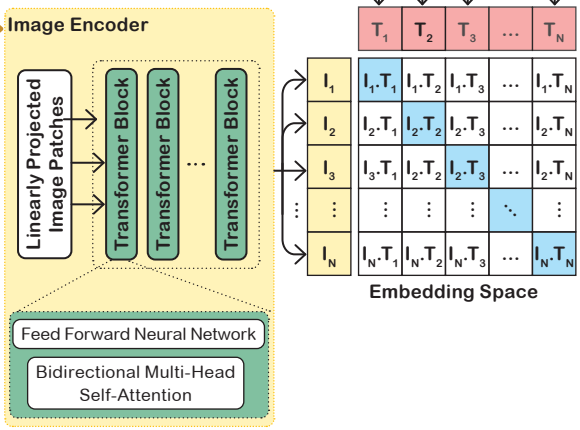
▷ An image of multiple objects to be detected and classified

2. ARCHITECTURE

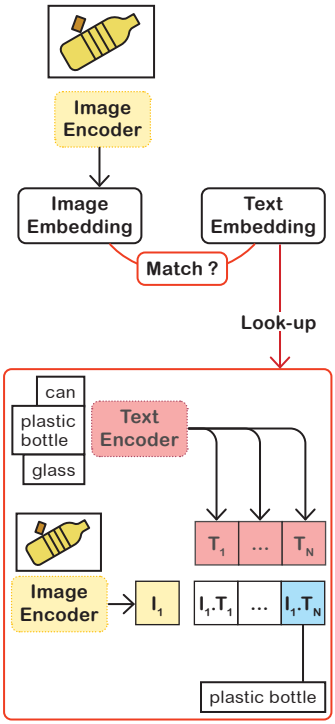
CONSTRASTIVE PRE-TRAINING







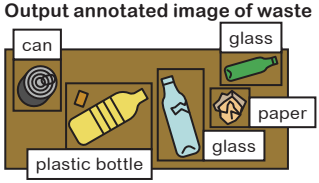
ZERO-SHOT CLASSIFICATION



3. OUTPUT

Input image of waste

Output annotated image of waste



FURTHER READING

F.S. Alrayes, et al., « Waste Classification Using Vision Transformer Based on Multilayer Hybrid Convolution Neural Network », Urban Climate, vol. 49, 2023.
https://doi.org/10.1016/j.uclim.2023.101483

N. Islam, et al., « EWasteNet: A Two-Stream Data Efficient Image Transformer Approach for E-Waste Classification », In Proceedings of 2023 IEEE International Conference On Software Engineering and Computer Systems (ICSECS), Penang, Malaysia, pp. 435-440, 2023.
https://doi.org/10.1109/ICSECS58457.2023.10256321

K. Huang, et al., « Recycling Waste Classification Using Vision Transformer on Portable Device ». Sustainability, vol. 13, issue 21, no. 11572, 2021.
https://doi.org/10.3390/su132111572

A. Kurz, et al., « WMC-ViT: Waste Multi-class Classification Using a Modified Vision Transformer », In Proceedings of 2022 IEEE MetroCon, Hurst, TX, USA, pp. 1-3, 2022.
https://doi.org/10.1109/MetroCon56047.2022.9971136

N.N.I. Prova, « Garbage Intelligence: Utilizing Vision Transformer for Smart Waste Sorting », In Proceedings of 2024 Second International Conference on Intelligent Cyber Physical Systems and Internet of Things (ICoICI), Coimbatore, India, pp. 1213-1219, 2024.
https://doi.org/10.1109/ICoICI62503.2024.10696177

A. Radford, et al., « Learning transferable visual models from natural language supervision », In Proceedings of International Conference on Machine Learning (ICML), pp. 8748-8763, 2021.
https://proceedings.mlr.press/v139/radford21a/radford21a.pdf

Scan the QR code to access state-of-the-art research papers, datasets, codes, benchmarks, real-world use cases, and educational materials.

SDG#13 FLOOD AREA SEGMENTATION FROM IMAGES USING UNET

Between 2013 and 2022, disasters worldwide claimed 42,553 mortalities each year. The number of persons affected by disasters per 100,000 population has increased by over two-thirds, from 1,169 in 2005-2014 to 1,980 in 2013-2022.

Satellite image segmentation of floods provides rapid, large-scale identification of affected areas, enabling timely and informed disaster response. It helps emergency teams prioritize regions needing immediate aid and plan evacuation or rescue operations more effectively with better coordination, resource allocation, and mitigation of further risks during and after the disaster.

Source (Retrieved on March 25th, 2025): https://sdgs.un.org/goals/goal13#progress_and_info

SDG#13 FLOOD AREA SEGMENTATION FROM IMAGES USING UNET

UNet is highly accurate in pixel-level segmentation with few labeled images in the training set.

UNet may struggle with complex, large-scale objects without additional context modules.

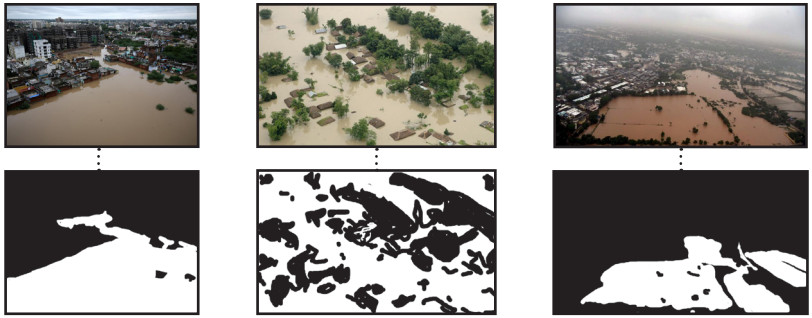
The U-Net architecture is a deep learning model primarily used for image segmentation tasks, where pixel-level classification is required. U-Net can capture both global context and fine spatial details, making it highly effective for medical imaging, satellite or drone imagery, and other pixel-level classification tasks. Its output is a segmentation mask, with pixel-wise predictions for each class in a multi-class or binary segmentation task.

METHOD

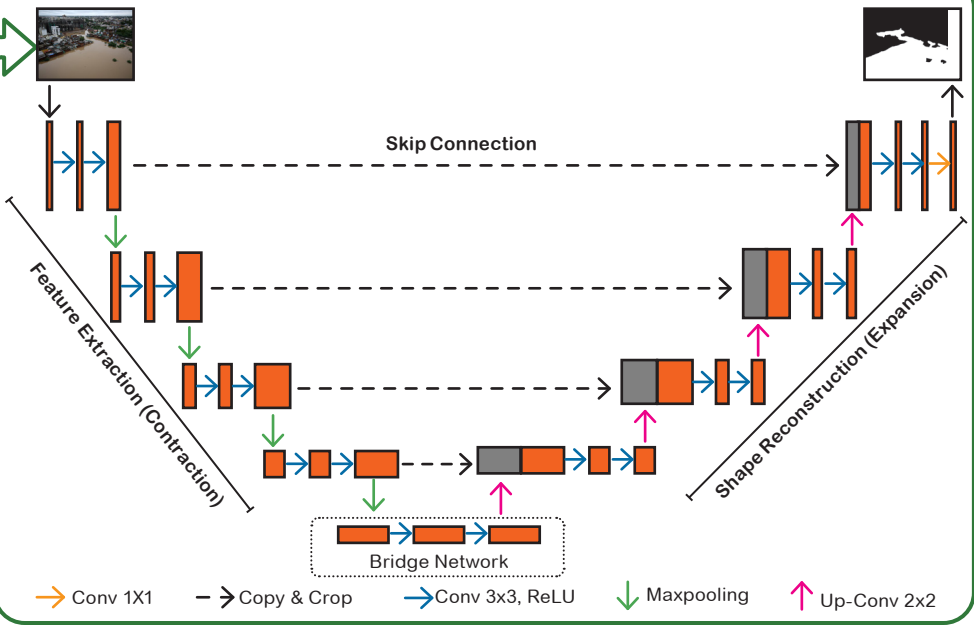
U-Net follows an encoder-decoder structure, designed to capture both local and global features while preserving spatial information. It consists of two main parts: a contracting path (encoder) and an expansive path (decoder). The encoder is typically made up of convolutional layers, followed by max-pooling operations, which progressively downsample the image to extract high-level features. In the encoder, each block consists of two convolutional layers with ReLU activations, followed by a max-pooling layer to reduce spatial dimensions. The bottleneck layer connects the encoder and decoder, where the feature map is at its smallest spatial resolution, capturing the most abstract features of the image. The decoder path upsamples the feature map using transposed convolutions (or deconvolutions), progressively increasing the resolution of the feature map. At each upsampling step, the decoder concatenates the corresponding feature maps from the encoder (via skip connections), which helps retain fine-grained spatial details lost during downsampling. These skip connections allow the model to combine low-level features from the encoder with high-level features from the decoder, enhancing the model's ability to localize precise segmentation boundaries. The final layer of the decoder typically uses a 1x1 convolution to map the feature map to the desired output dimension (e.g., the number of classes for segmentation).

