Machine Learning and Deep Learning Frameworks and Libraries for Large-Scale Data Mining

Aniket Agarwal

Abstract: ***The study contains the impact in emerging computing resources a technique along with increased avalanches of wider datasets to transfer several research areas and leads to the technological breakthrough which is used by the several people effectively. The advanced technologies are developed in the machine learning especially in Deep Learning. The technologies are provided here concentrating on the fields for analyzing and learning from the wider range of data from real world. This study presents the comprehensive overview of the libraries and frameworks in machine learning and Deep learning in wider scale data mining.***

Keywords: Machine Learning, Deep Learning, Large-scale Data mining, Libraries and Framework

1. Introduction

Data mining contains the primary stage of the processes of knowledge discovery which aims for extracting the useful and essential information from data. The term "data mining" directs to the core oriented large scale mining of data containing several techniques for performing effectively for the larger-scale datasets. Data mining can be responsible for serving machine learning and AI. There are various types of techniques in the fields like AI "Artificial intelligence", ML "Machine learning", NNS "Neural networks" and DL "Deep learning". The frameworks and libraries of machine learning and deep learning in large scale data mining are considered here.

2. Machine Learning and Deep Learning

2.1 Machine Learning process

The realization of the DM in wider areas can be responsible for the "Cross-Industry standard process for Data mining cycle" (CRISP-DM) which now contains the de facto standard for the DM applications containing six different phases.

a) Understanding the business depending on the quest formulations.
b) Understanding of data depending on documentations.
c) Preparing data containing the feature engineering, data transformation and EDA "Exploratory data analysis".
d) Application of ML algorithms having different parameter calibrations in the modelling phase.
e) Performing an evaluation phase based on various criteria for choosing the best model.
f) Conducting the deployment phase involving the usage of ML models for exploiting the functionality.

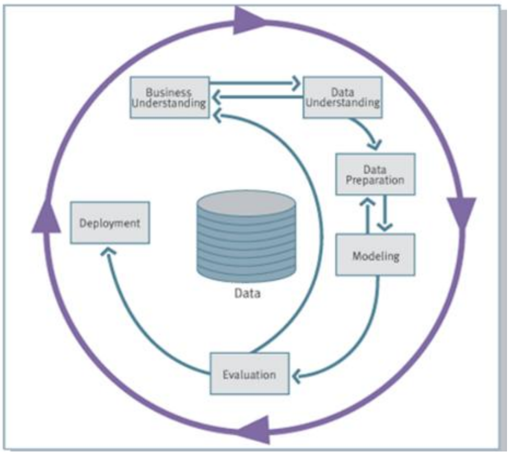

Figure 1: Cross-Industry process for data mining
(Source: https://medium.com/@thecodingcookie/cross-industry-process-for-data-mining-286c407132d0)

The entire cycle is repetitive. The categorization of the data preparation models can be conducted into various types of sub-categories like "dimensionality reduction, sampling (sub-sampling, oversampling), data augmentation, linear methods, statistical testing, feature engineering with feature extraction, feature encoding, feature transformation and feature selection". Various types of algorithms are also there for over-fitting prevention.

2.2 Neural Networks and Deep Learning

Neural networks are mainly the subside of Machine learning. The networks are not much realistic, but the string algorithms, along with data structures, can model the difficult problems. Neural networks contain the units which are organized in the three layers, i.e. input layer, hidden layer and the output layer. The capabilities of predictions in NNs derived from the "hierarchical multi-layers structure"[1]. The sufficient training provided to the network can be responsible for learning the inputs having several sales and combining the higher-order features related to the representations.

The "Deep neural networks" contain the capabilities to learn the higher-level features and have more complexities and abstractions consisting of various numbers of hidden layers. There are two types of dependent problems for achieving the higher accuracies in predictions which is to be addressed in solving the problems with neural networks.

