International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

Intelligent Data Processing and Automation in End-to-End Machine Learning Pipelines

Putty Srividya[1], Dr N. Subramanyam[2]

[1]Research Scholar, Annamalai University, Annamalai Nagar, Chidambaram - 608002, Tamil Nadu, India
[2]Professor, Department of CSE, Annamalai University, Annamalai Nagar, Chidambaram - 608002, Tamil Nadu, India

Abstract: ***Data processing plays an essential role in the development and performance of machine learning and deep learning models. Traditional machine learning relies on separate, manually designed stages for data preprocessing, feature extraction, feature selection, and model optimization. Deep learning offers a more integrated approach by using neural networks to learn useful features directly from data. However, the quality, quantity, and complexity of training data continue to influence model performance. Data preprocessing, augmentation, and feature engineering remain necessary to address missing values, noisy data, class imbalance, and limited training samples. Feature engineering also helps improve model efficiency and interpretability, particularly in applications such as healthcare, fraud detection, and industrial fault diagnosis. Automated Machine Learning (AutoML) provides methods for combining these tasks into a unified process, reducing manual intervention in data preparation, model selection, and hyperparameter optimization. Neural Architecture Search further supports the automatic design of suitable network structures. The discussion examines the differences between traditional and deep learning pipelines, the continuing need for data processing, and the role of automation in developing efficient, adaptable, and reliable machine learning systems.***

Keywords: Machine Learning, Deep Learning, Data Preprocessing, Feature Engineering, Automated Machine Learning

Data processing in traditional machine learning pipelines the machine learning problem typically involves the following sequence of steps: data collection, data preparation and preprocessing, data augmentation, extraction of useful features from the data, construction of new features and pruning of the generated feature set (i.e., feature selection), selection of a suitable machine learning model, and optimization of model hyperparameters. Traditional machine learning approaches [25,26] require all these stages of the machine learning process to be performed manually and as standalone processes (Fig. 5). This approach presents number of difficulties that make it extremely challenging to scale up machine learning models. For instance, traditional machine learning pipelines typically incorporate hand-crafted feature extractors to mine useful low-level features from training data for subsequent processing by the machine learning model. These feature extractors are basically case-specific algorithms aimed at finding the best set of features for the target task. Feature extractors, once designed, are typically fixed– i.e., in the process of learning, they cannot be modified automatically to work differently in response to new conditions. To adjust their operation, the algorithm may have to be redesigned and deployed anew.

Data processing in deep learning pipelines with modern deep learning models based on neural network frameworks [27,28], data processing tasks are usually implemented within the machine learning framework and trained end-to-end (Fig. 6). Deep learning approaches employ neural networks that are generally able to extract useful low-level information from raw data without the need for prior knowledge or additional processing.