1. INPUT

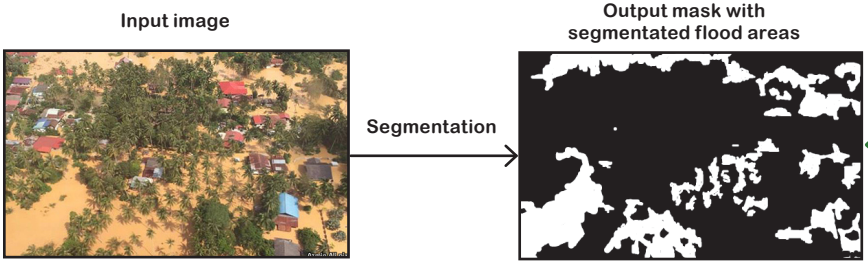
▷ Collection of drone images with their respective masks for training



2. ARCHITECTURE



3. OUTPUT



FURTHER READING

I. Chamatidis, D. Istrati, and N.D. Lagaros, « Vision Transformer for Flood Detection Using Satellite Images from Sentinel-1 and Sentinel-2 », Water, vol. 16, issue 12, no. 1670, 2024.
https://doi.org/10.3390/w16121670

B. Gaffinet, R. Hagensieker, L. Loi, and G. Schumann, « Supervised Machine Learning for Flood Extent Detection with Optical Satellite Data », In Proceedings of IGARSS 2023 IEEE International Geoscience and Remote Sensing Symposium, Pasadena, CA, USA, pp. 2084-2087, 2023.
https://doi.org/10.1109/IGARSS52108.2023.10282274

B. Ghosh, et al., « Automatic Flood Detection from Sentinel-1 Data Using a Nested UNet Model and a NASA Benchmark Dataset », PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science, vol. 92, pp. 1-18, 2024.
https://doi.org/10.1007/s41064-024-00275-1

A. Kazadi, et al., « FloodGNN-GRU: A Spatio-Temporal Graph Neural Network for Flood Prediction », Environmental Data Science, vol. 3, no. e21, 2024.
https://doi.org/doi:10.1017/eds.2024.19

O. Ronneberger, P. Fischer, and T. Brox, « U-Net: Convolutional Networks for Biomedical Image Segmentation », Lecture Notes in Computer Science, vol. 9351. Springer, Cham, 2015.
https://doi.org/10.1007/978-3-319-24574-4_28

Y. Tang, et al., « A Siamese Swin-Unet for Image Change Detection », Sci. Rep., vol. 14, no. 4577, 2024.
https://doi.org/10.1038/s41598-024-54096-8

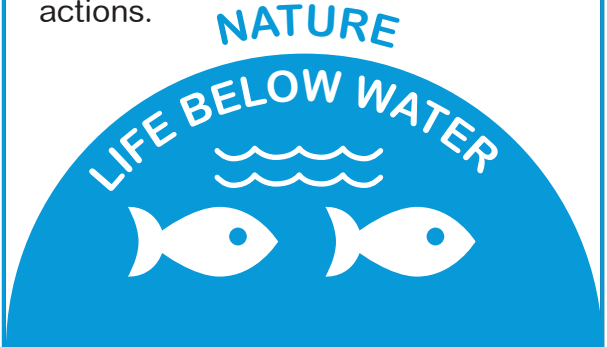
C. Wu, et al., « UNet-Like Remote Sensing Change Detection: A Review of Current Models and Research Directions », IEEE Geoscience and Remote Sensing Magazine, vol. 12, no. 4, pp. 305-334, Dec. 2024.
https://doi.org/10.1109/MGRS.2024.3412770

Scan the QR code to access state-of-the-art research papers, datasets, codes, benchmarks, real-world use cases, and educational materials.

SDG#14 CORAL REEF AUTOMATED ANNOTATION WITH TRANSFER LEARNING

About 25% of all marine species are found in, on, and around coral reefs, rivaling the biodiversity of tropical rainforests. In 2016, heat stress encompassed 51 percent of coral reefs globally. The most recent global bleaching event lasted from 2014 to 2017, with more than 75% mass bleaching-level heat stress of global reefs and nearly 30% mortality-level stress.

Automating coral reef monitoring allows for continuous, large-scale observation of reef health with greater speed and consistency than manual surveys. It enables early detection of threats such as bleaching, pollution, or overfishing, allowing for timely conservation actions.

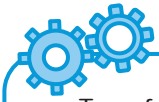


Source (Retrieved on March 25th, 2025): https://coast.noaa.gov/states/fast-facts/coral-reefs.html

SDG#14 CORAL REEF AUTOMATED ANNOTATION WITH TRANSFER LEARNING

Transfer learning enables faster training, better performance, and reduced data requirements for new tasks, when data is scarce.

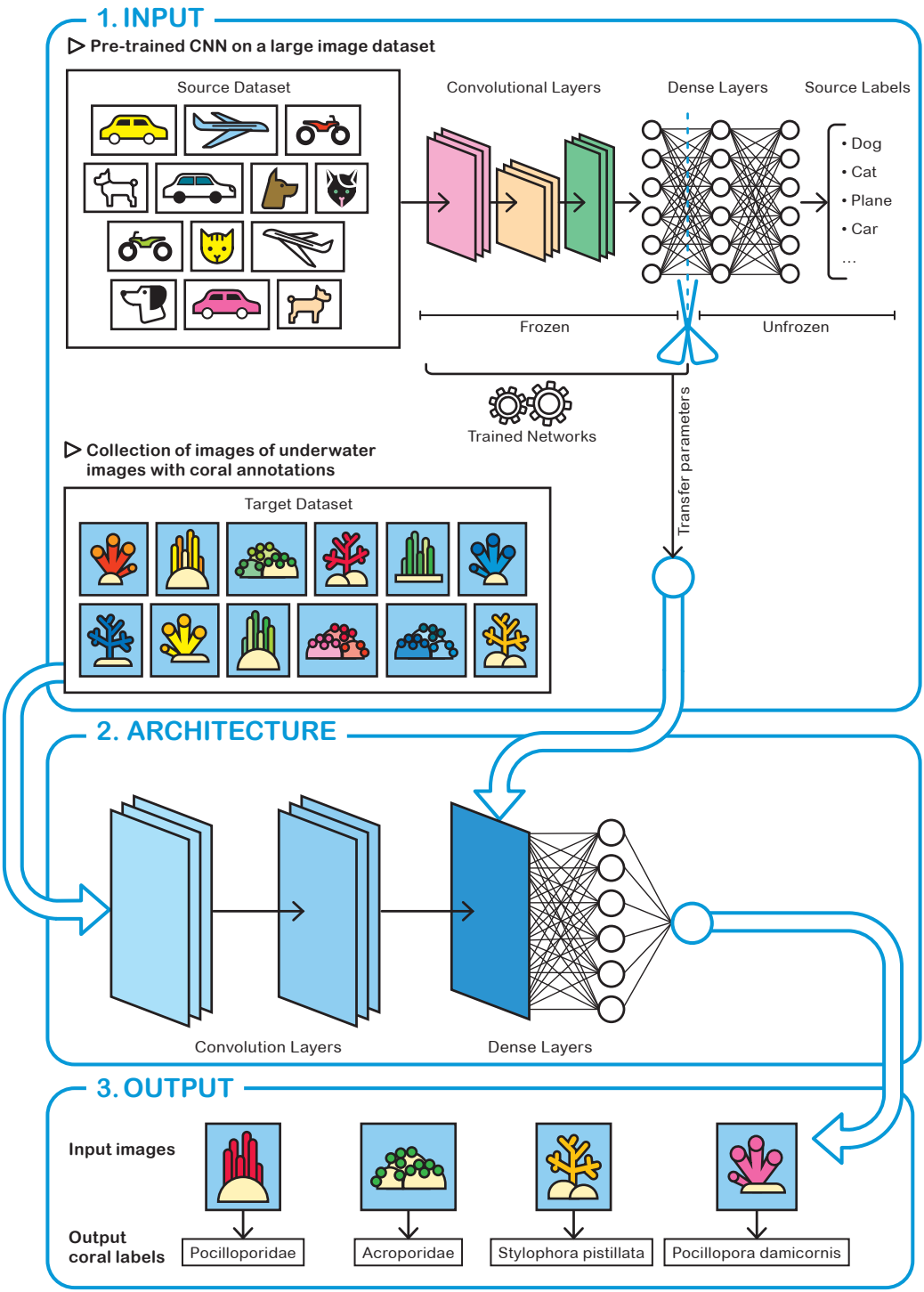
Transfer learning can lead to suboptimal performance if the pre-trained model's domain is too different from the target task.



