SCIENTIFIC TEST
& ANALYSIS
TECHNIQUES
CENTER OF EXCELLENCE

Survey of AI-Enabled Tools for Application in Test & Evaluation

August 2022

Joseph Lazarus, Contractor
Summer Paschal, Contractor

The STAT COE provides independent STAT consultation to designated acquisition programs and special projects to improve Test & Evaluation (T&E) rigor, effectiveness, & efficiency.

About this Publication:
This work was conducted by the Scientific Test & Analysis Techniques Center of Excellence under contract FA8075-18-D-0002, Task FA8075-21-F-0074.

For more information:
Visit, www.AFIT.edu/STAT
Email, CoE@afit.edu
Call, 937-255-3636 x4736

Scientific Test & Analysis Techniques Center of Excellence
2950 Hobson Way
Wright-Patterson Air Force Base, Ohio

Version: 1, FY22

Modernizing *the* Culture *of* Test *&* Evaluation

Executive Summary

Emerging technologies (e.g., artificial intelligence (AI), autonomy) and operating concepts such as Joint All-Domain Command and Control (JADC2) Test and Evaluation (T&E) will require systems to undergo continual testing and produce greater amounts of data. Using AI throughout the testing cycle will enable testers to process the data and make more objective decisions at speed and scale. Due to the significant complexity of the system(s) under test, there is not a one-size fits all software application. Instead, there are a variety of software packages when used in a practical manner could enhance the capabilities of trained T&E professionals to address the challenges of emerging technologies. This paper presents a list of AI-enabled software tools and summarizes their functionality for potential application in T&E. Providing the test community this list, and potentially utilizing collaborative sites such as DoDTechipedia, will improve awareness of available tools and their functionality, encourage communication and collaboration, and assist the employment of current and future tools.

Table of Contents

Introduction

Due to the rapidly increasing amounts of data in test and evaluation (T&E), the use of artificial intelligence (AI) tools will be required to utilize the data effectively at speed and scale. The development of AI tools has been enabled by programmatically encoding human-codified knowledge and the introduction of machine-learning techniques. T&E professionals may find they are already using AI in their practice as many AI-enabled applications can support testers in a variety of ways such as planning, visualization, and analysis. However, to keep pace with the technologies that produce greater amounts of data, users will need to further incorporate AI throughout the T&E process. In this paper, we review AI tools for application in T&E in seven functional areas where AI can assist T&E professionals in their work. These seven areas include planning, storage, transportation, preparation, visualization, analysis, and monitoring. The main contribution of this work offers a framework and a list of AI-enabled resources to analyze AI's role within T&E.

This report is organized as follows. The background presents the complex history behind defining AI and establishes which authority on the matter is used to ground our discussion. Next, we offer our methodology for finding and evaluating AI resources. Lastly, key findings are presented and summarized.

Background

Because our definitions and measures of intelligence are diverse, it is no surprise that there isn't a general agreement on the definition of AI. Attempts at defining AI date back to the 1950s with Alan Turing and the "Turing-Test." The term "AI" was coined in 1956 by John McCarthy as part of the Dartmouth Summer Research Project on Artificial Intelligence. The imaginative and aspirational roles of AI in pop culture have further contributed to the ambiguous definition of AI.

Gregory Allen, former Chief of Strategy and Communications for the Joint Artificial Intelligence Center (JAIC), explains that most of AI fits into two branches: human codified knowledge and machine learning (ML). According to Mr. Allen, most of the AI in operation today uses hand-crafted knowledge. In this approach to AI, subject matter experts codify their knowledge into a long series of programmed rules that can be understood and executed by computers. Tax preparation software, aircraft autopilots, missile guidance systems, and electromagnetic signal processing systems are examples of AI based on human-codified knowledge.

In the past decade, most of the focus in AI has been within ML. This subfield of AI is about designing algorithms and statistical models to analyze and draw inference from latent patterns in the data. ML enables AI to adapt to new circumstances that the original developer didn't envision, detect patterns in diverse data sets and big data, create new behaviors based on recognized patterns, and make decisions based on the success or failure of these behaviors. Appendix A contains further definitions or various types of ML.

Methodology

This research establishes a framework of AI functionalities to evaluate tools that can assist the reader in understanding the objective of a given tool. Our review of available tools consists of product documentation, internet search, and manual collection.