3. Accelerated Computing

Deep learning contains the profits in using the specialized formed hardware presenting in the "accelerated computing environments". The current solution is mostly based on the usage of GPUs in general processors providing huge parallelism to deal with the larger DM problem and allow the scaling algorithms which are computable using the traditional approaches[2]. The GPUs are providing better solutions for the real-time systems, which requires the learning and decision making techniques in DL. There are various types of alternatives like "Google tensor processing unit", "field-programmable gate array" etc. dedicated hardware also offered for the acceleration of DL[3]. The improvements in the memory consumptions and speed in DL algorithms can be conducted like Sparse computation, low precision data types etc. For speeding up the computations, involving the accelerated libraries are crucial like NVIDIA CUDA, Intel MKL, OpenCL etc. The other programming libraries supporting the "computations speed up, and parallel computing" are OpenMP, OpenMPI etc. for using the parallelism in the multicore nodes.

4. Machine Learning frameworks and libraries

There are a huge number of ML algorithms available containing several types of software implementation. Various types of software tools are in the development stage for DM using the ML techniques. The common goal is for facilitating the complicated analysis of data processes and for proposing the integrated environment on the standard processing language. The tools are developed depending on several purposes like predictive systems, analytic platforms, processes etc. The figure is provided below containing the entire overview of the machine learning frameworks and the libraries[4]. All the DM tools and ML/DL frameworks are considered here along with the libraries containing GPU supports here.

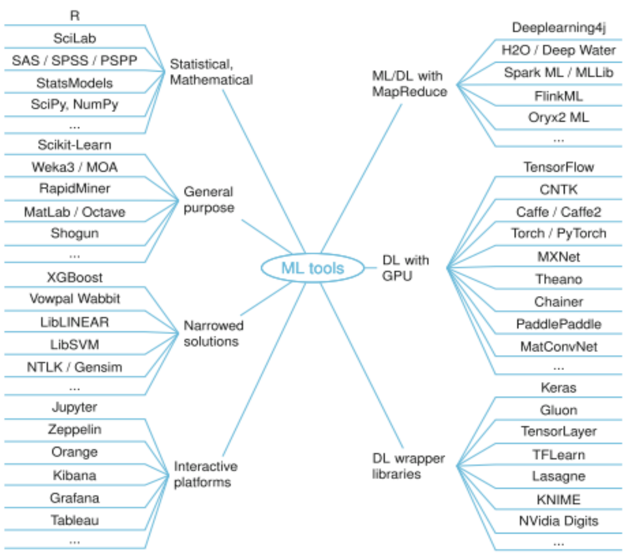

Figure 2: Overview of Machine Learning frameworks and libraries
(Source: G. Nguyen et al.)

4.1 Machine Learning frameworks and libraries without special hardware support

4.1.1 Shogun

The Shogun is an "open-source general-purpose ML library" which offers a wider range of effective L methods developed in the architecture written in the langu8age C++ and licensed under "GNU GPLv3 license"[5]. The development was started from the year 1999, and now it is sponsors by the NumFOCUS after 2017. The essential idea of Shogun was the underlying algorithms using transparency, accessible so that it can be used by anyone freely. There are 15 implementations in the SVM library combined with 35 kernel implementations which also contains the sub-kernel weighting. The use of Shogun can be done in several languages and environments containing Python, R, Java, C# and Octave.

4.1.2 RapidMiner

RapidMiner is the "general purpose data science software platform" in preparation of data, ML, text mining, DL and predictive analytics. It started development in 2001conatinnig the AI unit. It consists of the cross-platform framework, which is developed in the open core model written in Java language. The interactive modes, java API and command-line interface, are supported by the RapidMiner[6]. The primary architecture of RapidMiner inner source and contains a wider set of algorithms having the learning schemes, algorithms and models using the Weka and R scripts.

4.1.3 Weka3

Weka contains the popular wider ML algorithms, which is mostly implemented in Java. It has the package system for extending the functionalities having both the unofficial and official packages. It is responsible for offering different options for DM like "command-line interface (CLI), Explorer, Experimenter, and Knowledge Flow". Weka can be deployed with Hadoop containing the abilities for producing the wrapper set in the advanced version of Weka. It directly supports MapReduce but not Apache Spark still now. It supports the DM tasks containing the clustering selection of features, classifications, visualizations and regressions.