Transfer learning leverages models pre-trained on large image datasets, requiring fewer labeled coral images to fine-tune the model specific to coral reefs. This approach significantly reduces the manual effort required for coral reef monitoring and analysis, enabling more comprehensive and frequent assessments of these critical ecosystems.

METHOD

Transfer learning relies on using a model pre-trained on a large dataset (e.g., ImageNet) and adapting it to a new, related task with a smaller dataset (e.g., underwater images). The process begins by selecting a pretrained CNN model, such as VGG16, ResNet, or Inception, which has learned general feature representations from a large dataset of images. The initial layers of the pretrained model, which detect basic visual features like edges and textures, are kept fixed (frozen) to preserve their learned representations. The higher layers (closer to the output) of the model are unfrozen and fine-tuned to the new task-specific dataset, allowing the model to adapt to the new task. The feature extraction process begins with passing the input images through the frozen layers of the pretrained network, extracting relevant features from the image. The output from the last layer is passed to a fully connected layer, which is newly added to the architecture for task-specific classification. A softmax activation function is typically applied to the output layer to convert the model's raw output into a probability distribution across different classes. The model is then trained on the target dataset, using a cross-entropy loss function to measure the difference between the predicted class probabilities and the true labels. During fine-tuning, only the weights of the unfrozen layers are updated, allowing the model to specialize in the new task while retaining the general feature representations learned from the large dataset.


1. INPUT
Pre-trained CNN on a large image dataset
Source Dataset
Convolutional Layers
Dense Layers
Source Labels
• Dog
• Cat
• Plane
• Car
...
Frozen
Unfrozen
Trained Networks
Transfer parameters
Collection of images of underwater images with coral annotations
Target Dataset
2. ARCHITECTURE
Convolution Layers
Dense Layers
3. OUTPUT
Input images
Output coral labels
Pocilloporidae
Acroporidae
Stylophora pistillata
Pocillopora damicornis

FURTHER READING

S. Andréfouët, et al., « Choosing the Appropriate Spatial Resolution for Monitoring Coral Bleaching Events Using Remote sensing », Coral Reefs, vol. 21, pp. 147-154, 2002.
https://doi.org/10.1007/s00338-002-0233-x

C. Blondin, J. Guérin, K. Inagaki, G. Longo, and L. Berti-Equille. « Hierarchical Classification for Automated Image Annotation of Coral Reef Benthic Structures », In Proceedings of the NeurIPS 2024 Workshop on Climate Change AI (CCAI 2024), Vancouver, Canada, Dec. 2024.
https://arxiv.org/abs/2412.08228

Q. Chen, et al., « A New Deep Learning Engine for CoralNet », In Proceeding of the 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Montreal, BC, Canada, pp. 3686-3695, 2021.
https://doi.org/10.1109/ICCVW54120.2021.00412

R.R. Gunti and A. Rorissa, « A Convolutional Neural Networks based Coral Reef Annotation and Localization », In Proceedings of Conference and Labs of the Evaluation Forum, 2021.
https://ceur-ws.org/Vol-2936/paper-100.pdf

J.P. Leidig, « Coral Reef Image Collections for Machine Learning, Mapping, and Monitoring », In Proceedings of OCEANS 2022, Hampton Roads, VA, USA, pp. 1-4, 2022.
https://doi.org/10.1109/OCEANS47191.2022.9976984

O. Younes, et al., « Automatic Coral Detection with YOLO: A Deep Learning Approach for Efficient and Accurate Coral Reef Monitoring », In Proceedings of the European Conference on Artificial Intelligence, pp. 170-177, 2023.
https://doi.org/10.1007/978-3-031-50485-3_16

Scan the QR code to access state-of-the-art research papers, datasets, codes, benchmarks, real-world use cases, and educational materials.

SDG#15 ACOUSTIC BIODIVERSITY ASSESSMENT WITH VAE

Around 1 million animal and plant species are now threatened with extinction, many within decades, more than ever before in human history. The average abundance of native species in most major land-based habitats has fallen by at least 20%, mostly since 1900.

Assessing biodiversity with AI enables faster, more accurate identification and monitoring of species across large and complex ecosystems. By automating analysis from images, audio, or environmental data, AI enhances conservation efforts and informs data-driven environmental policies.

Source (Retrieved on March 25th, 2025): https://www.un.org/sustainabledevelopment/blog/2019/05/nature-decline-unprecedented-report/

SDG#15 ACOUSTIC BIODIVERSITY ASSESSMENT WITH VAE

VAEs can reduce signal dimensionality, extract robust features, and enable data generation for data augmentation.

VAEs require careful tuning, making the training process more complex and sensitive to hyperparameter settings.

Variational Autoencoders (VAEs) can assess acoustic biodiversity by first encoding audio spectrograms into a latent space that captures essential acoustic features. By training on diverse species sounds, the VAE learns to disentangle and represent unique acoustic signatures. During inference, the model can classify unseen recordings based on their latent representations, identifying different species or biodiversity indicators.

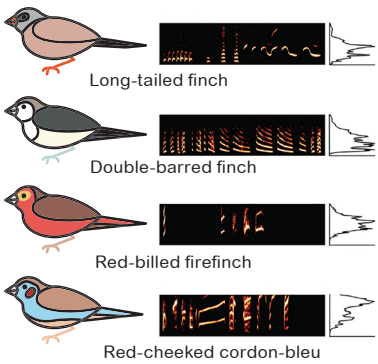
METHOD

A VAE is a generative model designed to learn a probabilistic mapping from input data to a latent space, enabling the generation of new data points similar to the input distribution. It is based on an autoencoder architecture but introduces probabilistic elements for better representation learning and generation. It first compresses input signals into a lower-dimensional latent space (Encoder), generating a mean and variance for the latent variables, thus capturing the signal's characteristics. It samples from the latent space using the mean and variance, allowing gradients to back-propagate and reconstructs the signal from the latent representation (Decoder). The model is trained to minimize the reconstruction loss (e.g., binary cross-entropy or mean squared error) between the original input and the reconstructed data. The VAE also minimizes a KL divergence that measures the difference between the learned latent distribution and a prior distribution (usually a standard Gaussian). The latent representations are fed into a classifier (e.g., a neural network) to predict signal classes. During training, the model learns both to reconstruct the input and to maintain a latent space that approximates the prior distribution, which is typically a standard normal distribution.

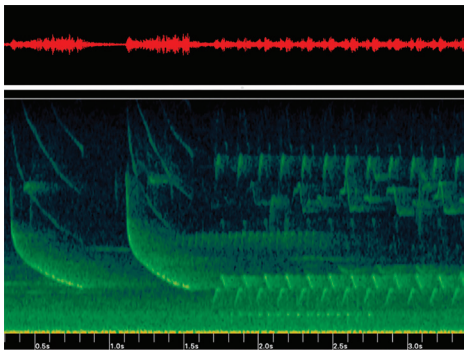
One of the key benefits of the VAE is that it enables unsupervised learning, as it does not require labeled data for training and can model complex data distributions.

