Informatics and Control Problems 43 Issue 2 (2023)
journal homepage: www.icp.az

Application of AI in software engineering: handling data management problems in production

G.A. Abdiyeva-Aliyeva

Institute of Control Systems, Baku, Azerbaijan

ARTICLE INFO

Article history:
Received 25.05.2023
Received in revised form 08.06.2023
Accepted 14.06.2023
Available online 20.09.2023

Keywords:
Machine learning
Software products
Distributed systems
Hadoop
Big data

ABSTRACT

With the help of Machine Learning, modern business companies have eased the processes within the business contexts. Moreover, having built artificial intelligence-based software products, effectiveness in not only automated processes but also quick monetary returns are attained. Since the amount of data is increasing sharply, building models and automate them within the software products have been one of the main challenges that business institutions face. In spite of the advanced high quality computers, processing those data, building advanced Machine Learning (ML) models on them and deploying those models within the software products are problematic points. Maintaining the lifecycle of Artificial Intelligence based software products after deployment is another point that business institutions are trying to solve in an optimal way. The paper aims to analyze the context and variety of applications of Artificial Intelligence (AI) in Software Engineering and discuss possible problems arising from these applications.

1. Introduction

In the contemporary period, digitalization of the processes has been leading to the increasing volume of the data which in turn aims to be utilized in industry levels. As a result of this increasing volumes, several projects in particularly Information Technology area are getting advanced every year. Even, today, due to digitalization, computers can be called as the one that is able to think and decide about the tale of the future based on processing the historical data and memorizing the patterns which humans sometimes fail to catch. The advantageous side of computers emerged with the magic of math and computers science which come together in AI and ML algorithms. However, despite the considerable importance of data-driven and ML based software platforms, still, the quality of the data and real-time prediction are one of the problematic points in deployed AI based digital products. Moreover, storing that huge amount data and processing the calculations, i.e. building pipelines after deployment become a great obstacle in pieces of software in deployment process.

The advanced mathematical calculations processed on a huge amount of data applied in natural language processing, computer vision, voice recognition and statistical prediction, and parameter estimation using variety of Gradient descent are the foundational focus of ML algorithms. The huge increase of data over the past decades lead ML to prevail in both industry and academia [1] and it processes such great computations that are beyond the human's capability. Especially, in the era of

E-mail address: gunay.abdiyeva@isi.az (G.A. Abdiyeva-Aliyeva).

www.icp.az/2023/2-12.pdf https://doi.org/10.54381/icp.2023.2.12

Big Data, Deep Learning models sometimes performs high computations which is sized by terabytes that in turn becomes very hard to process and deploy in software production. Handling such huge computations on single hardware resources is really a great headache for both experts in data fields and machines. Moreover, maintaining the success states in different stages of data pipelines in such resourceful products in deployment of software products become one of the major discussions of scientific research. Despite the existence of distributed systems that shed light on the responsibility for data management of software products, applications that use AI still struggle with data management due to the large amount of data stored. "A distributed system is a system that allows different components of different machines to communicate with each other and coordinate tasks, representing a single, consistent system for the end user" [1]. The ML distributed system creates a multi-node ML system. The system actually creates a high-quality environment in terms of higher accuracy, faster performance, and scaling to larger input data sizes, thereby reducing machine errors [2]. Based on this factor, analysis of large datasets provides more accurate results. In particular, cloud services such as Google Cloud, Amazon AWS, etc. [3] are the main practical ways to allocate virtual hardware resources over the network which decreases the overload on a single server machine. Despite the advantages of ML in distributed system, the fact that the speed of information exchange depends on the limited bandwidth of the communication network can lead to terrifying consequences. When one machine can fail at any time and the job can be prevented [3]. "The reliability of such systems deteriorates as the number of machines and workload size increases. These characteristics of distributed systems make for efficient and reliable machine learning applications with distributed computing. It poses a major challenge to development" [4]. In general, efficiency, fault tolerance, and ease of use can be seen as the most important and important characteristics of ML in distributed computing system. In the age of big data, the amount of data is one of the biggest challenges. Therefore, optimal use and storage of such data for both analytical purposes and descriptive and predictive statistics has been a major goal of scientific and technological research. The paper aims to analyze what problems occur in deployment of AI based applications, such as chatbots and business intelligence tools.