Framework for Evaluating AI in T&E

AI tools are available in a variety of forms and utilize different approaches to suit user needs. As

T&E professionals incorporate AI in their practice, they can reference this list of resources to inform their software implementation decisions to accomplish a particular objective. This research establishes a framework with seven tool functions: plan, store, transport, prepare, visualize, analyze, and monitor. The seven functionalities are defined below.

- *Plan* (PL): planning involves understanding the requirements, screening characterization, designing factors, recording conditions, identifying constraints, creating test matrices, and determining the confidence level and power of hypothesis tests.

- *Store* (S): secure storage of big data that is accessible, reliable, and scalable. Solutions support rapid access to data across cloud environments and edge computing. Automation of workloads configure file management, access controls, as well as route and balance workloads. Optimizing expensive hardware like High Performance Compute (HPC) Clusters and Graphics Processing Unit (GPU) accelerators prepares data for processing.

- *Transport* (T): transportation of data from one location to another. Special use cases include data masking and encryption of secured data. For reproducibility purposes, it is important that any manipulation on the raw data is documented. Decisions of handling meta-data, important in preserving data quality, include processes for data that are too large to be loaded into Random-Access Memory (RAM), compression, sparsity, chunking, and hashing, to name a few.

- *Prepare* (PR): transformation of data into clean formats so that the algorithms can successfully use the information contained. This involves the handling of missing values, feature engineering, managing outliers, as well as imputation, transformation, normalization, and standardization processes.

- *Visualize* (V): graphical representation of data in any format. Exploring the data through visual outputs helps both technical and non-technical personnel have an overview of the data. Graphs and charts help assess the consistency of the data. Additionally, evaluation of model performance through visualization tools aid in communicating the results to stockholders.

- *Analyze (*A): modeling techniques are selected to address specific goals. This includes model building, parameter tuning, model retraining, gaining insight from the models, and interpreting the results.

- *Monitor* (M)*:* model version history performance is tracked for verification, evaluation, and audits. Continual testing/continual experimentation frameworks are managed and automatically alert users of any model decay. Reproducible models and creating standards through pipelines allow users to design, deploy, and manage consistent workflows. Providing scalable runtime resources adds the capability to manage and deploy web applications.

Methods for Identifying Tools and Labeling Functionality

We evaluated each tool against the seven functional areas using information from product documentation, web resources, and manual assessment. The following sections offer an overview for which product documentation we reviewed and how we performed the web-based search and manual assessment.

Product Documentation

Software documentation provides information that describes the product to the user base that deploys and uses it. The documentation is often made available online and in many cases is a living document that is updated throughout the product's life cycle. There are two main types of software documentation: internal or system documentation and external, which encompasses end-user documentation and system admin documentation. End-user documentation was the main reference in this research.

End-user documentation is focused on facilitating understanding of the product, interface, and capabilities. The quality, thoroughness, and ease of understanding varies from product to product. Typically, end-user documentation contains user manuals, operation manuals, and generic how-to tutorials.

Internet Search

We conducted a web-based search to obtain knowledge about commercially licensed tools and tools whose documentation are not publicly available. By using the Google search engine, we entered the tool name in the search query, and reviewed the top 25 results.

Manual Collection

We collected a list of tools presented at the Advancements in Test and Evaluation of Autonomous Systems (ATEAS) FY22 workshop. The Office of the Undersecretary of Defense for Research and Engineering (OUSD(R&E)) and the Director of Developmental Test Evaluation and Assessments (DTE&A) sponsored the Scientific Test and Evaluation Techniques Center of Excellence (STAT COE) in hosting the ATEAS FY22 workshop. The ATEAS workshops bring authorities and experts in the fields of autonomy, robotics, computer science, and more together to accelerate progress in T&E methods of autonomy. The field of autonomy, though separate from AI, overlaps in meaningful ways. Several organizations presented tools in various stages of development during this workshop. We manually collected information regarding these tools from the briefings and recordings.

The next section offers a comprehensive list of AI tools, descriptions, and functionalities.

Results

The following list of AI tools, descriptions, and functionalities offer the reader the ability to compare available AI tools used in planning (PL), storage (S), transportation (T), preparation (PR), visualization (V), analysis (A), and monitoring (M) within T&E.