1. INPUT

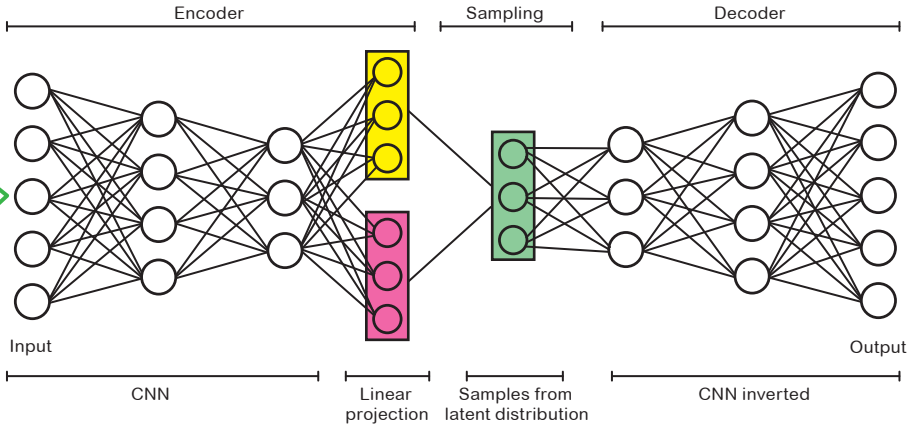
▷ Acoustic signals of identified birds



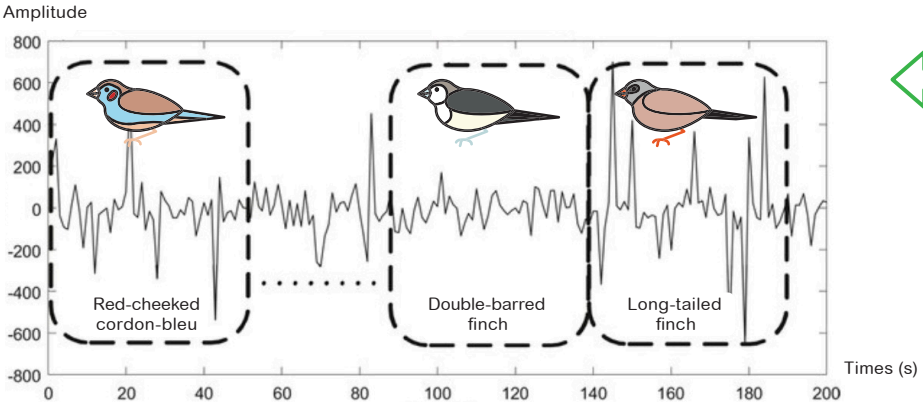
▷ Recording of the natural environment



2. ARCHITECTURE



3. OUTPUT



FURTHER READING

K.A. Gibb, et al., « Towards Interpretable Learned Representations for Ecoacoustics Using Variational Auto-Encoding », Ecological Informatics, vol. 80, 2024.
https://doi.org/10.1016/j.ecoinf.2023.102449

P. Lauha, et al., « Domain-specific Neural Networks Improve Automated Bird Sound Recognition Already With Small Amount of Local Data », Methods in Ecology and Evolution, 2022.
https://doi.org/10.1111/2041-210x.14003

D.A. Nieto-Mora, et al., « Soundscape Characterization Using Autoencoders and Unsupervised Learning », In Proceedings of Italian National Conference on Sensors, 2024.
https://doi.org/10.3390/s24082597

D.A. Nieto-Mora, et al., « Systematic Review of Machine Learning Methods Applied to Ecoacoustics and Soundscape Monitoring », Heliyon, vol. 9, 2023.
https://doi.org/10.1016/j.heliyon.2023.e20275

H. Purohit, et al., « Hierarchical Conditional Variational Autoencoder Based Acoustic Anomaly Detection », In Proceedings of 30th European Signal Processing Conference (EUSIPCO), pp. 274-278, 2022.
https://eurasip.org/Proceedings/Eusipco/Eusipco2022/pdfs/0000274.pdf

J. Ulloa, et al., « scikit-maad: An Open-Source and Modular Toolbox for Quantitative Soundscape Analysis in Python », Methods in Ecology and Evolution, 2021.
https://doi.org/10.1111/2041-210x.13711

Scan the QR code to access state-of-the-art research papers, datasets, codes, benchmarks, real-world use cases, and educational materials.

SDG#15 DETECTING DEFORESTATION USING CNN

Forests play a key role in the mitigation of climate change, removing an estimated 16 billion tons of carbon dioxide (CO_2) from the atmosphere annually. Globally, between 2000 and 2020, forest area declined by 2.4 percent or close to 100 million hectares. In 2020, forests accounted for almost a third of global land area.

Detecting deforestation with AI allows for real-time monitoring of forests using satellite and aerial imagery, enabling faster responses to illegal logging and land degradation. This timely and automated detection is crucial for enforcing environmental laws, protecting biodiversity, and mitigating climate change.

Source (Retrieved on March 25th, 2025): https://datatopics.worldbank.org/sdgatlas/goal-15-life-on-land

SDG#15 DETECTING DEFORESTATION USING CNN

CNN can extract spatial features from images with translation invariance and learn spatial hierarchies of features without manual feature extraction.

Due to limited contextual understanding, CNNs may miss subtle changes in satellite images with complex landscapes.

CNN-based approaches can extract features that are relevant for the detection of deforestation, such as texture, shape, and spectral information, enabling accurate and scalable segmentation of deforested areas from satellite images. CNNs have revolutionized image classification, object detection, segmentation, and video analysis, providing state-of-the-art performance on many visual recognition tasks.

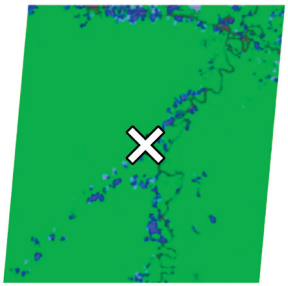
METHOD

A Convolutional Neural Network (CNN) is designed for processing grid-like data, such as images. It is composed of several layers that work together to automatically learn hierarchical feature representations from raw data. The input layer typically consists of an image, which is represented as a 3D tensor (height, width, and color channels, e.g., RGB). The convolutional layers are the core building blocks, where small filters (kernels) slide over the image and perform convolutions to extract local features like edges, textures, and patterns. Each filter in the convolutional layer generates a feature map that captures spatial hierarchies in the image, with deeper layers capturing more complex patterns. After convolution, a non-linear activation function, usually ReLU (Rectified Linear Unit) is applied to introduce non-linearity and enable the network to learn more complex relationships. Pooling layers (typically max pooling) follow convolutional layers to downsample the feature maps, reducing their spatial dimensions while retaining the most important information. Pooling reduces computational load, makes the network invariant to small translations, and helps to prevent overfitting. Fully connected (FC) layers are placed at the end of the network to perform high-level reasoning and classification based on the extracted features. The output of the last fully connected layer is often passed through a softmax activation function (for multi-class classification) to output class probabilities.

1. INPUT

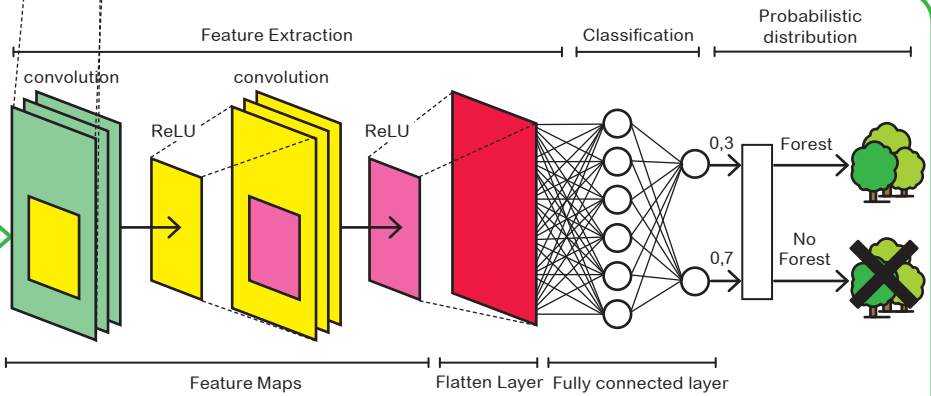
▷ **Sequence of Multispectral Satellite Imagery** Multispectral Satellite Imagery (2014-19) Source: LANDSAT

▷ **Historic annual deforestation**



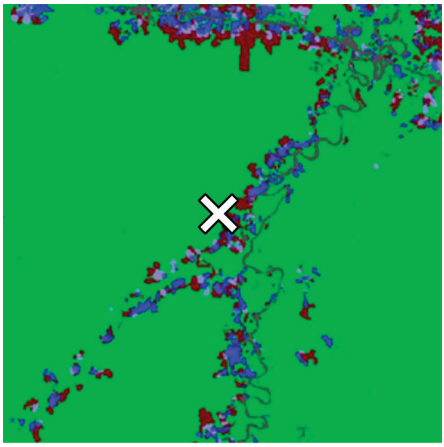
Historical annual deforestation (2000-2019) Source: Hansen, et al. (2012)