2. Background

A simple definition of Artificial Intelligence would be that it is the science of mimicking human mental faculties on a computer [5]. Machine Learning is a subset of Artificial Intelligence which concerns about improvement of computers to adapt learn the patterns of historical data and predict the future. There are 3 main types of ML algorithms:

- Supervised;
- Unsupervised:
- Semi-supervised.

Supervised ML algorithms requires labeled datasets to be trained to find the relative coefficients. The input values are trained in a way that the difference between prediction and actual values gets minimum in both training and testing stages. Based on the provided labels alongside input values, the computers target to find relationships between prediction and input features. Regression are Classification are one of the problems in ML to be solved with supervised approach. Extreme Gradient Boosting (XGBoost), CatBoost, Linear Regression, Logistic Regression, Naïve Bayes are one of the significant supervised learning algorithms of ML. On the other hand, unsupervised machine learning does not require a labeled dataset. Clustering and Dimension Reduction problems requires unsupervised machine learning algorithms. K-Means, Hierarchal Clustering, PCA., etc. are the unsupervised learning algorithms widely applied in industry. Furthermore, semi-supervised learning refers to learning problem that requires a small portion of labeled dataset and large unlabeled datasets from which model will target to learn and produce predictions on new unseen datasets. In other words,

semi-supervised learning can be considered as the combination of both supervised and unsupervised learning. In image recognition, voice detection problems, semi-supervised learning produces an effective results as pseudo labeling decreases the workload in the data labeling manually. The researchers examined that deployment of the ML models consists of generally 9 main steps (Fig. 1) [6].

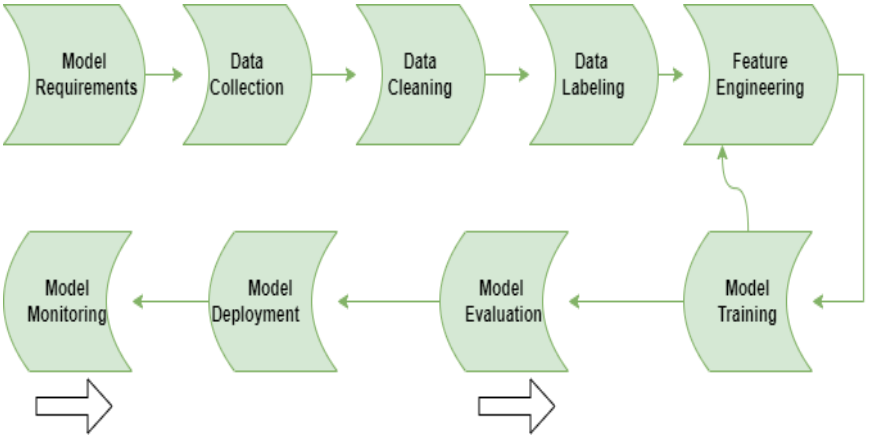

Fig 1. The nine stages of the machine learning workflow

In the Fig 1., "some stages are data-oriented (e.g., collection, cleaning, and labeling) and others are model-oriented (e.g., model requirements, feature engineering, training, evaluation, deployment, and monitoring)". There are many feedback loops in the workflow. The larger feedback arrows denote that model evaluation and monitoring may loop back to any of the previous stages. The smaller feedback arrow depicts that model training may loop back to feature engineering [6].