List of AI Tools for T&E

Table 1
List of AI Tools to Include Descriptions and Functionalities

AI Tool	Description	Functionality						
		PL	S	T	PR	V	A	M
Alphaa AI	Tableau NLP and NLG search engine, AI voice analyst, and time series analysis capabilities					*	*	
AdaStress (Adaptive Stress Testing, AST)	Simulator that implements the AST framework, which determines the likeliest failures for a system under test using Markov decision process					*	*	
AdvoCATE (Assurance Case Automation Toolset)	Toolset for creating and visualizing safety assurance cases			*	*	*	*	
Amazon SageMaker	Service for building, training, and deploying ML models		*	*	*	*	*	*
Anaconda	ML and data science platform containing thousands of open-source packages and libraries	*	*	*	*	*	*	*
Apache Mahout	Distributed linear algebra framework and mathematically expressive Scala domain specific language for implementing algorithms			*	*		*	
Apache MXNet	Deep learning library used to define, train, and deploy deep neural networks			*	*		*	
Assurance-based Learning-enabled Cyber-Physical Systems (ALC) toolchain	Toolchain for system modeling, experimental and training data construction, performance evaluation, and system safety assurance monitoring	*	*		*		*	*
CoCoSim (Contract based Compositional verification of Simulink models)	Framework that integrates analysis technologies for verifying and validating Simulink and Stateflow models				*		*	
Databricks	Open-sourced, multicloud platform for data, analytics, and AI		*	*	*	*	*	*
DataRobot	Builds, trains, evaluates, deploys, and monitors accurate ML models		*	*	*	*	*	*
DeepMind Lab	First-person 3D game platform designed to build a range of environments, tasks, and intelligence tests in R&D for AI and ML systems					*	*	
FRET	Framework for the elicitation,							

(Formal Requirements Elicitation Tool)	specification, formalization, and understanding of requirements				*	*	*	
H2O	Platform for building ML models on big data and providing easy productionalization of those models in an enterprise environment		*	*	*	*	*	
IKOS (Inference Kernel for Open Static Analyzers)	Static analyzer for C/C++ based on the theory of Abstract Interpretation				*		*	
JMP	Statistical analysis software suite	*		*	*	*	*	
Keras	Deep learning API			*	*		*	
KNIME (Konstanz Information Miner)	Data analytics, reporting, and integration platform; Uses data pipeline concept to combine different components for ML and data mining			*	*	*	*	*
Marabou	SMT-based framework for verifying deep neural networks				*		*	
MATLAB (Matrix Laboratory)	Mathematical computation software and scripting language; implements interactive apps, ML algorithms, autoML, Simulink, and data visualization			*	*	*	*	
Microsoft Azure Machine Learning	Cloud service for accelerating and managing the ML project lifecycle; drag and drop interface to train, deploy, and monitor models	*	*	*	*	*	*	*
Microsoft Cognitive Toolkit (CNTK)	Builds, trains, and describes neural network models as a series of computational steps via directed graphs						*	
MLflow	Platform to streamline ML development; Monitors, stores, and loads models into production code and creates pipelines	*	*	*	*	*	*	*
NNV (Neural Network Verification)	Set-based verification framework for deep neural networks and learning-enabled cyber-physical systems				*		*	
NumPy (Numerical Python)	Python library providing N-dimensional array objects and high-level mathematical functions fundamental for scientific computing				*			
OpenAI Gym	Library for developing and comparing reinforcement learning algorithms					*	*	
OpenNN	High performing neural network library	*		*	*		*	
pandas	Python library for data manipulation and analysis			*	*	*	*	