2. ARCHITECTURE



3. OUTPUT

For each input image, pixel classification at time t + 1



FURTHER READING

J.S. Almeida, et al., « EdgeFireSmoke: A Novel Lightweight CNN Model for Real-Time Video Fire–Smoke Detection », IEEE Transactions on Industrial Informatics, vol. 18, no. 11, pp. 7889-7898, Nov. 2022. https://doi.org/10.1109/TII.2021.3138752

P.P. de Bem, et al., « Change Detection of Deforestation in the Brazilian Amazon Using Landsat Data and Convolutional Neural Networks », Remote Sensing, vol. 12, no. 6:901, 2020. https://doi.org/10.3390/rs12060901

J.G.C. Ball, et al., « Using Deep Convolutional Neural Networks to Forecast Spatial Patterns of Amazonian Deforestation », Methods in Ecology and Evolution, vol. 13, pp. 2622-2634, 2022. https://doi.org/10.1111/2041-210X.13953

M.C. Hansen, et al., « High-Resolution Global Maps of 21st-Century Forest Cover Change », Science, vol. 342, pp. 850-853, 2013 https://doi.org/10.1126/science.1244693

J. Irvin, et al., « ForestNet: Classifying Drivers of Deforestation in Indonesia using Deep Learning on Satellite Imagery », In Proceedings of NeurIPS 2020 Workshop on Tackling Climate Change with Machine Learning, 2020. https://www.climatechange.ai/papers/neurips2020/22

R.V. Maretto, L.M.G. Fonseca, N. Jacobs, et al., « Spatio-Temporal Deep Learning Approach to Map Deforestation in Amazon Rainforest », IEEE Geoscience and Remote Sensing Letters 18, no. 5, pp. 771-775, May 2021. https://doi.org/10.1109/LGRS.2020.2986407

M.X. Ortega, et al., « Evaluation of Deep Learning Techniques for Deforestation Detection in the Amazon Forest », ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2019. https://isprs-annals.copernicus.org/articles/IV-2-W7/121/2019/

Scan the QR code to access state-of-the-art research papers, datasets, codes, benchmarks, real-world use cases, and educational materials.

SDG#16 PREDICTION OF SOCIAL CONFLICTS WITH GNN

Today, almost a billion people live in fragile and conflict-affected situations. In 2022, civilians across the world faced more than 116,000 violent events, a third of them in Ukraine.

Detecting social conflicts with AI enables early identification of rising tensions through the analysis of news, social media, and other data sources. This allows governments and organizations to intervene proactively, potentially preventing violence and reducing harm. By providing real-time insights into conflict dynamics, AI supports informed decision-making for peacebuilding and crisis management.

Source (Retrieved on March 25th, 2025): https://datatopics.worldbank.org/sdgatlas/goal-16-peace-justice-and-strong-institutions?lang=en

SDG#16 PREDICTION OF SOCIAL CONFLICTS USING GNN

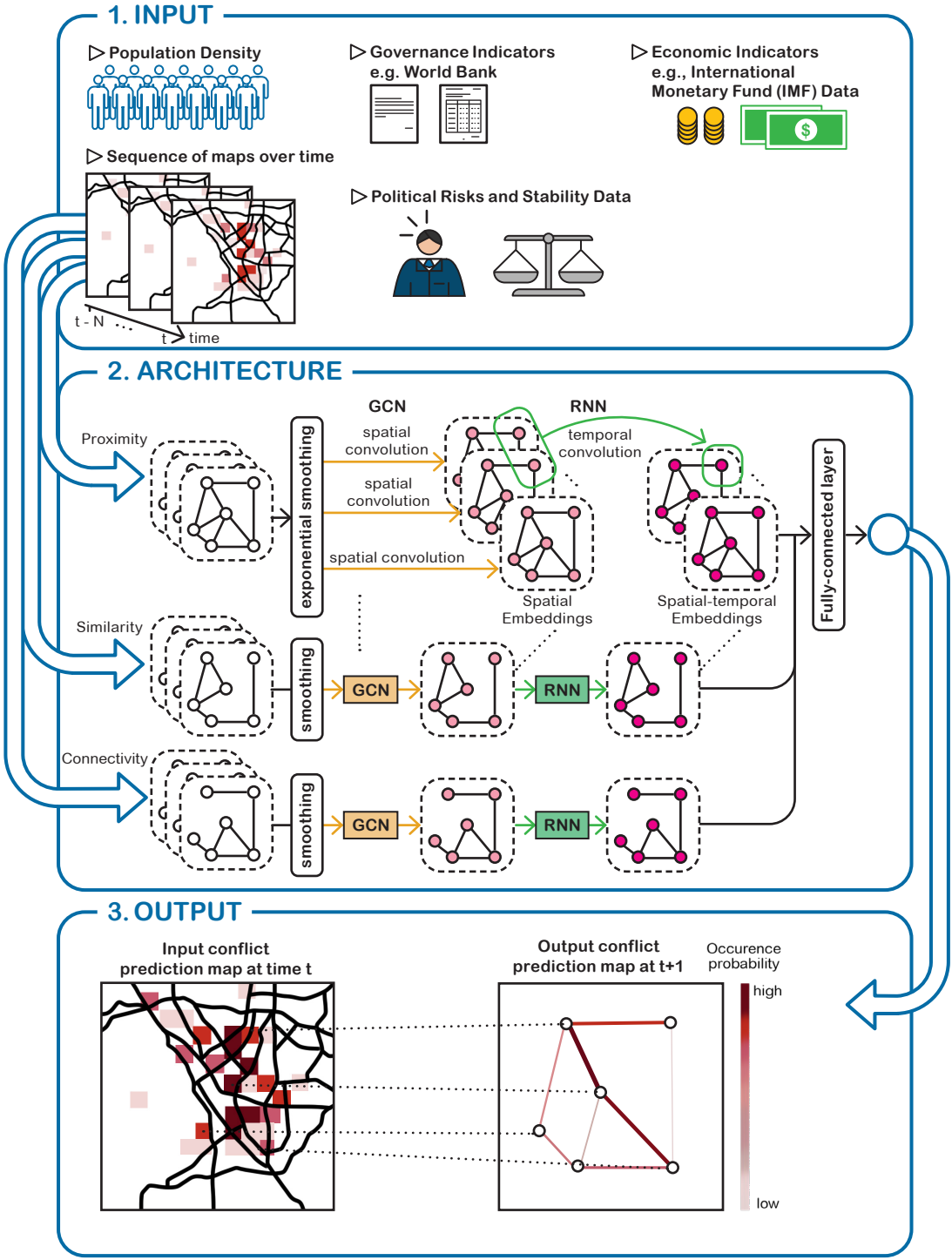
GNNs can capture complex spatial relationships and temporal dependencies or intricacies and scale robustly to large networks.

GNNs can be computationally intensive, require large, high-quality labeled datasets and have limited expressivity. for complex structures.

GNNs with spatial embeddings can effectively model non-Euclidean spatial data, representing the complex geographical and social relationships between regions or actors involved in social conflicts. Temporal embeddings allow GNNs to capture time-dependent patterns and evolving trends. This allows the prediction of conflict likelihood based on learned representations of the evolving social dynamics and network structure.

METHOD

A Graph Neural Network (GNN) is a type of deep learning model designed to handle graph-structured data, where nodes represent entities, and edges represent relationships between them. GNNs aim to learn node or graph-level representations by propagating information through the graph's structure. The input to a GNN typically consists of a graph with nodes and edges, where each node has a feature vector representing its properties. GNNs operate by iteratively updating the node representations through message passing between neighboring nodes based on the graph's connectivity. In each layer of the GNN, nodes aggregate information from their neighbors and update their feature vector using a neighborhood aggregation function (such as mean, sum, or max). The aggregation step combines the features of a node's neighbors to capture local graph structure, while the update function refines each node's feature vector. This process is repeated across multiple layers, allowing nodes to incorporate information from progressively larger neighborhoods in the graph. After several layers of message passing, the final node representations are typically used for node-level tasks (e.g., node classification) or aggregated for graph-level tasks (e.g., graph classification). The GNN can also use a readout function to pool information across all nodes in a graph, which is particularly useful for graph-level predictions.