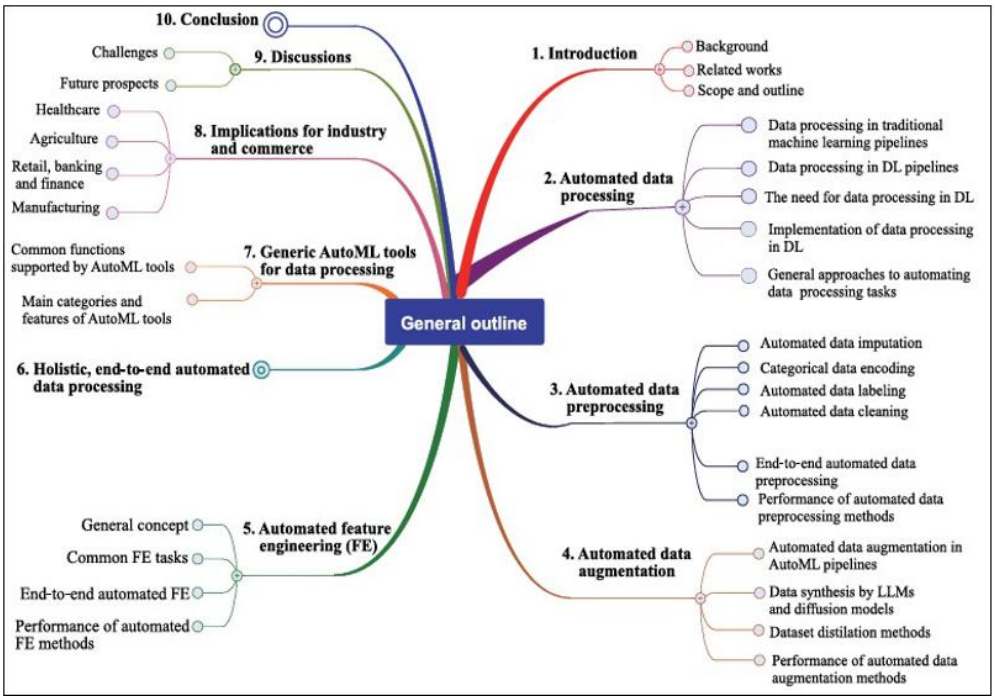


Figure 4: Broad Outline and Structure of this paper

Information processing in deep learning pipelines roughly mimics the information processing properties of the brains of biological systems, where different semantic features are extracted using a large and functionally diverse network of complex neural structures. Several hierarchical layers of artificial neural networks extract different types of features at different semantic levels. In the case of computer vision, for example, shallower layers extract useful information relating to more basic, low-level visual concepts such as lines, object contours and shapes. Deeper layers sequentially increase the abstraction level of extracted information, mining higher and semantically meaningful information such as composite shapes and whole objects.

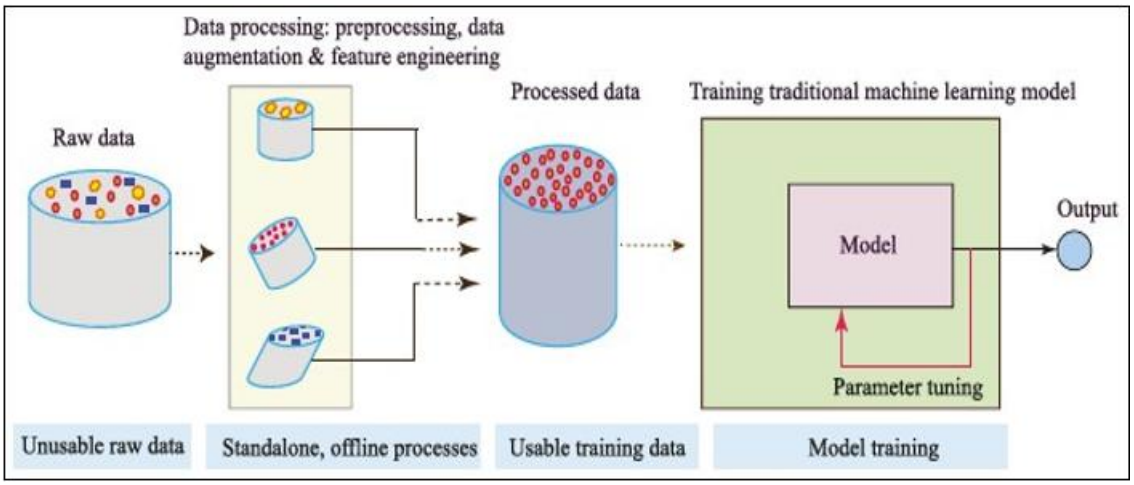


Figure 5: Data Processing in traditional machine learning. All processing steps are implemented manually as standalone tasks

Unlike traditional machine learning pipelines where fixed feature extractors are responsible for information processing, in deep learning pipelines, during training, the synaptic elements that extract useful features use tunable parameters to adjust their operation on the fly based on feedback on the performance of the network on the target task. Thus, in the learning process the model automatically adjusts its way of feature extraction and low-level data processing in response to performance results. The need for additional data processing in deep learning models.

The need for data preprocessing Despite the high degree of automation of data processing in deep learning models, their performance has been shown to be highly dependent on the quality of training data [29–31]. However, in many practical scenarios, the qualitative properties of raw data are not often consistent with the requirements of the target application or model [2,32]. Consequently, data preprocessing has become an essential task in the deep learning application development process. Preprocessing includes initial data preparation tasks such as data normalization, cleaning (removal of outliers and inconsistent data points), encoding of categorical values, imputation of missing values and labeling.

The need for data augmentation In many bigdata applications, the collected data may not be sufficient or representative enough for the target task. In such situations, after initial preprocessing, data augmentation is used to increase the quantity and variability of the training data by generating new