PyTorch	Optimized tensor library for deep learning using GPUs and CPUs			*	*	*	*	
RapidMiner	No-code development data science platform with visual workflow design and full automation			*	*	*	*	
RAPT (Range Adversarial Planning Tool)	Software framework for testing autonomous and robotic systems using adaptive sampling to discover and identify performance boundaries						*	
R2U2 (Realizable Responsive Unobtrusive Unit)	Framework for runtime monitoring of security properties and diagnosing of security threats on-board Unmanned Aerial Systems						*	*
RIOT (Robustness Inside Out Testing)	Scalable robustness testing technique for node-based autonomous systems						*	
RStudio	Statistical computing and visualization IDE for R	*	*	*	*	*	*	*
SAS (Statistical Analysis System)	Statistical analysis software suite	*	*	*	*	*	*	*
Scikit-Learn	Python ML library for predictive data analysis, built on top of NumPy, SciPy, and matplotlib			*	*	*	*	
Snowflake	Cloud-based data warehousing and analytics system which gives users access to store and analyze data		*	*	*	*	*	*
Tableau	End-to-end data analytics and visualization platform		*	*	*	*	*	*
TensorFlow	End-to-end open-source platform for building and deploying ML models			*	*	*	*	
VerifAI	Toolkit for the formal design and analysis of AI-based cyber-physical systems	*					*	
Weka (Waikato Environment for Knowledge Analysis)	Software containing a collection of visualization tools, algorithms for data analysis and predictive modeling, and GUIs for easy access		*	*	*	*	*	

*Limitations: The list provided here is not comprehensive. However, it does provide a survey of tools that exist today. As the field of ML continues to grow and innovate, new tools are expected to be released. Absent from this research are tools for AI-enabled software testing. Though awareness of their existence is known, it is an area beyond the expertise of this paper's authors. Further research on tools for AI-enabled software testing is needed.

Conclusion and Recommendations

In the future, the amount of data that will be generated in testing will vastly increase. To process the data efficiently and make objective decisions, testers will need to use AI tools. This research identifies seven functional areas that AI can assist testers and presents the reader with a list of AI tools, descriptions, and functionalities. However, the list is not comprehensive, and the ML space is constantly developing new tools.

No one tool provides complete coverage for every use case. Each package has their strength and weaknesses, though not all weaknesses are well known or obvious. To enhance this product and leverage the collective knowledge of the community, this work should be made available on a collaborative website such as DoDTechipedia. Once published on the site, users may add tools they have worked with and provide their input regarding evaluation criteria. Sourcing the collective knowledge is one way to keep pace with the changing testing environment as emerging technologies start appearing in the acquisition pipeline.

References

Amazon Web Services. (2017, November 29). *Introducing Amazon SageMaker*. https://aws.amazon.com/about-aws/whats-new/2017/11/introducing-amazon-sagemaker/

Anaconda. (n.d.). *Anaconda: The World's Most Popular Data Science Platform*. https://anaconda.cloud/most-popular-data-science-platform

Arnaldo, M. (2021, October 4). *Intro to Rapidminer: A No-Code Development Platform for Data Mining (with Case Study)*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/10/intro-to-rapidminer-a-no-code-development-platform-for-data-mining-with-case-study/

Artelnics. (2022, July 1). *OpenNN neural networks*. GitHub. https://github.com/Artelnics/OpenNN

Arthaud, M., Bailleux, T., Brat, G., Decoodt, C., Hamon, A., Navas, J., Simms, E.-J., Shi, N., Thompson, S., Venet, A., Wimmers, A., Kim, S.K., & Thakur, A. (2022, June 23). *IKOS*. GitHub. https://github.com/NASA-SW-VnV/ikos

Basoglu, C., Pulavarthi, P., Miguel, G., Fu, L., Gronlund, C.J., Crepaldi, T., Iao, M., Barrett, S., Jindal, M., Manousek, W., Hillebrand, M., Orlov, A., & Ahlers, D. (2022, February 16). *The Microsoft Cognitive Toolkit*. Microsoft. https://docs.microsoft.com/en-us/cognitive-toolkit/

Beattie, C., Leibo, J.Z., Teplyashin, D., Ward,T., Wainwright, M., Küttler, H., Lefrancq, A., Green, S., Valdés, V., Sadik, A., Schrittwieser, J., Anderson, K., York, S., Cant, M., Cain, A., Bolton, A., Gaffney, S., King, H., Hassabis, D., …Petersen, S. (2016, December 14). DeepMind Lab. arXiv. https://doi.org/10.48550/arXiv.1612.03801

Chen, T., Li, M., Li, Y., Lin, M., Wang, N., Wang, M., Xiao, T., Xu, B., Zhang, C., & Zhang, Z. (2015, December 3). *MXNet: A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed Systems*. arXiv. https://doi.org/10.48550/arXiv.1512.01274

Databricks. (n.d.). *The Databricks Lakehouse Platform*. https://databricks.com/product/data-lakehouse