1. INPUT
Population Density
Governance Indicators
e.g. World Bank
Economic Indicators
e.g., International
Monetary Fund (IMF) Data
Sequence of maps over time
t - N
t
time
Political Risks and Stability Data
2. ARCHITECTURE
GCN
RNN
Proximity
exponential smoothing
spatial convolution
spatial convolution
spatial convolution
temporal convolution
Spatial Embeddings
Spatial-temporal Embeddings
Fully-connected layer
Similarity
smoothing
GCN
RNN
Connectivity
smoothing
GCN
RNN
3. OUTPUT
Input conflict prediction map at time t
Output conflict prediction map at t+1
Occurence probability
high
low

FURTHER READING

Z. Binbin, et al., « HDM-GNN: A Heterogeneous Dynamic Multi-view Graph Neural Network for Crime Prediction », ACM Trans. Sen. Netw., May 2024. https://doi.org/10.1145/3665141

F. Ettensperger, « Comparing Supervised Learning Algorithms and Artificial Neural Networks for Conflict Prediction: Performance and Applicability of Deep Learning in the Field », Qual. Quant., vol. 54, pp. 567-601, 2020. https://doi.org/10.1007/s11135-019-00882-w

G. Jin, et al., « Spatio-Temporal Graph Neural Networks for Predictive Learning in Urban Computing: A Survey », IEEE Transactions on Knowledge and Data Engineering, vol. 36, no. 10, pp. 5388-5408, Oct. 2024. https://doi.org/10.1109/TKDE.2023.3333824

Z.A., Sahili, and M. Awad, « Spatio-Temporal Graph Neural Networks: A Survey », arXiv:2301.10569, 2024. https://arxiv.org/abs/2301.10569

L. Yuan, et al., « Prediction of Airport Surface Potential Conflict Based on GNN-LSTM », IET Intell. Transp. Syst., vol. 19, no. e12611, 2025. https://doi.org/10.1049/itr2.12611

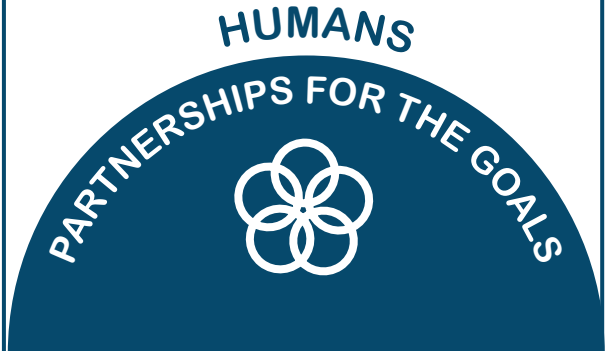
K. Bluwstein, et al., « Credit Growth, the Yield Curve and Financial Crisis Prediction: Evidence from a Machine Learning Approach », Bank of England, Working Paper, no. 848, January 2020. http://dx.doi.org/10.2139/ssrn.3520659

Scan the QR code to access state-of-the-art research papers, datasets, codes, benchmarks, real-world use cases, and educational materials.

SDG#17 CLIMATE AGREEMENT NEGOTIATION WITH MARL

The surge in aid from 2019 to 2022 was driven by extraordinary spending related to the COVID-19 pandemic and the war in Ukraine with a record $211.3 billion. But during this period, less aid (-1.2% Official Development Assistance) has been allocated for activities not related to the pandemic and the war in Ukraine.

AI agents can simulate negotiation scenarios, balance competing interests, and explore win-win outcomes among diverse stakeholders. They can process complex climate, economic, and policy data to recommend fair and effective solutions in real time.



Source (Retrieved on March 25th, 2025): https://datatopics.worldbank.org/sdgatlas/goal-17-partnerships-for-the-goals?lang=en

SDG#17 CLIMATE AGREEMENT NEGOTIATION WITH MARL

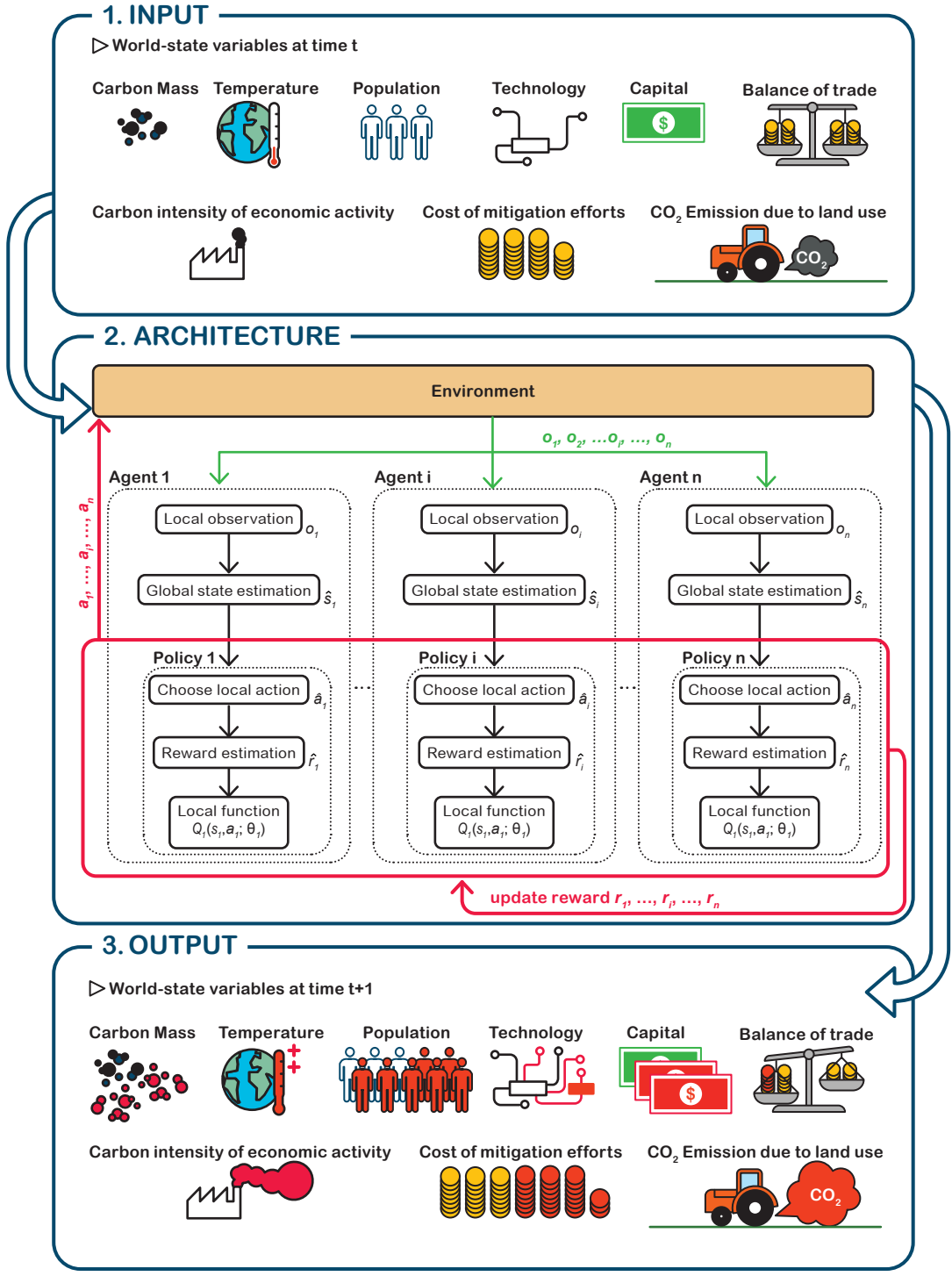
MARL enables efficient problem-solving through agent collaboration and competition in dynamic environments.

MARL training can become unstable due to non-stationary and changing environments, as other agents adapt and learn.

Multi-Agent Reinforcement Learning (MARL) enables agents to solve complex, real-world problems by leveraging cooperation, competition, and coordination, making it ideal for tasks involving multiple decision-makers in dynamic environments. Communication protocols can be used in some MARL setups, allowing agents to exchange information about their states and actions to improve coordination in cooperative environments.

METHOD

MARL is a framework in which multiple agents learn to make decisions and interact with each other in a shared environment, each aiming to maximize its reward while considering the actions of other agents. It is an extension of traditional reinforcement learning (RL) to multi-agent settings. In MARL, each agent has its policy that dictates how it behaves based on its observations of the environment and possibly other agents. State space in MARL refers to the collective states of all agents and the environment, while action space is the set of all possible actions each agent can take individually. Each agent receives feedback in the form of rewards from the environment, and the goal is to maximize the expected sum of rewards, often through value-based, policy-based, or actor-critic methods. Agents must consider the behavior of other agents when making decisions, which can lead to cooperative, competitive, or mixed strategies, depending on the task. In cooperative MARL, all agents share a common goal (e.g., maximizing a team's total reward) and may share information about their states and actions. In competitive MARL, agents work against each other (e.g., in games like chess or poker), where the goal is to maximize individual rewards at the expense of others. Centralized training with decentralized execution is a common paradigm, where agents are trained with global information but act based on local observations during execution.


