PLOMS Journal of
Artificial Intelligence
(PLOMS AI)

PLOMS AI

Review Paper

Artificial Intelligence Tools and Their Capabilities

Harisu Abdullahi Shehu[1], Rabie A, Ramadan[2,,3,*] and Md. Haidar Sharif [2]

[1]School of Engineering and Computer Science, Victoria University of Wellington, Wellington 6012, New Zealand.

[2]College of Computer Science and Engineering, University of Hail, Hail 2440, Saudi Arabia

[3]Computer Engineering Department, Cairo University, Cairo 12613, Egypt.

Correspondence: rabie@rabieramadan.org

Received: 01-06-2021; Accepted: 10-06-2021; Published: 14-06-2021

Abstract: Artificial intelligence (AI) is a scientific discipline that focuses on computing models simulating the human brain for problem-solving. AI is the science of helping computers to achieve things that humans do better. Intelligent agents are also called Artificial Intelligence. Some of the major AI issues are reasoning, knowledge, planning, learning, communication, perception, and movement. This article aims to explore artificial intelligence tools that can be used to implement any AI technique. It compares these tools, their advantages, their disadvantages, and their capabilities. The authors believe that the paper could be of interest of researchers from different fields to select the suitable tool(s) for their research problems.

Keywords: Artificial Intelligence; AI Tools; AI; Explainable AI; AI Applications.

1. Introduction

Artificial Intelligence is the most in-demand subject in computer science since it is concerned with the computer simulation of human intelligent behavior. AI approaches are easily identifiable as it is in many of the characteristics of the commercial product. These techniques operate in the background, enhancing the system's overall performance. AI may be utilized in conjunction with software APIs and user interfaces. It uses machine learning to map between model inputs and parallel outputs for accessible data by continually providing model inputs and outputs instances. It chooses the optimal model that delivers the desired outcomes within the given budget and training data constraints. AI analyzes systems performance and helps businesses in establishing client confidence.

Problem-solving ability demonstrates intelligence. Consider a mouse attempting to locate the piece of cheese located in the top-right corner of the picture; intelligence could develop more than one solution to the situation. There are many types and degrees of intelligence in humans, animals, and machines. For example, AI provides critical answers when it comes to cancer therapy and food safety assurance for a growing population. AI is also significant in strategic games such as poker, chess, etc., where it uses heuristic knowledge to choose among several alternative positions. Interactions with computers that comprehend normal human languages are also possible with AI. AI can recognize distinct accents, background noise, and changes in a user's speech due to health problems. A huge benefit of AI is utilized in detecting fraud. For instance, master cards utilize intelligence in detecting fraud, boost real-time accuracy, and reduce false failures.

AI technologies are used in many of our daily life applications. They are able to recognize and transform the handwritten text into machine-readable text. It is also able to analyze and interpret visual information. AI is employed as a decision support system for the clinics to provide diagnostic information. Fuzzy logic controllers have been developed for automatic gearboxes in automobiles. Home water quality monitoring applications are built using artificial intelligence. AI strives to explain, by computing, all of the human intellect. It is able to interact with the surroundings through sensors, and it has AI computer-generated thinking. Intelligence may be seen as a trait that can be separated from all other qualities.

Explainable AI (XAI) is a growing research topic, as the demand to explain and argue for ideas and laws continues to expand. Explanation systems were initially created in the early 1980s to explain

the conclusions of expert systems. Later, human-computer systems, such as intelligent tutoring systems, changed to assist human cognition better. In this situation, the 'black box' does not have a positive reputation owing to the model's lack of explainability and transparency. Most authors use "explainability" and "interpretability" interchangeably. When it comes to an intelligent system (i.e., AI-based system), the logical conclusion is that explainability is greater than interpretability.

This paper is a step forward towards AI tools, their properties, advantages, and disadvantages. The paper is organized as follows: in the next section, some related terminologies are presented, followed by the AI tools explained in section III. Finally, the paper is concluded in section IV.

2. AI Related Terminology

It is critical to differentiate between many terminologies that have been used interchangeably in the literature. Therefore, the following pertinent terms: (1) validation, (2) verification, (3) testing, and (4) assurance are defined.

1. ***Verification***: It is the process of verifying that the results of a specific development phase fulfill the standards established at the start of that phase.
2. ***Validation*****:** It is the process of determining if a system or its components satisfy certain specifications in the design and development stages.
3. ***Testing*****:** It is defined in all of the lifecycle activities, both static and dynamic, that are involved with planning, preparation, and assessment of software products and associated work to make sure they fulfill specific requirements.