DataRobot. (2018, April 16). *How DataRobot Works* [Video]. YouTube. https://www.youtube.com/watch?v=RrbJLm6atwc

Denney, E., Pai, G., & Pohl, J. (2012). AdvoCATE: An Assurance Case Automation Toolset. In Ortmeier, F., & Daniel, P. (Eds.), *Computer Safety, Reliability, and Security SAFECOMP 2012: Lecture Notes on Computer Science* (Vol. 7613, pp. 8-21). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-33675-1_2

Department of Defense Joint AI Center. (2020, September 14). *Explaining Artificial Intelligence and Machine Learning* [Video]. YouTube. https://www.youtube.com/watch?v=y_rY0ZIn5L4

Dreossi, T., Fremont, D.J., Ghosh, S., Kim, E., Ravanbakhsh, H., Vazquez-Chanlatte, M., &

Seshia, S.A. (2019). VERIFAI: A Toolkit for the Formal Design and Analysis of Artificial Intelligence-Based Systems. In Dillig, I. & Tasiran, S. (Eds.) *Computer Aided Verification Lecture Notes in Computer Science* (Vol. 11561, pp. 432-442). Springer, Cham. https://doi.org/10.1007/978-3-030-25540-4_25

Giannakopoulou, D., Mavridou, A., Rhein, J., Pressburger, T., Schumann, J., & Shi, N. (2020, March 24). *Formal Requirements Elicitation with FRET*. Nasa Technical Reports Server. https://ntrs.nasa.gov/citations/20200001989

Gilley, S., Zhong, J., Buck, A., Salgado, S., Gayhardt, L., Gronlund, C.J., Hansen, D.P., & Mungi, R. (2022, June 8). *What is Azure Machine Learning?* Microsoft. https://docs.microsoft.com/EN-US/azure/machine-learning/overview-what-is-azure-machine-learning

H2O.ai. (2022, July 8). *Welcome to H2O 3*. https://docs.h2o.ai/h2o/latest-stable/h2o-docs/welcome.html

IBM Technology. (2021, July 14). *What is Machine Learning?* [Video]. YouTube. https://www.youtube.com/watch?v=9gGnTQTYNaE

Institute for Software Integrated Systems. (2022). *ALC Toolchain Overview*. Vanderbilt University. https://editor-alc.isis.vanderbilt.edu/doc/tutorial/Overview.html

JMP. (n.d.). *JMP: Data analysis software for Mac and Windows*. https://www.jmp.com/en_us/software/data-analysis-software.html

Kahsai, T., Garoche, P., Bourbouh, H., Pagetti, C., Loquen, T., Noulard, E., & Dieumegard, A. (2017, May 2). *CoCoSim*. GitHub. https://coco-team.github.io/cocosim/

Katz, D.S., Zizyte, M., Hutchison, C., Guttendorf, D., Lanigan, P.E., Sample, E., Koopman, P., Wagner, M., & Goues, C.L. (2020). Robustness Inside Out Testing. *2020 50th Annual IEEE-IFIP International Conference on Dependable Systems and Networks -Supplemental Volume (DSN-S)*, 1-4. https://doi.org/10.1109/DSN-S50200.2020.00013

Katz, G., Huang, D.A., Ibeling, D., Julian, K., Lazarus, C., Lim, R., Shah, P., Thakoor, S., Wu, H., Zeljic, A., Dill, D., Kochenderfer, M.J., & Barrett, C. (2019). The Marabou Framework for Verification and Analysis of Deep Neural Networks. In Dillig, I. & Tasiran, S. (Eds.), Computer Aided VerificationcLecture Notes in Computer Science (Vol. 11561, pp. 443-452). Springer, Cham. https://doi.org/10.1007/978-3-030-25540-4_26

Keras-Team. (2022, July 28). *Keras: Deep Learning for humans*. GitHub. https://github.com/keras-team/keras

Knime. (n.d.) *KNIME Software: End to end data science for better decision making*. https://www.knime.com/software-overview

LeCun, Y., & Misra, I. (2021, March 4). *Self-supervised learning: The dark matter of intelligence*. MetaAI. https://ai.facebook.com/blog/self-supervised-learning-the-dark-matter-of-intelligence/