4.1.4 Scikit-Learn

It is known as the popular open-source python tool containing the comprehensive libraries having DM/ML algorithms. It was started as the code project on google in 2015. It is still under development. It is responsible for extending the functionalities of SciPy and NumPy packages along with various DM algorithms and providing the functions for performing the activities like regression, clustering, classifications, reduction of dimensionality and selection of models along with preprocessing[7]. The Matplotlib package can also plot the charts effectively. The crucial strong points in Scikit-learn is its well-updating cycle and the set of algorithms with implementations.

4.1.5 LibSVM

LibSM contains the specialized libraries to support the SVM "Support vector machine". The development was started in 2000. the entire library was written in the C/C++ language and also contained the java source codes. The learning tasks of the library are:

a) Supporting vector classification in binary and multi-class.
b) Supporting vector regression
c) Distribution estimations.

It is also responsible for providing the inferences for R, MATLAB, Python, Ruby, Weka, Perl, PHP, Haskell etc. reusing the DM tools i.e. Weka, KNIME and RapidMiner[8]. It is declared that LibSVM uses Scikit-learn for handling the computations effectively having improvements and modifications.

4.1.6 LibLinear

LibLinear could be a library designed for finding "large-scale linear classification" issues. It was developed beginning in 2007 at National Taiwan University. C/C++ was used to write the library. The supported metric capacity unit tasks are supply regression and linear SVM[9]. The supported drawback formulations are L2-regularized supply regression, L2-loss and L1-loss linear SVMs. For multi-class issues, LibLinear implements the one-vs-rest strategy and the Crammer & Singer technique. LibLinear provides interfaces for "MatLab, Octave, Java, Python and Ruby". The DM tools also reuse the codes for handling the computations. The library has huge popularity in the "open -source ML community".

4.1.7 Vowpal Wabbit

Vowpal Wabbit is mainly the effective implementation of the online Machine Learning techniques supporting the several types of incremental method in ML[10]. It is the "open-source, fast out-of-core learning system" developed by Yahoo! Microsoft Azure offers the VW as the ML options contain the features like "reduction functions, importance weighting, selection of different loss functions and optimization algorithms". The VW is considered for learning the feature data set on 1000 nodes at the time of one hour and can be run effectively in a specific machine. Various top IT companies are using VW, as it is scalable and effective in online learning.

4.1.8 XGBoost

XGBoost is the "optimized distributed gradient boosting library" which is designed having more efficiency along with portable and flexible capabilities. It is an open-source library which is responsible for implementing the "gradient boosting decision tree algorithm". It contains huge popularity and attention because of its consideration in various ML competitions. XGBoost generally implements the ML algorithms under "Gradient Boosting framework"[11]. It also provided the "parallel tree boosting" which is responsible for solving the data science problems effectively in a faster way along with huge accuracy—iterative learning used by XGBoost for boosting the weak learning models effectively.

4.1.9 Interactive data analytic and visualization tools

The tools are responsible for displaying the analytic results having the interactive ways in this category, providing an enhanced manner for understanding the major concepts without difficulties in supporting the decision-makers. Several data visualization packages contain several abstraction levels in Python or R like "matlablib, ggplot, seaborn, plotly, bokeh". The web-based applications are nowadays used by the users more and gaining huge popularity[12]. The integrated data analytic environment is considered here for creating and sharing eh documents which contain the "data-driven live code, equations, visualizations and narrative text". The most renowned are Jupyter notebook and Zeppelin. Besides that, there are more open-sources tools in data analytics, integrating platforms and reporting like, kibana, Grafana, Tableau containing the big data frameworks having cloud-based resources. The Reich ecosystem is there containing interactive tools which are designed based on several specific purposes.