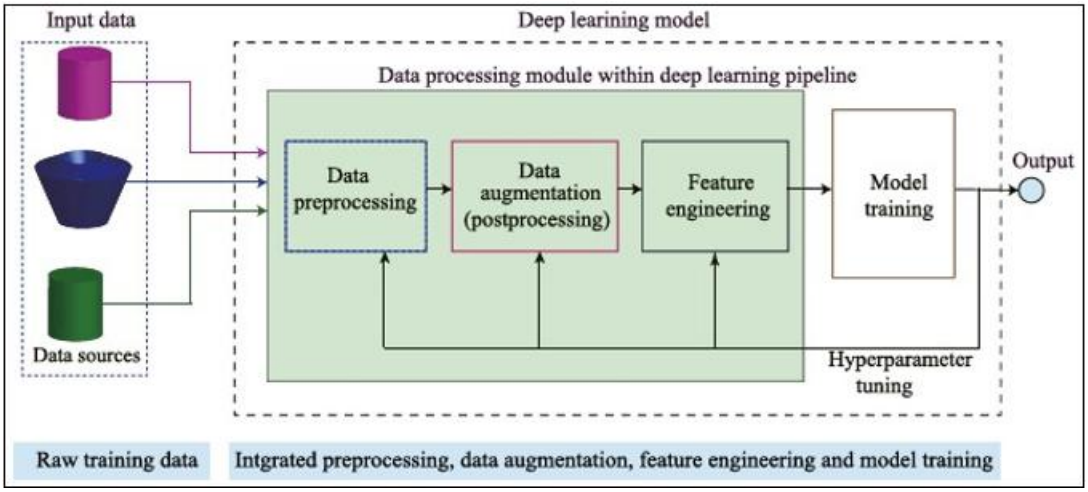


Figure 6: Data processing in deep learning: All processing tasks– preprocessing, data augmentation and feature engineering functions– as well as model generation and training are realized as a unified process samples through manipulating the existing data. This is the most popular approach to enhancing the generalization power of deep learning models. Data augmentation has also been used to addressed problems such as bias [33], imbalanced data [34] and domain shift [35].

The need for feature engineering While machine learning models constructed on the basis of deep neural architectures

can automatically learn useful features [36], in some application settings these methods still require explicit feature processing to guarantee satisfactory performance. This is particularly so for complex and data-scarce applications such as fault diagnosis in machinery (e.g., in Ref. [37]), online fraud detection (e.g., Ref. [38]), credit risk evaluation [39] and some industrial applications [40]. Indeed, it has been shown in Ref. [41] that employing dedicated feature extractors in deep learning networks provides more useful feature representations of the input data that results in significant performance improvements. Also, for big data applications, datasets are generally very large, heterogeneous, complex and noisy, leading to the need to handle very large volumes of features, often with redundant and duplicated samples. These intricacies may complicate the training process and lead to undesirable outcomes that can be harmful to performance [3, 42]. This problem can be mitigated by feature engineering methods, which aim to find compact feature representations for the data. Another common shortcoming of features naturally learned in the usual process of training deep neural networks is that they are typically not interpretable. This can be a major limitation in mission-critical application settings such as medical domains [43], where the decisions of deep learning systems need to be understood and validated by human experts. Feature engineering is also useful for tasks where training data is insufficient to adequately encode the desired information [44].

Implementation of data processing functions in deep learning pipelines Unlike in shallow learning where data processing is performed offline as a standalone process, modern deep learning favors an end to-end learning paradigm that aims at realizing all functions in an integrated manner using a single model. This means in deep learning, additional data processing, when needed, is usually integrated into the machine learning process and implemented as a unified process (Fig. 6). The fact that this process is carried out online means that no additional data samples are stored. This has the advantage of increasing the volume and diversity of training data without a corresponding increase in the amount of storage requirements. Instead of applying explicitly-designed transformation operations to create additional data, many new works (e.g., Refs. [45–50]) have devised special neural network architectures to learn the relevant transformations directly from the training data. In this case, the data processing tasks become an integral part of the overall learning process.

General approaches to automating data processing and feature engineering tasks Many different techniques are used to automate data processing tasks. These techniques have different strengths and limitations related to complexity, flexibility and predictive performance. While the crudest approaches to automating data processing tasks involve using hand-crafted routines, more advanced methods [51–53] utilize learned mechanisms to perform processing. The most advanced methods, however, employ automated machine learning (AutoML)techniques [12,54–56] to automate data processing functions. These methods can perform all the underlying data preprocessing tasks end-to-end without delegating intermediate tasks to human developers. A typical AutoMLpipeline for end-to-end data processing is shown in Fig. 7. The first part of the pipeline is concerned with data processing tasks. These include preprocessing, data augmentation and feature engineering. The second part of the AutoML pipeline performs functions such as model construction, hyperparameter setting, algorithm selection and result evaluation and validation. Additional functions such as result visualization (e.g., in Refs. [57,58]) may also be automated. In some cases (e.g., Ref. [59]), the automated machine learning problem involves composing an optimum model architecture to solve the machine learning task. Approaches for automating model structure construction belong to a class of automated machine learning methods called neural architecture search or NAS [60]. We describe the main features and functions of generic automated machine learning systems in Section 7.