Lipkis, R. (2022, February 15). *AdaStress.jil: Reinforcement learning framework to find and*

analyze the likeliest failures of a system under test. GitHub. https://github.com/NASA-SW-VnV/AdaStress.jl

MathWorks. (n.d.). *MATLAB for Machine Learning: Train models, tune parameters, and deploy to production or the edge*. https://www.mathworks.com/solutions/machine-learning.html
MLflow. (2022, July 20). *MLflow Documentation*. GitHub. https://github.com/mlflow/mlflow/blob/master/docs/source/index.rst

Mullins, G. (2018). *Adaptive Sampling Methods for Testing Autonomous Systems* [Doctoral dissertation, University of Maryland]. Digital Repository at the University of Maryland. https://doi.org/10.13016/M2PZ51Q5N

NumPy Developers. (n.d.). *NumPy: the absolute basics for beginners*. NumPy. https://numpy.org/doc/stable/user/absolute_beginners.html

Openai. (2022, July 26). *Gym*. GitHub. https://github.com/openai/gym

Owen, S., Anil, R., Dunning, T., & Friedman, E. (2011). *Mahout in Action*. Manning. https://livebook.manning.com/book/mahout-in-action/chapter-1/

Pandas development team. (n.d.). *Package overview*. Pandas. https://pandas.pydata.org/docs/getting_started/overview.html

Pervez, S. (2019, July 10). *Introducing Alphaa: where artificial intelligence meets analytics*. Medium. https://medium.com/alphaa-ai/introducing-alphaa-where-artificial-intelligence-meets-analytics-3bd860698913

PyTorch. (2022, July 29). *PyTorch*. GitHub. https://github.com/pytorch/pytorch

RStudio. (n.d.). *RStudio: Take control of your R code*. https://www.rstudio.com/products/rstudio/

SAS Institute. (n.d.). *Artificial Intelligence (AI) Solutions: Augmenting human creativity and endeavors with AI*. https://www.sas.com/en_us/solutions/ai.html

Schumann, J., Moosbrugger, P., & Rozier, K.Y. (2017, April 12). R2U2: monitoring and diagnosis of security threats for unmanned aerial systems. *Formal Methods in System Design*, 51**,** 31–61. https://doi.org/10.1007/s10703-017-0275-x

Scikit-learn. (n.d.). *Scikit-learn: Machine Learning in Python*. https://scikit-learn.org/stable/index.html

Smith, C., McGuire, B., Huang, T., & Yang, G. (2006, December). *The History of Artificial Intelligence*. University of Washington. https://courses.cs.washington.edu/courses/csep590/06au/projects/history-ai.pdf

Snowflake, Inc. (n.d.). *Snowflake for Data Science & ML*. https://www.snowflake.com/workloads/data-science/

Tableau. (n.d.). *What is Tableau?*. https://www.tableau.com/trial/what-is-tableau

TensorFlow. (2022, July 29). *TensorFlow*. GitHub. https://github.com/tensorflow/tensorflow

Tran, H.-D., Yang, X., Lopez, D.M., Musau, P., Nguyen, L.V., Xiang, W., Bak, S., & Johnson, T.T. (2020). NNV: The Neural Network Verification Tool for Deep Neural Networks and Learning-Enabled Cyber-Physical Systems. In Lahiri, S. & Wang, C. (Eds), *Computer Aided Verification Lecture Notes in Computer Science* (Vol. 12224, pp. 3-17). Springer, Cham. https://doi.org/10.1007/978-3-030-53288-8_1

Turing, A.M. (2009). Computing Machinery and Intelligence. In Epstein, R., Roberts, G., & Beber, G. (Eds.), *Parsing the Turing Test* (pp. 23-65). Springer, Dordrecht. https://doi.org/10.1007/978-1-4020-6710-5_3

Weka (machine learning). (2022, February 17). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Weka_(machine_learning)&action=history

Appendix A

Additional Machine Learning Definitions

ML is divided into five categories:

- *Supervised Learning (SL)*: rows in a data set are labeled and used to train algorithms to find the relationships of inputs and known outputs, classify data, or predict outcomes. The robustness of the model is judged by what is known in the real world. Examples of SL models include linear regression, logistic regression, support vector machines, and ensemble models. These models can be applied to predictive analytics, pattern recognition, natural language processing, and facial detection to name a few.