4.1.10 Other data analytic frameworks and libraries

There are a huge number of libraries and frameworks to use the ML/NN and DL techniques. The familiar subsets are:

a) Matlab

Matlab provides the "multi-paradigm numerical computing environment" using the proprietary programming language which we developed by MathWorks. It is very popular, having 2 million users.

b) SAS

The Statistical Analysis System started as the project for analyzing the archival data in the "North Carolina State University". It is software written in the C language having more than 200 components.

c) R

It is a free software used in statistical computing in linear and nonlinear modelling[13]. R is very efficient to be used and can be extensive by packages. It provides more than 10000 packages.

d) Python

Python is the programming language developed by Guido van Rossum. It is much successful being used in a number of real-world businesses worldwide. It features huge types of dynamic nature along with automatic management of memories containing huge libraries in scientific computations.

e) Scipy

It is the “Open source Python library” used to scientifically and technically compute. It is built on the “NumPy array object” and belongs to a part of the “NumPy stack” including the tools like “Matplotlib, Pandas, and SymPy”.

f) Pandas

It contains the python package to provide flexible, expressive and faster data structures which is designed for making it a much easier song with rational and labelled data[14]. It contains n two primary data structures, i.e. series and Dataframe, for handling the huge majority having typical use cases in several engineering areas, finance, statistics and social science.

4.2 Deep Learning frameworks and libraries

There are various types of popular ML frameworks and libraries available for offering the possibilities in using the GPU accelerators for speeding up the learning processes along with various supported interfaces. A significant overview of the poplar DL frameworks and libraries are provided here efficiently.

4.2.1 TensorFlow

TensorFlow contains the "open-source software library" for conducting the numerical computations by using several data flow graphs. It was created and maintained by the Google Brain team having the machine intelligence for DL and ML. It mainly designed focusing on the larger and wider distributed interference and training[15]. The Nodes in the graph is responsible for representing the mathematical operations along with the graph edges representing the "multidimensional data arrays" for communicating among them. The architecture also contains the worker services and distributed master along with the implementation of the kernel. It consists of more than 200 standard operations with "mathematical, array manipulation, control flow, and state management operations" which is generally written in the C++ language. It is developed for being used in the development, research and productions. Additionally, the TensorFlow lite is there having the lightweight solutions for the embedded and mobile devices enabling the low latency infrastructure for providing more flexibility to be used in more devices.

4.2.2 Keras

Keras is considered to be a Python "wrapper" library which helps in giving bindings to different DL tools which include Deep learning, TensorFlow, Theano, CNTK and MXNet that have beta versions. With the help of Keras, fast experimentation is possible and possesses the license of MIT for release. It applies 2.7-3.6 version of Python and performs execution on CPUs and GPUs with the help of suitable frameworks[16]. Guiding principles are modularity, minimalism and user-friendliness, easy extensibility and suitable with the use of Python code. This is a fast, evolving and open source and involves a popular API that performs documentation.

4.2.3 Microsoft CNTK

Microsoft CNTK (Cognitive Toolkit) is defined to be the DL framework which involves grade distribution of large-scale datasets which are obtained through Microsoft Research. Efficient training with DNNs for handwriting, speech, text data and image. There is a generation of graphs which involves symbols for vector operations and hence used for specifying its network—for example, multiplication and addition of building blocks[17]. CNTK supports RNN, CNN, FFNN architectures and helps in the implementation of SGD learning which involves parallelization and automatic differentiation across servers and GPUs. It supports ONNX format that helps in the model transformation between MXNET, PyTorch and CNTK.

4.2.4 Caffe

Caffe is known to be a framework in DL which comprises modularity, speed and expression in mind. YangqingJia established Caffe at Berkeley Artificial Intelligence Research which is explained through layer-by-layer. For computational purposes, this layer is considered to be a fundamental unit and essence for a particular model[18]. The data is being inserted with the help of data layers. HDF5, LMDB and PDF or GIFs are considered to be those data sources which are accepted. C++ Cuda must be used for writing new layers and custom layers in spite of less efficiency are "supported" in Python.

4.2.5 Caffe2

Caffe2 is defined to be a scalable, modular and lightweight DL framework which was established by "YangqingJia". In spite of the involvement of straightforward and easy methods for performing experiments with DL and implementing leverage "community" contributions for developing new algorithms and models, Caffe2 is majorly used within the levels of production at Facebook. The difference between Caffe and Caffe2 is in improvement directions which involves hardware support and mobile deployment through the involvement of CUDA and CPU[19]. It basically focuses on industrial-strength applications for enhancing the performance of mobile functions. Python scripts which involve command lines are used by Caffe2 for making suitable translations from Caffe models and involve ONNX format.