I. AI Tools and Platforms

To help researchers and industry, many tools and platforms are developed. In the following subsections, AI tools and platforms are reviewed. **Table 1** shows the pros and cons of the reviewed tools.

A. Scikit Learn

The popularity of Scikit-learn makes it one of the most well-known machine learning libraries. Many forms of supervised and unsupervised learning algorithms are based on it. When looking at precedents, we should think about both direct and calculated relapses, as well as choice trees, clustering, and K-implies. For those things like accurate picture categorization and reliable real-time language parsing and translation, which are crucial but complex tasks, Scikit-learn does not provide solutions. Also, it cannot be used in languages other than Python. It cannot support PyPy or GPUs, and it doesn't feature graphical models or sequence prediction. For machine learning other than neural networks, the machine learning support included in Scikit-learn works well even without the assistance of GPUs or PyPy. Python, especially when Cython is used to create C code for inner loops, may sometimes be faster than people think, and Cython combined with Scikit-learn helps keep Scikit-learn from becoming a bottleneck. **Figure 1** shows the Scikit-learn properties.

B. Tensorflow [2]

TensorFlow is utilized as a programming environment for deep learning algorithms, specifically for the following purposes:

1. Recommendations: Picking the best templates to suggest with 100 languages supported.
2. Classification: When users offer input, natural language processing algorithms written in TensorFlow are utilized to categorize concerns, allowing stakeholders to quickly discover the most critical concerns with a product/release.
3. Learning-to-rank: There has been some progress in improving search results by moving from gradient boosting to deep learning methods, and TensorFlow enables this.
4. Computer vision: Several experiments on object recognition and picture categorization were conducted.

InfoWorld Scorecard	Models and Algorithms (25%)	Ease of Development (25%)	Documentation (20%)	Performance (20%)	Ease of Deployment (10%)	Overall Score (100%)
Scikit-Learn. 0.18.1	9	9	9	8	9	8.8

Figure 1. Scikit-learn properties [https://www.infoworld.com/article/3158509/review-scikit-learn-shines-for-simpler-machine-learning.html].

C. Theano [3]

Theano is a well-wrapped library over Keras, a neural system library virtually running in parallel with Theano. Keras is a sophisticated Python package that can remain running even when it's running on Theano or TensorFlow. It was developed to help people realize deep learning models quickly and simply. It is able to execute on GPUs and CPUs dependably. What distinguishes Theano is that it makes use of the GPU in the PC. This allows it to perform information escalation counts up to multiple times faster than when the CPU is used exclusively. Due to Theano's speed, it is especially advantageous for deep learning and other computationally difficult tasks.

D. Caffe [4]

Caffe is a learning framework that aims for speech speed, articulation, and quality. The Berkeley Vision and Learning Center (BVLC) and donors created it. DeepDream, for instance, relies on Caffe Framework. This is a BSD-licensed C++ library with a Python interface. Deep Learning architectures may be supported with Caffe. "Image classification" or "Image segmentation" may be on the list, as well. Caffe enables GPU acceleration using the CuDNN library from Nvidia. Thus, Caffe is an excellent option for quick, lightweight Deep learning network building.

E. Keras [5]

Keras is an open-source neural network library implemented in Python. Deeplearning4j, MXNet, TensorFlow, Theano, and Microsoft cognitive toolkit can operate on the top Keras. This framework has been developed to do lightning-fast neural network tests. Thus, because of this, Keras is minimalist, stable, customizable, and extendable by design. In 2017, TensorFlow's core library supported Keras. It provides for a streamlined neural network setup regardless of the underlying backend computing framework.

F. MxNet [6]

MXNet is a sophisticated open-source deep learning framework that accelerates the creation of deep learning algorithms. Deep learning has significantly influenced a broad range of industries in the previous several years, including healthcare, transportation, manufacturing, and more. This kind of learning, referred to as deep learning, is popular among corporations because it tackles tough issues like speech recognition, object identification, and machine translation. MXNet is often used to design, train, and deploy deep neural networks. It is versatile, ultra-scalable, and nimble.