- *Unsupervised Learning (UL)*: algorithms are used to analyze and cluster unlabeled data to aid in the discovery of hidden patterns or groupings without the need of human intervention. UL consists of centroid hierarchical distribution and density-based clustering. Clustering algorithms group data points that are similar to each other based on their relations to surrounding data points and are used for pattern identification and feature engineering. Sometimes used in conjunction with clustering is principal component analysis (PCA). PCA allows visualization and exploration of data and reduces data set size to speed up computational time.

- *Semi-supervised Learning*: a combination of SL and UL approaches using labeled and unlabeled data where the algorithms can exploit the extra information contained in the unlabeled data. This can help to improve the accuracy of the models that are learned. Some of the most popular use cases are speech analysis as well as text and image classification.

- *Self-supervised Learning:* supervisory signals are obtained from the data, often leveraging the underlying data structure. The general technique of self-supervised learning is to predict an unobserved or hidden part of the input from any observed part of that input. This makes use of a variety of supervisory signals across co-occurring modalities without relying on labels. Contrastive energy-based self-supervised learning is based on the idea of constructing pairs of x and y that are not compatible and adjusting the parameters of the model so that that corresponding energy is large. Non-contrastive energy-based self-supervised models compute virtual target embeddings for groups of similar images.

- *Reinforcement Learning (RL)*: training data are collected by an autonomous, self-directed AI agent that perceives, learns, and reacts with a simulated or real-world environment. The environment will then either use positive or negative reinforcement to increase the strength and frequency of the behavior. Through many iterations, a system can be taught a particular task. RL is used in gaming applications, robotics manipulation, and autonomous driving.

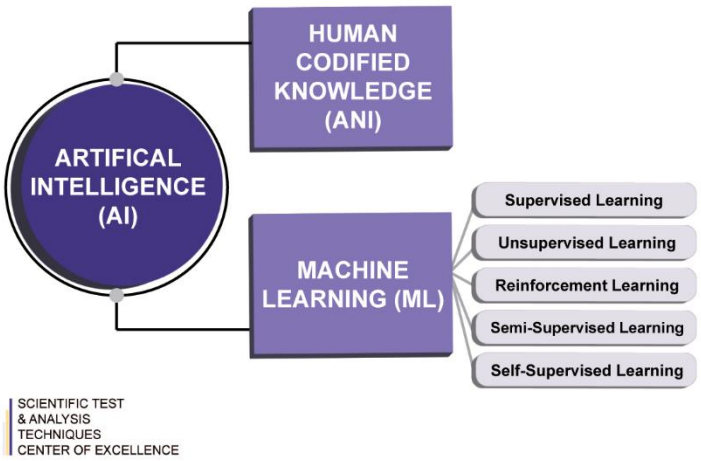

Figure 1
AI Ontology

Terms commonly used in ML include Neural Networks, Deep Learning (DL), and Natural Language Processing (NLP).

- *Neural Networks* - a subset of ML, also known as Artificial Neural Networks (ANNs). Their name and structure were designed to mimic the process of the human brain. A network is made up of many units of neurons which are grouped into three layers: an input layer, any number of hidden layers, and an output layer. Each neuron interacts with the neurons of the next layer through weighted connections and a bias term. The neurons inside a neural network work as a regression ensemble. There are multiple types of neural networks, such as Convolutions Neural Networks (CNN) and Recurrent Neural Networks (RNN). CNNs have a unique architecture that is well suited for identifying patterns such as image recognition tasks. RNNs are identified by their feedback loops and are used in time series analysis and NLP tasks. ANNs are used in a variety of ML domains such as SL, RL, as well as Semi and Self-Supervised learning.

- *Deep Learning* - a subset of ML involving Neural Networks or ANNs. ANNs are said to be deep if there are multiple hidden layers. Most implementations of ANNs today are DL.

- *Natural Language Processing* - a subset of ML, referring to the development and use of machine-based methods to process content in natural language. Within NLP, there are two subcategories: Natural Language Understanding (NLU) and Natural Language Generation (NLG). NLU is the practice of getting machines to produce useful representation of some natural language input. NLG is the practice of using machines to produce usable natural language output that is not identical to its input. Approaches to NLP use SL, UL, and ANNs.