4.2.6 Torch

Torch is considered to be a scientific framework which performs computing with support for developing ML algorithms which involve the use of "Lua programming language". Facebook, Google, Twitter and DeepMind use Torch, which is being certified with a license from BSD. The object-oriented paradigm is being used by Torch and hence finds its utility in C++. It is flexible, readable and involves high speed and modularity. The tensor library is used for developing its core which is available with GPU and CPU backends. Parallelism is supported by this framework and hence being used on GPUs.

4.2.7 PyTorch

PyTorch is considered to be a library function of Python for its application in GPU-accelerated DL. The involvement of C libraries which are being used by Torch is present in the Python interface of PyTorch. Since 2016, the research group for AI of Facebook has been involved in its development. CUDA, C and Python are used for writing PyTorch. Acceleration libraries like NVIDIA and Intel MKL are being integrated with the use of PyTorch. Tensor computation is being supported by PyTorch which involves strong acceleration in GPU whereas DNNs are developed on "tape-based" auto generated systems. It helps in the generation of dynamic "computational" graphs for reverse-mode-auto-differentiation.

4.2.8 MXNet

Apache MXNet is basically considered to be a framework in DL, which is designed for flexibility and efficiency. It permits the mixing of imperative and symbolic programming that helps in increasing productivity and efficiency[20]. A "dynamic dependent scheduler" is being present at the core which parallelism imperative and symbolic operations on-the-fly automatically. The top layer involves graph optimization, which helps in making memory efficient and fast execution through symbols. MXNet is lightweight and portable and provides effective scaling to multiple machines and GPUs. Apache-2.0 license is provided to MXNet and hence involves API language for Julia, R and Python. It helps in auto-parallelism.

4.2.9 Chainer

Chainer involves standalone, Python-based open-source DL framework for models. DL models like reinforcement learning, CNN and RNN are developed with the help of Chainer. Automatic differentiation, e.g. APIs, are developed with Chainer, which involves the approach of Define-By-Run. Construction and training of NNs are being done with the help of these APIs and DCGs. It involves the use of CUDA which in turn uses CuPy for providing high-performance interference and training and hence uses Intel MKL for DNNs which helps in accelerating DL frameworks based on Intel architectures. It helps in providing libraries for "industrial purposes" and hence used by Toyota and NTT.

4.2.10 Theano

Theano is considered to be a pioneering tool in DL which provides support to GPU computation since 2007. It possesses a BSD license which configures this project to be open-source and maintenance is done by "LISA group". Theano acts as a compiler and makes use of NumPY for converting codes from different structures. Mathematical operations are being performed in Python with the help of the compiler before performing such transformations. Parallelism using multiple GPU-data is performed through extensions that are supported by Theano and hence helps in the development of cross-platform and open source projects. Looping control is supported through symbolic API, which helps in the efficient implementation of RNNs.

4.2.11 Performance-wise preliminary

There are basically two issues that are required to be focused on a large scale which is related to Dl frameworks' performance which is run-time performance and model performance. Considering model performance interest at the top, run-time performance is issued with severe efforts for benchmarking and hence making comparisons. The activities which are done are as follows.

- DL frameworks are being compared with or without Gluon and Keras.
- Backend of Keras are being tested.
- Figuring out the difference between PyTorch and Keras.
- Using CNNs with numerous Dl frameworks for benchmarking.
- Dl frameworks are being revised.

4.2.12Deep Learning wrapper libraries

Keratosis the wrapper library for the DL libraries which is intended for hiding the lower level implementations which are already elaborated above. The other wrapper libraries are:

a) TensorFlow

TensorFlow contains a huge number of wrappers, including internal and external wrapper packages. The wrappers in the native TensorFlow are tf.slim, tf.keras, tf.layers and tf.contrib.learn.

b) Gluon

It is the wrapper for MX Net. The API specification of Gluon consists of the effort for improving the flexibility, speed and the accessibility in DL technologies for all the developers having the choice of DL framework.

c) Nvidia Digits

It is the web application to train the DNN in classifying the images, segmenting and object detecting tasks through DL backend like Caffe, Torch and TensorFlow having wider varieties in image sources and formats.

d) Lasagne

It is a lightweight library for building and training the NN containing six different principals, i.e. "Simplicity, Transparency, Modularity, Pragmatism, Restraint and Focus".