"The need for adjusting software engineering practices in the recent era has been discussed in the context of hidden technical debt and troubleshooting integrative AI" [7]. This work identifies certain ML system architectural needs that must be taken into account while designing the system. Hidden feedback loops, component entanglement, eroding boundaries, continuous quality states, nonmonotonic error propagation, and mismatches between the actual world and evaluation sets are a few of these elements. In the past five years, several initiatives have been made in industry to automate procedures by creating frameworks and settings that allow the experimental workflow of ML. Engineers continue to have difficulty operationalizing and standardizing work procedures, according to continuing studies and polls. "To ensure that system deployment goes smoothly, several engineers recommend not only to automate the training and deployment pipeline, but also to integrate model building with the rest of the software, use common versioning repositories for both ML and non-ML codebases, and tightly couple the ML and non-ML development sprints and standups" [7]. Based on these factors, software products with ML and AI models does not solely require the simple maintenance, instead it demands to monitor the model, retraining process and move the data with the processes which in turn even in the 21st century with technology advancements, it is still a great issue in business institutions.

3. Solution problems with AI-based software products

a. Customer Segmentation Case

In several institutions, such as retail banking, telecommunications, etc., the customer segmentation or clustering are considered as one of the main applications to draw the business states further in terms of monetary and customer personalization aspects. Business analysts usually require having an application automatically clustering the customers using ML algorithms and describe the insights behind each of centroids of the clusters. In order to create such products, usually, customers'

transactional data are utilized which in turn sometimes reaches millions in a day. To handle such a large data, clustering customers in every request of the user on the platform, maintaining the training process after deployment and building the data pipeline should be built, however, these are quite challenging steps that makes application of AI & ML in software engineering difficult.

One of the applications named as RFM Clustering requires these steps to be completed successfully to fill such business needs. RFM stands for respectively recency, frequency and monetary value of a customer within the specified period.

- Recency (R) – it is the difference of time units between first and last transaction date;
- Frequency (F) – it is the number of repeated purchases in the time unit. In other words, it is the count of time periods that the customer had transactions;
- Monetary Value (M) – it is the sum of amount that the customer has spent within the selected time period.

In every request on the platform, the program should read the whole dataset of customers and cluster them based on RFM values in different categories, such as card type, purchasing behavior, or both. Whereas one user would like to cluster the customers frequently going to grocery stores, at the same time, another one may wish to observe the customers often buying clothes only with credit cards. Due to great variety of joints in users' requests, the application should be built in a dynamic format where each time a model should be trained to produce the clusters based on the user request. Because of high computations of clustering algorithms on large-scaled dataset, such as K-Means, Hierarchical clustering, sometimes user may have to wait for half an hour to get the results back. Therefore, it is possible to face issues such as automatic disconnect from the server in such products due to high computational resource requirements.

Reading data from different resources, opting the data based on the user selection and preparation of the data for modelling are the most resourceful stages of this process. In the next phase, the number of the cluster should be automatically determined in different ways, such as optimal silhouette scores, Elbow methods, etc. Followingly, one of the clustering algorithms, namely K-Means, starts to cluster the data based on the mathematical calculations on each customer's features. Performing these steps in each request create a crucial challenge for both developers and data scientists as it really becomes hard to meet in a point that satisfies the software products from both data and development wise. Therefore, automated customer segmentation/clustering using real-time analytics is one of the challenging problems for institutions in terms of computation and time complexity of algorithms in real time production.

b. AI chatbots

In the contemporary period, AI agents, such as chat bots, are one of the most important products in business areas, especially bank and telecommunication sectors as the number of customers of such institutions is quite huge. Handling those customers' requests or complaints sometimes becomes quite a hard challenge to overcome by humans. Therefore, automated chatbots representing the call centers artificially in production help the institutions to overcome on huge number of customers messages in time which also increase the customer satisfaction level directly linked to the monetary profit of the businesses.