G. PyTorch [7]

Facebook has designed a machine learning (AI) system named PyTorch. At the time of writing, it had more than 22,000 ratings on GitHub and 22,705 total files. It has picked up a tremendous amount of

energy since 2017, and that development has just begun. PyTorch is an open-source deep learning framework with the stability and support needed for production deployment. PyTorch is a Python package with capabilities including tensor computing with GPU acceleration and TorchScript for simple transitions between eager and graph processing. PyTorch is now offering graph-based execution, distributed training, mobile deployment, and quantization.

H. CNTK [8]

CNTK enables users to rapidly implement and integrate common model types such as feedforward Deep Neural Networks (DNNs), Convolutional Neural Networks (CNNs), and Recurrent Neural Networks (RNNs/LSTMs). It provides Stochastic Gradient Descent (SGD, error backpropagation) learning over several GPUs and servers with automated differentiation and parallelization. CNTK is freely accessible to download and use under an open-source license.

I. Auto ML [9]

Auto ML is arguably the most robust among the tools and libraries mentioned above and is a relatively new addition to the arsenal of tools accessible to a machine learning developer. Optimizations are critical in machine learning activities. While the rewards are substantial, establishing the ideal hyperparameters is not a simple process. This is particularly true for black boxes like neural networks, where discerning what matters gets more difficult as the network's depth grows.

J. OpenNN [10]

OpenNN (Open Neural Networks Library) is a C++ library for implementing neural networks. It supports supervised learning with an unlimited number of layers of non-linear processing units. It incorporates methods for regression, classification, forecasting, and association. Parallelization on the CPU and GPU may be accomplished using OpenMP and CUDA, respectively. The library is used to address real-world problems in the energy, marketing, and health sectors. Philips, Airbus, and Thales are among the companies that employ Notab.

K. H2O

H2O is a freely available platform for deep learning. It is a business-oriented artificial intelligence technology that assists users in deriving insights from data and assisting them in making decisions. It comes in two open-source flavors: regular H2O and Sparkling Water. Predictive modeling, risk and fraud analysis, insurance analytics, advertising technology, healthcare, and consumer intelligence can all be done with it.

Table 1. Pros and cons of AI tools and platforms

Platform / Library	Advantages	Disadvantages
Scikit Learn	• A robust Python machine learning library • A diverse collection of well-established algorithms with integrated visuals • Relatively straightforward installation, training, and usage. • Appropriate examples and tutorials	• deep learning or reinforcement learning are not covered in • There are no graphical models or sequence predictions. • It is not really possible to utilize it from other programming languages than Python. • Does not support PyPy or graphics processing units (GPUs).
Tensorflow	• The implementation of numerous algorithms in a more detailed and functional form • Excellent visualization for training models under the TensorFlow board.	• Prior to TensorFlow 2.0, setting data ingestion for TensorFlow might be a time-consuming and frustrating process.

	• Support for multiple GPUs as well as the availability of TPU for large-scale model training. • There is a large user community.	• It may be difficult to interpret certain TensorFlow error messages. • It will be great if there are better bindings with the JVM rather than only Python, given that many firms have a JVM-based stack, which would make it simpler to integrate.
Theano	• Theano is capable of creating symbolic graphs on its own in order to compute gradients. • Theano has a specific feature that allows it to discover numerically unstable formulas more quickly. It may then use this information to compute them for more stable algorithms. • Theano's APIs are well-known for being stable for a longer period of time despite changes.	• It may be difficult to work with AWS. • It is possible to run it on a single GPU. • Large models may need lengthy build times. • When it comes to debugging, error messages are of little assistance.
Caffe	• Caffe is a great framework for image processing and Feedforward networks. • It is well suited for fine-tuning the performance of existing deep neural networks. • Users may create models without the need to write a large amount of programming code.	• Caffe is not appropriate for the development of recurrent networks. • Working with large networks such as ResNet, GoogLeNet, and other similar networks may be time-consuming when using the Caffe framework. • A lack of commercial support discourages enterprise-grade developers from continuing their projects.
Keras	• Keras comes with a straightforward API that was influenced by the design of the 'Torch' framework. • Keras is a framewor that is quickly expanding. • On the backend, it supports integrated working with TensorFlow, Theano, Deeplearning4j, and CNTK. • Keras will very likely become the de facto standard Python API for Neural Network programming.	• Keras is not very suitable for applications that need low-level abstraction. • Customization that goes beyond the surface level may be time-consuming and frustrating. • Data processing tools are not high-performance, and it may be necessary to write sequence/non-sequence pre-processing procedures by hand to get the desired results.
MxNet	• The Gluon API is a high-level interface that simplifies model creation. • Extends to multiple GPUs distributed across several hosts with an efficiency of 85 percent. • Outstanding development speed and programmability • Excellent training speed is essential. • Python, R, Scala, Julia, Perl, C++, and Clojure are all supported languages.	• Keras/TensorFlow has a smaller ecosystem than the others. • It does not support TPUs or FPGAs. • According to the Apache Software Foundation, it is still in the "incubating" stage.
PyTorch	• A large number of modular elements that are simple to assemble. • It is simple to create your own layer types and execute them on a GPU. • A large number of pre-trained models	• Normally, you develop your own training code for each session. • There is no commercial support. • Documentation that is inconsistent
CNTK	• The notion of a computation graph is straightforward and simple to comprehend.	• If the syntax of the configuration file is unfamiliar, it may be preferable to

