A Vivid Analysis of Machine Learning Papers

Md. Efaz and Susmoy Biswas
efaz15-14193@diu.edu.bd
saukhin15-5647@diu.edu.bd

Abstract: Machine Learning (ML) has witnessed remarkable growth and innovation in recent years, becoming a pivotal technology with far-reaching implications across diverse industries and applications. This paper presents an extensive Analysis of some review papers on the field of machine learning .The review commences with a robust analysis on some achievement of papers done by some previous research done by some of proficient people of this field .As we traverse through them we can learn more about what they intended and want to improve as well. We will learn about CoReJava, Taxonomy and some other analysis done by them too.

Keywords: 1. Machine Learning 2.Remarkable growth 3.Achievement 4. Analysis 5.Outcome

---

1. Introduction:

Machine Learning (ML), a dynamic and ever-evolving field within the broader domain of artificial intelligence This paper embarks on a comprehensive journey through the intricate landscape of machine learning based on some papers which are given bellow, aiming to provide a panoramic view of its evolution, applications, recent advancements, and ethical considerations. As the field continues to expand and diversify, it becomes increasingly essential to discern its key components, navigate its complexities, and appreciate the profound implications it has for society. To provide a forward-looking perspective, we will scrutinize the latest trends and advancements within the papers their visions and achievements and impact on our society.

2. Method of Analysis:

The Method of analysis we used to create the review analysis consists of:

[1] Understanding the Paper's Context [2] Review and analysis of Introduction and Background for each paper [3] Evaluating Methodology and Experimental Design of the paper and its outcome [4] Checking for if the results are clearly and comprehensively presented through tables, figures, and charts.[5] Evaluating the statistical significance of results[6] Evaluating how well the results are interpreted and related to the research question[7] Checking that the research makes a novel contribution to the field [8] Assessing the overall organization and logical flow of the paper[9] Evaluating the clarity of explanations, algorithms, and equations[10]Then we constructed an efficient feedback for each papers that are discussed bellow

3. Results and Discussion:

Review on The [1] R package mlr3 and its associated ecosystem of extension packages authored by (R Core Team, 2019):

The creators of the R package mlr3 (R Core Team, 2019) and its ecosystem of extension packages builds a robust, flexible, object-oriented machine learning (ML) framework in R. It offers a unified interface to many learning algorithms that are available on CRAN, enhancing them with model-independent general-purpose functionality required in every machine learning project, such as train-test-evaluation, resampling, preprocessing, hyper parameter tuning, nested resampling, and visualization of ML experiment results. The package is an entirely new implementation of the mlr (Bischl et al., 2016) package, drawing on years of expertise and learnt best way to deliver a cutting-

edge system that is strong, versatile, expandable, and maintainable. We want to deliver researchers who want to develop, test, benchmark, and compare their novel approaches in a controlled environment as well as practitioners who want to swiftly deploy ML algorithms to their challenges. As a basis to develop new ML (meta-)algorithms (for example, Auto ML systems), mlr3 is ideal for small scripts that test a concept, complex multi-level experiments with sophisticated capability, and anything in between. Comprehensive unit and integration tests assure functional accuracy. There are also other general-purpose ML toolboxes available for various programing languages. The most populer ones are scikit-learn (Pedregosa et al., 2011) for Python, Weka (Hall et al., 2009) for Java, and mlj (Blaom, Kiraly, Lienart, & Vollmer, 2019) for Julia. The three toolboxes MIR, caret (Kuhn, 2008), and tidymodels (Kuhn & Wickham, 2019) are the most crucial ones for R.

Review on [2] Object detection in design with machine learning:

Paper [2] states that Businesses and companies have amassed a lot of historical data over the years. The value of the data is rising thanks to contemporary data mining and machine learning technology. Therefore, it's crucial to be able to transform legacy data into a format that computers can interpret. In this research, they look into the conversion of schematic designs like P&I diagrams (process and instrumentation diagrams). To find high-level objects in diagrams that have been scanned from paper archived database or generated in pixel or vector form, such as pumps or valves, they used modern machine learning based technologies, in particular, the Yolo neural network system. This is a crucial step for the reuse of outdated planning data, together with connection detection and OCR. Their findings demonstrate that Yolo, a contemporary object detection system based on machine learning, performs effectively with schematic designs. In their concept, they employ a simulated environment to produce labeled training materials for the system automatically. Then, they retrain a network that has already been trained to find the components of interest. Large components are accurately detected, however minor components with sizes less than 15% of the page size are overlooked. This study looked into the issue of creating a "intelligent form" out of schematic designs. Here, the term "intelligent" refers to a system that can recognize the items in a drawing and their relationships. Making the legacy data relevant for data mining, machine learning, and brownfield design activities requires this crucial step. They retrained a pre-trained Yolo neural network to recognize the components they are interested in using data produced with their own simulator and the pre-trained network. Although the trials were conducted with process and instrumentation diagrams from the process industry, the same methodology applies to other types of schematic diagrams, such as electrical circuit diagrams.