AI bots is the Natural Language Processing branch of Data Science, where textual data is collected, cleaned, labeled and trained in the models to calculate the coefficients. One of the main problems with the chat bots in software production is that at the same time, a single bot should be available for every customer and reply to their questions or constraints accurately. In case of the failing to understand the customer's need, the bot should automatically transfer message to the humans. The main problems arise in transferring the messages as in some cases, chatbot channels may not support this stage. Additionally, the software products deployed with AI based chatbots have

to store the conversation history with conversation id and user id so that monitoring of the model can be possible after the deployment. Beside monitoring the model, storing the historical conversation can help retraining process of the model and adjustment of new types of labels in the conversations. However, all of these retraining processes are currently being implemented manually. In the production, the pipelines should be built in a way that the data scientists should configure the retraining process high level after once they trained the model in the first time. The errors should be monitored and adjusted accordingly in the retrained model.

In terms of ML algorithms, Naïve Bayes, Decision Trees, Recurrent Neural Networks, Long Short-Term Memory algorithms are believed to perform the best. However, a single model in deployment within the software products should be attached in a way that retraining and monitoring processes can be possible based on the stored data. Additionally, sometimes chat-bots could provide completely wrong replies, stick in the loop and do not transfer the message to the humans. In these cases, the customer in the online service on software product may fail to get the appropriate answer. Considering these factors, AI based chat-bots might cause problems with retraining process of machine learning models and storing historical data despite the advantages of speeding up customer-business communication.

4. Ecosystem in scalable machine learning products

In the modern world, professional companies and organizations consider Hadoop ecosystem as simple and efficient model in terms of better performance in working with large-scaled dataset. Apache Hadoop comprise of 5 different daemons and each of these daemons run its own Java Virtual Machine (JVM) [8].

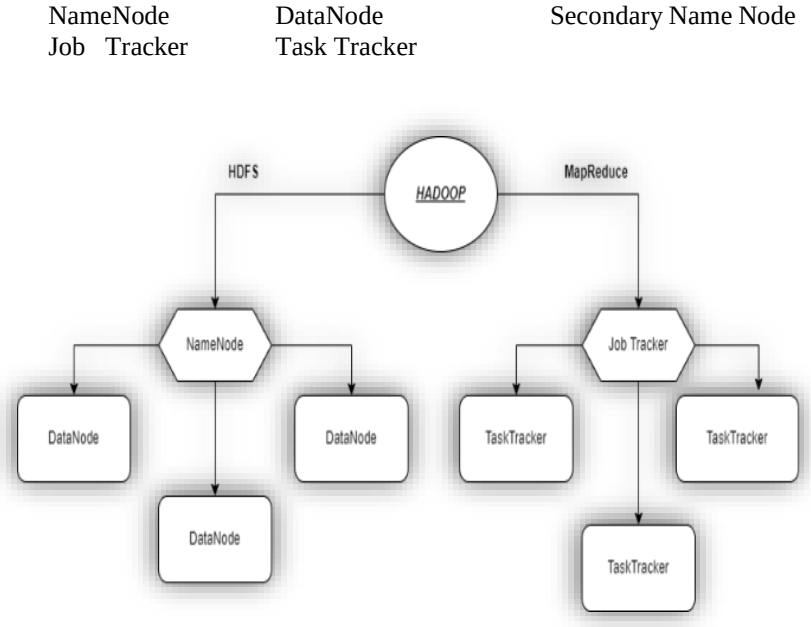

Fig 2. Hadoop Daemons

Hadoop is a platform that is generally considered to be highly scalable. It has the ability to store very large datasets and distribute them to multiple servers running in parallel [8] that in turn is considered as main advantage of the ecosystem. And for large processing datasets, Hadoop supports a cost-effective and effective storage solution. In other words, as a scaled architecture, Hadoop can store all of your business organization's data for later use. In addition, fault tolerance is considered the main reason for using Hadoop. In particular, when data is sent to each node, the data is copied to

the other nodes; therefore, completion of any process is guaranteed in advance. It is because that if an error occurs, another node is able to continue the process. Hadoop, on the other hand, applies Kerberos principles for security which in turn makes the management of it quite hard. Considering the fact that Kerberos lacks storage and network encryption, in terms of security, Hadoop could fail in providing secure data environment for business institutions.