4.3 Machine Learning and Deep Learning frameworks and libraries with MapReduce

Nowadays, the emerging distributed frameworks are introduced for addressing the scalabilities in algorithms for analyzing the big data using the "MapReduce programming model" containing the popular implementation like Apache Hadoop and Apache Spark. The crucial advantage of the system is the capability of reliability and scalability in a very user effective way. It is responsible for providing the users along with easier and automated distribution of fault-tolerant workload without mentioning the inconveniences in underlying the hardware architecture in a cluster. These technologies can be compliments of each other and responsible for targeting the scopes in complimentary computing like DL and ML.

4.3.1 Deeplearning4j

Deeplearning4j is different from the frameworks provided by ML/DL in spite of its integration, intent and API languages. It is detected as one of the modern distributed and open source library implementations in Java aiming to word the development of the industrial java Ecosystem and processing of big data. Deeplearning4j framework consists of integrated GPU support, which is one of the essential features in the process of training and supporting Hadoop and Spark. It is responsible for supporting several formats in inputting the data extensively by using specialized formats. DL4J consists of some primary NLP tools like tokenizer, sentence iterator and vocab.

4.3.2 Apache Spark MLlib and Spark ML

The introduction of Apache is conducted based on MapReduce. Mahout came with various types of ML algorithms. Besides that, ML algorithms required several limitations to make Mahout slower generally. Hence, spark MLlib and spark ML was introduced by Apache having the spark ecosystem, which is much faster than Mahout.

4.3.3 H2O, Sparkling Water and Deep Water

H2O.ai is responsible for developing the H2O, sparkling water and deep water. These are much compatible with Hadoop framework in ML and DL containing prediction analysis of big data. H2O uses the distributed key for accessing and referencing the data models and objects throughout the machine and nodes. DL is dependent on the FFNN in H2O trained with SGD ``stochastic gradient descent" by using back probation. Sparkling water also consists of similar functionality and features by providing the usage of H2O with a spark. Deepwater is H2O DL having the native implementation strategies of DL models to the backend of optimized GPU like MXNet, TensorFlow and Caffe.

4.3.4 Other frameworks and libraries with MapReduce

The remaining libraries and framework working with MapReduce are:

a) FlinkML

It is a part of Apache Flink containing the open-source framework in the distributed stream having the aim for providing the set of robust ML algorithms and APIs adopted in the Flink distributed framework.

b) Oryx 2

It contains the ML layer along with realization of Lambda architecture which is built with Apache Spark and Apache Kafka for the real-time large scale ML.

c) KNIME

It is the platform for analytic data integration and reporting the components for DM and ML having the modular concept of data pipelining.

5. Conclusions

It is evident that deep learning and ML consist of a huge research area in computer science containing regular developments because of its advance in the research 0f data analysis, including big data. This work is responsible for providing the effective elaboration and in-depth comparison of various types of frameworks and libraries which exploit the larger scale dataset effectively.

References

[1] S.Aridhi and E. MephuNguifo, "Big Graph Mining: Frameworks and Techniques", *Big Data Research*, vol. 6, pp. 1-10, 2016. Available: 10.1016/j.bdr.2016.07.002.

[2] A.Gittens et al., "Alchemist: An Apache Spark ⇔ MPI interface", *Concurrency and Computation: Practice and Experience*, 2018. Available: 10.1002/cpe.5026.

[3] S. Nag, H. Thakur, R. Shrivastava and A. Gupta, "Multi-class instance-incremental framework for classification in fully dynamic graphs", *International Journal of Computational Science and Engineering*, vol. 1, no. 1, p. 1, 2017. Available: 10.1504/ijcse.2017.10016991.