Review on [3] a machine learning toolbox named SHOGUN:

The paper's authors [3] they have created a toolbox for machine learning called SHOGUN that is intended for unifying large-scale learning for a variety of feature kinds and learning environments. It provides a broad selection of machine learning models, including linear discriminant analysis, hidden Markov models, multiple kernel learning, and support vector machines. The majority of the specialized algorithms can handle a variety of data classes. This toolkit has been utilized in various computational biology applications, some of which include at least 50 million training instances and others which have 7 billion test cases. SHOGUN has more than a thousand installations worldwide and is already well-liked both within and outside of the machine learning field. In addition to MATLAB, TM R, Octave, Python, and a standalone command line interface, SHOGUN is

implemented in C++. Under the terms of the GNU General Public License, the source code is freely accessible.

Review on [4] Elixir an effective object-oriented program repair:

On paper [4] the widespread usage of method invocations in object-oriented (OO) programs and, in fact, their predominance in patching of OO-program problems, are the driving forces behind this research. They suggest ELIXIR, a generate-and-validate repair method that is made to be able to produce such patches. Method calls are aggressively used by ELIXIR to create more expressive repair-expressions that are used to synthesize patches, on par with local variables, fields, or constants. By utilizing a machine-learnt model to prioritize concrete repairs, it is possible to efficiently address the resulting expansion of the repair space caused by the increasing use of method calls. Four features from the program context, or the code surrounding the potential repair location, and the problem report are used by the machine-learnt model. They put ELIXIR into practice and assess it using two datasets—the well-known Defects43 dataset and a brand-new one they produced called Bugs jar—as well as two baseline iterations of our technique and five other techniques that represent the state of the art in program repair Our analysis reveals that ELIXIR significantly outperforms the other 12 state-of-the-art repair techniques, including ACS, HD-Repair, NOPOL, PAR, and jGenProg, by increasing the number of correctly repaired bugs in Defects4J by 85% (from 14 to 26) and by 57% in Bugs.jar (from 14 to 22). They consider this work to be a promising example of how AI/ML methods might be utilized to broaden the use of automatic program repair methods. They intend to pursue and fully exploit this area of study in the future.

Review on [5] Object-based image classification of summer crops with machine learning methods:

The authors of [5] states that Crop field monitoring is a necessary part of managing agricultural fields strategically every year. Remote sensing-based crop discrimination is a challenging task, particularly when crops exhibit identical spectral responses and cropping patterns. In these situations, integrating object-based picture analysis with sophisticated machine learning techniques may help to enhance crop recognition. In this study, we assessed the performance of the C4.5 decision tree, logistic regression (LR), support vector machine (SVM), and multilayer perceptron (MLP) neural network methods for the mapping of nine major summer crops (both woody and herbaceous) from ASTER satellite images taken on two different dates. After the distant images were segmented in an object-based framework, various combinations of spectral and textural properties were acquired, and these features were used to build each technique. With only one classifier, MLP and SVM were able to outperform LR (86%) and C4.5 (79%), respectively, with a maximum overall accuracy of 88%. The best strategy, SVM+SVM, increased these outcomes to 89%. In the majority of situations, hierarchical classifiers significantly improved the accuracy of the class with the lowest classification sensitivity. For all of the crops under study, the SVM+SVM method significantly increased classification accuracy compared to the traditional decision tree classifier, with improvements ranging from 4% for safflower to 29% for com. This finding points to the potential use of advanced machine learning techniques and Eject-based image analysis in challenging crop classification tasks. The favorable results across a wide range of fields in the agricultural region under study, as well as for crops and cropping systems that are also typical of many other agricultural regions, have significant ramifications because they provide opportunities

to apply these classifiers to other regions around the world, such as the Mediterranean Basin, even though further research is required to assess their robustness.

Review on [6] CoReJava(Constraint Optimization Regression in Java):

In a paper [6] the proposed and realized language CoReJava (Constraint Optimization Regression in Java) adds regression analysis, or the ability to do parameter estimation for a function, to the Java programming language. In CoReJava, functional forms for regression analysis are expressed as first-class citizens, i.e., as Java programs, where some parameters are not known a priori but must be discovered from training sets given as input. Calibration of computational processes' parameters —which are referred to as OO programs—is one of CoReJava's typical uses. The CoReJava compiler (1) analyzes the parameterized Java program's structural representation of a functional form in order to implement regression learning. (2)It then automatically generates a constraint optimization problem, where the objective function to be minimized is the sum of squared errors relative to the training set, (3) and solves it using an external nonlinear optimization solver. CoReJava then runs as a regular Java application with the discovered optimal values in place of the initially unknown parameters. A straightforward supply chain example is used to explicitly define and illustrate ReJava syntax and semantics. To the best of our knowledge, this is the first study that introduces regression learning as a first-class citizen into a programming environment. They think that a range of applications, where complicated computational processes have unknown parameters that can be discovered through learning sets, can employ CoReJava. Many unanswered research questions exist. They are: (1) finding the best optimization solvers and classifying CoReJava programs into families; (2) extending Java to support other learning types, such as classification, i.e. when the functional forms return binary or finite domain values; and (3) creating specialized hybrid optimization algorithms appropriate for learning that derives from simulation processes referred to as OO programs.