	• Using NDL to build a network is a quick process.	use the json format or Python to get started. • It is not yet comprehensive, and some of the commands in the examples are not included in the final version of the manual. • Text processing readers (e.g., LM sequence reader, LU sequence reader) are difficult to comprehend because they operate in a counterintuitive way.
Auto ML	• Prediction of the optimal pipeline for the labeled data by an automatic algorithm. • It automates a variety of iterative machine learning operations (like model selection, featurization) • Does not need any prior knowledge of Data Science or a technological background. • Low development costs and a shorter development time.	• Performance is not optimal. • It is not appropriate for sophisticated data structures and issues. • If the Dataset is too tiny, there will be performance problems.
OpenNN	• high performance • Outstanding in terms of execution speed and memory allocation	• Lack of full documentation • It could be better if major algorithms are added.
H2O	• AutoML • Algorithms are readily accessible and simple to implement in your analytical tasks. • More rapid than the Python scikit-learn module. • In addition to Java and other programming languages, Python may be used. • Well documented, making it ideal for quick training or self-study.	• There should be more state-of-the-art algorithms available. • Containerization technologies such as Docker should be available. • Improve the visual presentations, such as charts and graphs.

II. Conclusions

This paper introduced the concept of artificial intelligence and reviewed its important tools and libraries. It also lists the advantages and disadvantages of the reviewed libraries. The paper also introduces the importance of these libraries and platform in developing new algorithms and extending the current one. The future work involves more details about these libraries. More examples could be provided for researchers benefits.

References

[1]. Martin Heller, (2018). Review: Scikit-learn shines for simpler machine learning. Online at https://www.infoworld.com/article/3158509/review-scikit-learn-shines-for-simpler-machine-learning.html.
[2]. Goldsborough, P. (2016). A tour of tensorflow. arXiv preprint arXiv:1610.01178.
[3]. Team, T. T. D., Al-Rfou, R., Alain, G., Almahairi, A., Angermueller, C., Bahdanau, D., ... & van Tulder, G. (2016). Theano: A Python framework for fast computation of mathematical expressions. arXiv preprint arXiv:1605.02688.

[4]. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., ... & Darrell, T. (2014, November). Caffe: Convolutional architecture for fast feature embedding. In Proceedings of the 22nd ACM international conference on Multimedia (pp. 675-678).

[5]. Ketkar, N. (2017). Introduction to keras. In Deep learning with Python (pp. 97-111). Apress, Berkeley, CA.

[6]. Chen, T., Li, M., Li, Y., Lin, M., Wang, N., Wang, M., ... & Zhang, Z. (2015). Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. arXiv preprint arXiv:1512.01274.

[7]. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. arXiv preprint arXiv:1912.01703.

[8]. Seide, F., & Agarwal, A. (2016, August). CNTK: Microsoft's open-source deep-learning toolkit. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 2135-2135).

[9]. He, X., Zhao, K., & Chu, X. (2021). AutoML: A Survey of the State-of-the-Art. Knowledge-Based Systems, 212, 106622.

[10]. Lee, J. Y., Oh, J. W., & Lee, D. (2017). A Benchmark of Open Source Data Mining Package for Thermal Environment Modeling in Smart Farm (R, OpenCV, OpenNN and Orange). In Proceedings of the Korean Society for Agricultural Machinery Conference (pp. 168-168). Korean Society for Agricultural Machinery.

[11]. LeDell, E., & Poirier, S. (2020, July). H2o automl: Scalable automatic machine learning. In Proceedings of the AutoML Workshop at ICML (Vol. 2020).