In addition, "Hadoop works efficiently with a small number of large files. Hadoop saves files in the form of file blocks with sizes from 128MB (default) to 256MB" [8]. Many of these small files overload the namenode, making it more difficult to work with. Based on these factors, Hadoop is one of the foundations on storing large-scale dataset in the contemporary working environments.

Hadoop speeds up the analysis and calculation on data which is what makes data most valuable for business entities to draw the processes further in terms of analysis and prediction models. Moreover, Hadoop prevents the delay that occur in the process of bringing data to processing centers. The key point in Hadoop is that instead of moving data to process, the process is moved to data.

Deploying a Scalable Machine Learning Model consists of several steps. First of all, the data should be attained from different sources, such as databases, data lakes, etc. In the next stage, it should be cleaned, well prepared for model. Building a machine learning model requires high computational tasks, such as mainly gradient descent, stochastic gradient descent, etc. Having finalized the model, one of the major steps is to convey the model to the users to automatically perform their analysis, or predictions. In this stage, all data engineers, data scientists and developers should work co-operatively as deployment of the ML models within the software products has several challenges, such as building right data flows, pipelines, etc. To support the ideas of scaling, flowless machine learning model train process on data of different sizes data scientists and engineers uses a diverse range of software ecosystems and tools such as Hadoop, Apache Spark, HDFS, Map-Reduce Paradigm [9]. When a model is ready to be deployed, it can be packaged as a web service and delivered to a scale-out supported cloud, for example, an enterprise machine learning execution environment like Kubernetes, or Cloudera using Apache Spark.

Hadoop can be very useful in processing the huge computations on large scale dataset. In the product for business analysis, considering the huge variety of input selection, some software products require dynamic analytics. Therefore, instead of moving data to the front of the software products, the request can be sent to the servers via network and Hadoop can perform relative analytics based on the user's preferences. However, in real time analytics, Hadoop should not be considered directly due to Time Lag. First, Hadoop needs to accumulate data from the file in batches and then process each batch one by one. The process of accumulation is time-consuming. Thus, if the Software System needs to listen to incoming events and process them immediately, then Hadoop is not a good choice. One of main advantageous side of Hadoop ecosystem is that it can store all types of data such as structured, semi-structured, as well as unstructured data in vast amounts. Since the performance on data is more efficient with Hadoop ecosystems, the cleaning and feature extractions can be also implemented within the ecosystem. One such example is the Hadoop Tool called Apache Spark Structured Streaming. As soon as a stream is generated Spark can process it. The reason for Hadoop Jobs being slower than Spark is that Hadoop Jobs works directly with the HDFS file system. Therefore, for processing data first it needs to read it from the file on the disk, while Spark does the computation that is available on the memory leveraging RDD. Additionally, Hadoop should not be used with small datasets, because it can be costly for a lot of small datasets rather than the bigger ones. On the other hand, if it is needed to have the data live and run forever, then Hadoop can help with that using its Scalability features, but there are several tools on the web that can be put on top of Hadoop to secure the data. One specific example of a potential security problem in Hadoop is if there is a need to implement a system that makes the data visible based on the role of the user, then it can bring some challenges [10]. Generally, if the data that the system is dealing with weighs terabytes,

then Hadoop is the right tool to go for. The easy-to-scale-hardware ability allows Hadoop to flexibly manage quickly scaling amounts of data.

Based on these findings, it is believed that Hadoop ecosystem enables user to perform advanced analytics using lower operating expenses and capital expenses. In order to add value by extracting meaningful insights from datasets, Hadoop ecosystem is quite useful to apply. Therefore, in case of analyzing large-scaled dataset, performing advanced mathematical calculations, Hadoop can be considered as the main ecosystem.