1. INPUT
▷ World-state variables at time t
Carbon Mass
Temperature
Population
Technology
Capital
Balance of trade
Carbon intensity of economic activity
Cost of mitigation efforts
CO2 Emission due to land use
2. ARCHITECTURE
Environment
o1, o2, ...oi, ..., on
a1, ..., ai, ..., an
Agent 1
Agent i
Agent n
Local observation o1
Local observation oi
Local observation on
Global state estimation ŝ1
Global state estimation ŝi
Global state estimation ŝn
Policy 1
Policy i
Policy n
Choose local action â1
Choose local action âi
Choose local action ân
Reward estimation r̂1
Reward estimation r̂i
Reward estimation r̂n
Local function Q1(s1,a1; θ1)
Local function Qi(si,ai; θi)
Local function Qn(sn,an; θn)
update reward r1, ..., ri, ..., rn
3. OUTPUT
▷ World-state variables at time t+1
Carbon Mass
Temperature
Population
Technology
Capital
Balance of trade
Carbon intensity of economic activity
Cost of mitigation efforts
CO2 Emission due to land use

FURTHER READING

K. Georgila, C. Nelson, and D. Traum, « Single-Agent vs. Multi-Agent Techniques for Concurrent Reinforcement Learning of Negotiation Dialogue Policies », In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, vol. 1, pp. 500-510, Baltimore, MA, USA, 2014. https://aclanthology.org/P14-1047.pdf

T. Gu, T. Zhi, X. Bao, and L. Chang, « Credible Negotiation for Multi-agent Reinforcement Learning in Long-term Coordination », ACM Trans. Auton. Adapt. Syst., vol. 20, issue 1, no. 1, March 2025. https://doi.org/10.1145/3706110

T. Li, et al., « Applications of Multi-Agent Reinforcement Learning in Future Internet: A Comprehensive Survey », IEEE Communications Surveys & Tutorials, vol. 24, no. 2, pp. 1240-1279, 2022. https://doi.org/10.1109/COMST.2022.3160697

Z. Liu, et al., « Graph Neural Network Meets Multi-Agent Reinforcement Learning: Fundamentals, Applications, and Future Directions », IEEE Wireless Communications, vol. 31, no. 6, pp. 39-47, December 2024. https://doi.org/10.1109/MWC.015.2300595

C. Sun, S. Huang, and D. Pompili, « LLM-based Multi-Agent Reinforcement Learning: Current and Future Directions », arXiv:405.11106, 2024. https://arxiv.org/abs/2405.11106

L. Yuan, et al., « A Survey of Progress on Cooperative Multi-agent Reinforcement Learning in Open Environment », arXiv:2312.01058, 2023. https://arxiv.org/abs/2312.01058

Scan the QR code to access state-of-the-art research papers, datasets, codes, benchmarks, real-world use cases, and educational materials.

GLOSSARY

Ablation Study: An experimental approach in machine learning where specific components or features of a model are systematically removed or altered to assess their impact on performance.

Activation: Mathematical function applied to a neuron's output that introduces non-linearity into the model, enabling neural networks to learn complex relationships and patterns in data.

Non-Linear Activation: Function applied to a neuron's output that introduces non-linearity into the model, allowing it to learn complex patterns (e.g., ReLU, tanh).

ELU (Exponential Linear Unit): Activation function that smooths the negative part of the input using an exponential curve, helping improve learning by allowing small negative outputs.

GELU (Gaussian Error Linear Unit): Activation function that weights inputs by their value and probability under a normal distribution, used in Transformer models like BERT for smoother learning.

Leaky ReLU: Variant of ReLU that allows a small, non-zero gradient for negative input values, helping to mitigate the dying neuron problem.

Parametric ReLU (PReLU): Adaptive version of Leaky ReLU where the slope of the negative part is learned during training, improving model flexibility.

ReLU (Rectified Linear Unit): Activation function that outputs the input if it is positive and zero otherwise, widely used for its simplicity and effectiveness in deep networks.

SELU (Scaled Exponential Linear Unit): Scaled version of ELU designed to self-normalize the activations of neurons, maintaining a mean and variance close to zero and one, respectively.

Sigmoid: Activation function that maps any real-valued input to a value between 0 and 1, commonly used in binary classification tasks to produce probabilities.

Softmax: Activation function that transforms a vector of real numbers into a probability distribution, used in the output layer for multi-class classification tasks.

Tanh: Activation function that maps inputs to a range between -1 and 1, offering zero-centered output which can be beneficial for learning dynamics.

Adam: An optimization algorithm that computes adaptive learning rates for each parameter by considering both the first and second moments of the gradients, enhancing convergence speed and stability.

Adversarial Training: A technique in machine learning where models are trained on adversarial examples—inputs intentionally modified to mislead the model—to improve robustness and resistance to such attacks.

Backpropagation: A supervised learning algorithm for training neural networks by propagating the error backward through the network, adjusting weights to minimize the loss function.

Bagging: A machine learning ensemble technique that trains multiple models (usually of the same type) on different subsets of the training data and combines

their predictions to improve accuracy and reduce variance.

Balanced Dataset: A dataset where each class or category is represented equally, preventing the model from being biased toward the majority class.

Batch Normalization: A technique to normalize the inputs of each layer in a neural network, improving training speed and stability by reducing internal covariate shift.

Bias: A parameter in a neural network that allows the model to make predictions even when all input features are zero, enabling the model to fit the data more flexibly.

Boosting: An ensemble learning method that combines multiple weak learners to create a strong learner, typically by focusing on correcting the errors of previous models.

Cell State: Internal memory component of the LSTM cell that can carry information over long sequences. It is controlled by gates to regulate what information should be added or removed from it.

Complexity: In machine learning, it refers to the capacity of a model to capture intricate patterns in data; higher complexity can lead to overfitting if not properly managed.

Contrastive Feature: A characteristic or attribute in data that highlights differences between classes, aiding in distinguishing between them.

Contrastive Learning: A self-supervised learning approach where models learn by comparing similar and dissimilar pairs of data, encouraging the model to learn useful representations.

Convergence: The process where a machine learning algorithm's performance stabilizes, indicating that further training will not significantly improve results.

Convolution: A mathematical operation used in convolutional neural networks (CNNs) to extract features from input data by applying a filter or kernel over it.

> **Atrous Convolution:** A convolution operation that introduces gaps between kernel elements, allowing the network to capture multi-scale context without increasing the number of parameters.
>
> **Depthwise Convolution:** A type of convolution where each input channel is convolved with its own set of filters, reducing the number of parameters and computation compared to standard convolutions.
>
> **Pointwise Convolution:** Type of convolution that uses 1×1 kernels to transform the number of channels in the input without affecting spatial dimensions.
>
> **Separable Convolution:** Efficient form of convolution that splits the process into depthwise and pointwise convolutions to reduce computation and parameters.

Cross-Entropy: A loss function commonly used in classification tasks that measures the difference between two probability distributions, typically the true labels and the predicted probabilities.

Cross-Product Transformation: A mathematical operation that combines two vectors to produce a third, often used in tasks like computing attention scores in neural networks.

Decoder: A component in models like sequence-to-sequence architectures that generates output sequences from encoded representations, such as in machine translation.

Deconvolution: Also known as transposed convolution, it's an operation used to upsample data, often used in tasks like

image segmentation to increase spatial resolution.

Dense Layer: A fully connected layer in a neural network where each neuron is connected to every neuron in the previous layer, enabling complex representations.

Discriminator: In Generative Adversarial Networks (GANs), a model that distinguishes between real and generated data, guiding the generator to produce more realistic outputs.

Down-Sampling: The process of reducing the spatial dimensions of data, typically to decrease computational load and capture broader context.

Dropout: A regularization technique where randomly selected neurons are ignored during training, preventing overfitting by ensuring the model doesn't rely on specific neurons.

Embedding: A technique to represent discrete variables, like words, as continuous vectors in a lower-dimensional space, capturing semantic relationships.

Encoder: A component in models like sequence-to-sequence architectures that processes input sequences into a fixed-size context vector, which is then used by the decoder.