[4] H. Li, H. Liang, Q. Hu, M. Wang and Z. Wang, "Deep learning for position fixing in the micron scale by using convolutional neural networks", *Chinese Optics Letters*, vol. 18, no. 5, p. 050602, 2020. Available: 10.3788/col202018.050602.

[5] M. Muhammed, O. Waeal J and R. Bijan, "Applying Deep Learning Techniques for Big Data Analytics: A Systematic Literature Review", *Archives of Information Science and Technology*, vol. 1, no. 1, 2018. Available: 10.36959/863/756.

[6] R. NEELAPU, G. LAVANYA DEVI and K. SRINIVASA RAO, "Deep learning based conventional neural network architecture for medical image classification", *Traitement du signal*, vol. 35, no. 2, pp. 169-182, 2018. Available: 10.3166/ts.35.169-182.

[7] E.Boachie and C. Li, "Big Data Processing with Apache Spark in University Institutions: Spark Streaming and Machine Learning Algorithm", *International Journal of Continuing Engineering Education and Life-Long Learning*, vol. 28, no. 4, p. 1, 2018. Available: 10.1504/ijceell.2018.10017171.

[8] K. Chen, E. Cofer, J. Zhou and O. Troyanskaya, "Selene: a PyTorch-based deep learning library for sequence data", *Nature Methods*, vol. 16, no. 4, pp. 315-318, 2019. Available: 10.1038/s41592-019-0360-8.

[9] R. Van Noorden, "ArXiv preprint server plans multimillion-dollar overhaul", *Nature*, vol. 534, no. 7609, pp. 602-602, 2016. Available: 10.1038/534602a.

[10] B. Prasad and S. Agarwal, "Stream Data Mining: Platforms, Algorithms, Performance Evaluators and Research Trends", *International Journal of Database Theory and Application*, vol. 9, no. 9, pp. 201-218, 2016. Available: 10.14257/ijdta.2016.9.9.19.

[11] S.Ramasamy Ramamurthy and N. Roy, "Recent trends in machine learning for human activity recognition-A survey", *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1254, 2018. Available: 10.1002/widm.1254.

[12] S.Scardapane and D. Wang, "Randomness in neural networks: an overview", *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 7, no. 2, p. e1200, 2017. Available: 10.1002/widm.1200.

[13] M. Sheng, A. Vasilakos, Q. Yu and L. You, "Guest Editorial: Big Data Analytics and the Web", *IEEE Transactions on Big Data*, vol. 2, no. 3, pp. 189-189, 2016. Available: 10.1109/tbdata.2016.2613758.

[14] L. Wang and Y. Zhan, "A conceptual peer review model for arXiv and other preprint databases", *Learned Publishing*, 2019. Available: 10.1002/leap.1229.

[15] S. Zhang, L. Yao, A. Sun and Y. Tay, "Deep Learning Based Recommender System", *ACM Computing Surveys*, vol. 52, no. 1, pp. 1-38, 2019. Available: 10.1145/3285029.

[16] K. Chen, E. Cofer, J. Zhou and O. Troyanskaya, "Selene: a PyTorch-based deep learning library for sequence data", *Nature Methods*, vol. 16, no. 4, pp. 315-318, 2019. Available: 10.1038/s41592-019-0360-8.

[17] Y. Tang, "TF.Learn: TensorFlow's High-level Module for Distributed Machine Learning", *arXiv.org*, 2020. [Online]. Available: https://arxiv.org/abs/1612.04251. [Accessed: 03- Aug- 2020].

[18] *Arxiv.org*, 2020. [Online]. Available: https://arxiv.org/pdf/1812.00073.pdf. [Accessed: 03- Aug- 2020].

[19] *Elib.dlr.de*, 2020. [Online]. Available: https://elib.dlr.de/124423/1/Large-Scale%20Machine%20Learning%20with%20the%20HeAT%20Framework-WaW3.pdf. [Accessed: 03- Aug- 2020].

[20] T.Doleck, D. Lemay, R. Basnet and P. Bazelais, "Predictive analytics in education: a comparison of deep learning frameworks", 2020.