References

[1] A. L'Heureux, K. Grolinger, H.F. Elyamany, M.A. Capretz, Machine learning with big data: Challenges and approaches, IEEE Access 5 (2017) 7776–7797.
[2] M.M. Najafabadi, F. Villanustre, T.M. Khoshgoftaar, N. Seliya, R. Wald, E. Muharemagic, Deep learning applications and challenges in big data analytics, Journal of Big Data 2 (1) (2015) 1–21.
[3] P. Rouzrokh, B. Khosravi, S. Faghani, M. Moassefi, D.V. Vera Garcia, Y. Singh, K. Zhang, G.M. Conte, B.J. Erickson, Mitigating bias in radiology machine learning: 1. data handling, Radiology: Artificial Intelligence 4 (5) (2022) e210290.
[4] S. Carta, A.S. Podda, D.R. Recupero, R. Saia, A local feature engineering strategy to improve network anomaly detection, Future Internet 12 (10) (2020) 177.
[5] S. Lu, X. Wei, Y. Li, L. Wang, Detecting anomaly in big data system logs using convolutional neural network, in: Proceedings of the 2018 IEEE 16th International Conference on Dependable, Autonomic and Secure Computing, 16th International Conference on Pervasive Intelligence and Computing, 4th International Conference on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), IEEE, Piscataway, 2018, pp. 151–158.
[6] S. Garcia, S. Ramirez-Gallego, J. Luengo, J.M. Benitez, F. Herrera, Big data preprocessing: Methods and prospects, Big Data Analytics 1 (1) (2016) 1–22.
[7] J. Waring, C. Lindvall, R. Umeton, Automated machine learning: Review of the state-of-the-art and opportunities for healthcare, Artificial Intelligence in Medicine 104 (2020) 101822.
[8] G. Liu, D. Lu, J. Lu, Pharm-AutoML: An open-source, end-to-end automated machine learning package for clinical outcome prediction, CPT: Pharmacometrics and Systems Pharmacology 10 (5) (2021) 478–488.
[9] J. Drozdal, J. Weisz, D. Wang, G. Dass, B. Yao, C. Zhao, M. Muller, L. Ju, H. Su, Trust in AutoML: Exploring information needs for establishing trust in automated machine learning systems, in: Proceedings of the 25th International Conference on Intelligent User Interfaces, ACM, New York, 2020, pp. 297–307.
[10] Z. Li, H. Guo, W.M. Wang, Y. Guan, A.V. Barenji, G.Q. Huang, K.S. McFall, X. Chen, A blockchain and AutoML approach for open and automated customer

service, IEEE Transactions on Industrial Informatics 15 (6) (2019) 3642–Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), IEEE, Piscataway, 2018, pp. 151–158.
[11] S. Garcia, S. Ramirez-Gallego, J. Luengo, J.M. Benitez, F. Herrera, Big data preprocessing: Methods and prospects, Big Data Analytics 1 (1) (2016) 1–22.
[12] J. Waring, C. Lindvall, R. Umeton, Automated machine learning: Review of the state-of-the-art and opportunities for healthcare, Artificial Intelligence in Medicine 104 (2020) 101822.
[13] G. Liu, D. Lu, J. Lu, Pharm-AutoML: An open-source, end-to-end automated machine learning package for clinical outcome prediction, CPT: Pharmacometrics and Systems Pharmacology 10 (5) (2021) 478–488.
[14] J. Drozdal, J. Weisz, D. Wang, G. Dass, B. Yao, C. Zhao, M. Muller, L. Ju, H. Su, Trust in AutoML: Exploring information needs for establishing trust in automated machine learning systems, in: Proceedings of the 25th International Conference on Intelligent User Interfaces, ACM, New York, 2020, pp. 297–307.
[15] Z. Li, H. Guo, W.M. Wang, Y. Guan, A.V. Barenji, G.Q. Huang, K.S. McFall, X. Chen, A blockchain and AutoML approach for open and automated customer service, IEEE Transactions on Industrial Informatics 15 (6) (2019) 3642–3651.
[16] A. Crisan, B. Fiore-Gartland, Fits and starts: Enterprise use of AutoML and the role of humans in the loop, in: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, ACM, New York, 2021, pp. 1–15.
[17] X. He, K. Zhao, X. Chu, AutoML: A survey of the state-of-the-art, Knowledge-Based Systems 212 (2021) 106622.
[18] M.-A. Zo€ller, M.F. Huber, Benchmark and survey of automated machine learning frameworks, Journal of Artificial Intelligence Research 70 (2021) 409–472.
[19] K. van der Blom, A. Serban, H. H Hoos, J. Visser, AutoML adoption in ML software, in: Proceedings of the 8th ICML Workshop on Automated Machine Learning, 2021, pp. 1–13.
[20] U. Gain, V. Hotti, Low-code AutoML-augmented data pipeline - A review and experiments, Journal of Physics: Conference Series 1828 (2021) 012015.
[21] A. Alsharef, K. Aggarwal, M. Kumar, A. Mishra, Review of ML and AutoML solutions to forecast time-series data, Archives of Computational Methods in Engineering 29 (7) (2022) 5297–5311.