5. Conclusion

To sum up, the application of artificial intelligence in software engineering is becoming one of the main targets in different business institutions in order to build high quality services and advanced systems. Considering the fact that every year the amount of data is unbelievably increasing, in the contemporary period, companies reached such a level that analyzing customer's personal data alone and providing customer-personalized approach is the initial step of data science, rather they started to analyze web-site or any mobile apps clicks. However, services maintaining the lifecycle of software products, even with the high technical parameters, could sometimes fail to provide higher efficiency in making advanced mathematical computations on those highly scaled datasets. AI based software products, such as automated segmentation, clustering and chat-bots, require advanced development skills to maintain lifecycle of the products.

Beside the technical challenges of the AI based software products, relying the services on AI agents could sometimes lead to less efficiency in business-customer communication. Therefore, the deployed models should also be monitored frequently and updated so that they work more effectively in reaching customers' need. Also, drifts in data should be considered in which as time passes, new datasets should be included in the training process of the model because with old time period data, prediction of future might produce unreliable results. As an example, since COVID19, the predictions and automated analytics have started to mislead in results; in order to prevent these, the data should be adjusted with recent time periods and these processes should be re-implemented. Based on these factors, application of ML models in software engineering does not solely require just a single built-in model, rather, it requires monitoring tools for data pipeline, right predictions and accuracy measures of model.

References

[1] M. Li, D. Andersen, J. Dean &amp, B. Poczos, Scaling distributed machine learning with system and algorithm Co-design. school of computer science carnegie Mellon University, (2017). https://www.cs.cmu.edu/~muli/file/mu-thesis.pdf

[2] A. Ulanov, A. Simanovsky &amp, M. Marwah, Modeling scalability of distributed machine learning, 2017 IEEE 33rd International Conference on Data Engineering(ICDE), (2017). https://doi.org/10.1109/icde.2017.160

[3] L. Bottou, F.E. Curtis &amp, J. Nocedal, Optimization methods for large-scale machine learning, SIAM Review, 60 No.2 (2018) pp.223-311. https://doi.org/10.1137/16m1080173

[4] A. Saleema, A. Begel, B. Christian, D. Robert, G. Harald, E. Kamar, N. Nichappan, B. Nushi, T. Zimmermann, Software engineering for machine learning: A Case Study, Microsoft Research, (2019). https://www.microsoft.com/en-us/research/uploads/prod/2019/03/amershi-icse 2019_Software_Engineering_for_Machine_Learning.pdf

[5] S. Khokad, G. Bhalerao, D. Daivashala, A study based on Cloudera s distribution of Hadoop Technologies for Big Data, International Journal of Advance Engineering and Research Development. 4 No.8. (2017). https://doi.org/10.21090/ijaerd.39918

[6] S. Maitrey, C.K. Jha, MapReduce: Simplified data analysis of Big Data, Procedia Computer Science, 57 (2015) pp.563-571. https://doi.org/10.1016/j.procs.2015.07.392

[7] M.I. Jordan, T.M. Mitchell, Machine learning: Trends, Perspectives, and prospects, Science. 349(6245) (2015) pp.255-260. https://doi.org/10.1126/science.aaa8415

[8] M.R. Ghazi, D. Gangodkar. Hadoop, MapReduce and HDFS: A developers perspective, Procedia Computer Science. 48 (2015) pp.45-50. https://doi.org/10.1016/j.procs.2015.04.108

[9] G. Abdiyeva-Aliyeva, J. Aliyev, U. Sadigov, Application of classification algorithms of Machine learning in cybersecurity, Procedia Computer Science. 215 (2022) pp.909-919. https://doi.org/10.1016/j.procs.2022.12.093.

[10] G. Abdiyeva-Aliyeva, "AI-Based Network Security Anomaly Prediction and Detection in Future Network," 2023 11th International Symposium on Digital Forensics and Security (ISDFS), Chattanooga, TN, USA. (2023) pp. 1-5. doi: 10.1109/ISDFS58141.2023.10131845.