Ensemble: A method that combines multiple models to improve overall performance, often by reducing variance and bias.

Entropy Maximization: A strategy in machine learning where the model is encouraged to make predictions with high uncertainty, often used in semi-supervised learning to explore data distributions.

Epsilon-Greedy: A policy in reinforcement learning where the agent mostly chooses the best-known action but occasionally selects a random action to explore the environment.

Error Analysis: The process of examining the types and sources of errors in a model's predictions to identify areas for improvement.

Feature: An individual measurable property or characteristic of a phenomenon being observed, used as input to machine learning models.

Feedforward Layer: A layer in a neural network where connections between the nodes do not form cycles, allowing data to flow in one direction from input to output.

Flattening: The process of converting multi-dimensional data into a one-dimensional vector, often used before feeding data into fully connected layers.

Forget Gate: Group of mathematical operations responsible for deciding which information from the previous cell state should be discarded or forgotten based on the current input and the previous hidden state.

Gating Mechanism: Group of mathematical operations used in neural network architectures such as LSTMs and GRUs to control the flow of information, deciding what should be passed on, updated, or forgotten.

Generalization: Model's ability to perform well on unseen data by capturing the underlying patterns rather than memorizing the training set.

Generator: Component of a Generative Adversarial Network (GAN) responsible for creating synthetic data samples intended to resemble the real data.

Gradient: Vector of partial derivatives indicating the direction and rate of fastest increase of a function, used during

training to update model weights via backpropagation.

Hidden State: Memory of a recurrent cell. It stores temporal information from previous time steps and is passed along the sequence to influence future predictions.

Hyperparameter: Configuration variable that is set before training a model (e.g., learning rate, number of layers) and governs the training process and structure of the model.

Inference: Phase in which a trained model is used to make predictions on new, unseen data without updating its parameters.

Input Gate: Group of mathematical operations that determines which new information from the current input and the previous hidden state (h) should be added to the cell state.

Interpolation: Process of estimating intermediate values between two known data points, often used in data augmentation or image resizing.

Inverted CNN: Architecture where convolutional layers are applied in reverse to upsample feature maps, commonly used in image generation and segmentation tasks.

IoT (Internet of Things): Network of physical objects—"things"—that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet.

Kernel Function: Mathematical function used to compute similarity between data points in high-dimensional space without explicitly mapping them, enabling non-linear classification.

KL (Kullback-Leibler) Divergence: Statistical distance that measures how much a model's probability distribution is different from a true probability distribution.

Kernel Trick: Technique in machine learning that applies a kernel function to compute inner products in a high-dimensional space without explicitly performing the transformation.

Label: Target output or class assigned to a data point, used during supervised learning to guide model predictions.

Latent Space: Abstract feature space where high-dimensional data is represented in a compressed and meaningful way, often learned by autoencoders or GANs.

Layer: Building block of neural networks consisting of a set of neurons that process input data and pass output to the next layer in the network.

Layer Normalization: Technique that normalizes the inputs across the features of a layer, stabilizing and speeding up training of deep neural networks.

Logit: Raw output value of a model's final layer before applying an activation function like softmax or sigmoid, typically used in classification tasks.

Loss Function: Mathematical function that quantifies the difference between predicted outputs and true targets, guiding the optimization of the model.

> **InfoNCE Loss:** Loss function used in contrastive learning that encourages similar samples to have similar representations while dissimilar samples are pushed apart in the embedding space.
>
> **Reconstruction Loss:** Loss function that measures the difference between the original data and its reconstructed version, commonly used in autoencoders.

Memory Cell or Unit: Core component of LSTM networks that retains long-term

dependencies by selectively adding and removing information through gating mechanisms.

Meta-Learner: Higher-level model trained to learn how to optimize other models or learning tasks, commonly used in meta-learning or few-shot learning.

Model Collapse: Failure mode in training GANs where the generator produces limited or identical outputs, reducing diversity and usefulness of generated data.

Model Performance Metrics: Quantitative measures (e.g., accuracy, precision, recall) used to evaluate the effectiveness of a model on specific tasks or datasets.

MSE: Mean Squared Error. A regression loss function that calculates the average of the squares of differences between predicted and true values.

Multi-Head Self-Attention: Mechanism that allows a model to jointly attend to information from different representation subspaces at different positions in the sequence.

NLP: Natural Language Processing. A field of AI focused on enabling computers to understand, interpret, and generate human language.

Output Gate: Group of mathematical operations that regulates what information from the updated cell state should be included in the current hidden state (h), which will be passed to the next time step in the sequence.

Overfitting: Condition in which a model learns noise and details from the training data to the extent that it performs poorly on new, unseen data.

Patch: Subsection or region of an input image or data sample used for localized processing in tasks like vision transformers or convolutions.

Pattern: Repeated or recognizable structure in data that a model attempts to learn and generalize during training.

Policy: Strategy used by an agent in reinforcement learning to decide which action to take based on the current state of the environment.

Pooling (Max / Average): Operation used in convolutional networks to reduce spatial dimensions by summarizing local regions, commonly through max or average functions.

Pretrained Network: Neural network model that has already been trained on a large dataset and can be fine-tuned for specific tasks to improve efficiency and performance.

Prior Distribution: Assumed probability distribution over model parameters or latent variables before observing any data, used in Bayesian methods.

Readout Function: Component in neural networks that maps internal representations to output predictions, often used in graph neural networks or recurrent architectures.

Regularization: Set of techniques (e.g., L1/L2 penalty, dropout) used to prevent overfitting by penalizing complex models or reducing reliance on specific features.

Residual Connection: Shortcut path in deep networks that adds the input of a layer directly to its output, enabling the training of very deep architectures.

Reward: Signal received by an agent in reinforcement learning indicating how good an action was in a given state, guiding future behavior.

Sampling: Process of selecting a subset of data from a larger dataset or drawing

data points from a probability distribution for training or generation.

Downsampling: Technique used to decrease the number of data samples by removing data from the majority class, often used to correct imbalanced datasets.

Upsampling: Technique used to increase the resolution or number of data, often by inserting values or interpolating between existing data points.

Scalability: Model or system's ability to efficiently handle increased data, complexity, or computational load without significant performance degradation.

Scaling: Transformation that adjusts the size or range of data values, often used to standardize features before training.

Self-Attention: Mechanism that allows a model to focus on different parts of a single sequence when computing representations, crucial in transformer architectures.

> **Masked Self-Attention or attention Masking:** A method used in attention mechanisms to prevent certain positions in the input sequence from contributing to the output, often used in tasks like language modeling to maintain causality.

SGD: Stochastic Gradient Descent. Optimization algorithm that updates model parameters using a subset (mini-batch) of the data at each iteration to speed up learning.

Skip Connection: Shortcut pathway that bypasses one or more layers in a network, facilitating gradient flow and helping mitigate vanishing gradient problems.

Sliding Window: Technique for processing data in overlapping or non-overlapping chunks, useful for sequence modeling and object detection.

Stacking: Ensemble method that combines the outputs of multiple models using a meta-model, leveraging their strengths for improved prediction.

Tensor: Multi-dimensional array of numerical values used to represent data in machine learning models, supporting operations across various dimensions.

Testing Set: Subset of data reserved for evaluating a trained model's performance on unseen examples, ensuring generalization.

Token: Smallest unit of text (word, character, or subword) that is processed by NLP models during training and inference.

Training Set: Portion of the dataset used to fit and train the model by minimizing the loss function and adjusting parameters.

Tuning: Process of adjusting hyperparameters to improve model performance, often performed using grid search, random search, or Bayesian optimization.

Validation Set: Subset of the dataset used during training to monitor performance and tune hyperparameters without affecting the final test evaluation.

Vanishing Gradient: Problem in deep neural networks where gradients become too small during backpropagation, leading to extremely slow learning in earlier layers.

Variance: Measure of how much model predictions fluctuate for different training datasets; high variance indicates potential overfitting.

Vector Representation: Numerical encoding of data (e.g., words, images) in a fixed-length vector format that captures its essential features for model input.

Weight: Trainable parameter in a neural network that determines the strength of

the connection between neurons, updated during training to minimize loss.

XGBoost: Extreme Gradient Boosting: A scalable, distributed gradient-boosted decision tree (GBDT) open source machine learning library.

Zero-Shot Learning: Learning paradigm where a model is able to make predictions on classes it has never seen before by leveraging semantic or descriptive information.

First Edition: September 2025