Review on [7] Taxonomy of machine learning algorithms in software fault predication using object oriented metrics:

The authors of paper [7] states that the fascinating area of research in the world of software testing is the prediction of a software component's Fau proneness. Analysis of The fascinating area of research in the world of software testing is fault proneness of a software component. Through defect prediction and complexity measures, software coupling is crucial in determining the program quality. The object-oriented metrics have been employed by a number of fault prediction algorithms to forecast and localize faults. Many of these measures directly affect how well software works. Furthermore, if the fault propensity of a component is known in advance, testing time and effort can be greatly reduced. To estimate fault proneness, one might utilize metrics of object-oriented properties like inheritance, polymorphism, and encapsulation, among others. The application of object-oriented metrics in the prediction of software faults has been the subject of extensive research. In this work, taxonomy is presented.

Review on [8] Object-oriented curriculum generation for reinforcement learning:

The author of [8] is one of the key current ideas in the machine learning community is to speed up the learning process for Reinforcement Learning (RL) tasks. Although Curriculum Learning in RL is a new and promising technique, the earlier efforts required manual source task base development and properly gathered domain knowledge in order to function. Reinforcement Learning (RL)

agents take a very long time to autonomously learn a complex job. By breaking a difficult activity down into multiple manageable subtasks and arranging them in a Curriculum 20t guides Transfer Learning (TL) ways to reuse knowledge in an efficient order, one can learn more quickly. According to their experiments, their solution outperforms the state-of-the-art technique in two distinct areas when employing manually provided and generated subtasks.

Review on [9] Mitigating the compiler optimization phase-ordering problem using machine learning:

In the statements in paper [9] the study has demonstrated that Jikes RVM JIT compiler performance can be significantly enhanced by method-specific optimization orderings. Additionally, it has shown that a neural network may be automatically generated using the approach of neuro-evolution and a well-designed optimization ordering. They demonstrate improvements in total execution time of up to 20%. In order to comprehend the behavior of other environments on our system, they would like to integrate comparable phase-ordering strategies in a static compiler for future work. The method has no characteristics that make it unique to dynamic compilation. In order to enhance our forecasts, we would also like to include profile information in the feature set.

Review on [10] Pre-Implementation Method Name Prediction for Object-Oriented Programming:

The author of [10] suggesting method names presuppose that the method has already been used. However, in reality, developers require assistance when coming up with names for the methods that will probably be used once a class is developed. We refer to this work as PI-MNP (Pre-Implementation Method Name Prediction) in the literature. We also suggest Mario to handle the PI-MNP work. In order to forecast the names of the methods for the target classes, Mario uses the class names as inputs and draws on knowledge from classes with names that are semantically close. The results of the experiments show that Mario is effective at guessing method names and provides performance that is comparable to or better than existing methods that make use of implementation information. Their upcoming work will concentrate on how to use the nearby courses to improve the software development process even further. For instance, they see a situation in which they can suggest developers use code from nearby classes to improve the implementations of the class they are creating.

5. Conclusions:

In conclusion, these research publications make contributions to a number of areas related to software development, object-oriented programming, and machine learning. They demonstrate the potential of machine learning approaches in a variety of applications, such as crop classification, object detection, and program correction. The significance of well-designed tools and procedures for improving software development and computing processes is further shown by these research. Overall, these publications show how machine learning and related approaches are becoming more and more important in contemporary research and application sectors.

References:

[1] Lang, M., Binder, M., Richter, J., Schratz, P., Pfisterer, F.. Coors, S., & Bischl, B. (2019). mir3: A modem object-oriented machine learning framework in R. Journal of Open Source Software, 4(44), 1903. 16

[2] Nurminen, J. K., Rainio, K., Numminen, J. P., Syrjänen, T., Paganus, N., & Honkoila, K. (2020). Object detection in design diagrams with machine learning. In Progress in Computer Recognition Systems 11 (pp. 27-36). Springer International Publishing. 19

[3] Sonnenburg, S., Rätsch, G., Henschel, S., Widmer, C., Behr, J., Zien, A.,. & Franc, V. (2010). The SHOGUN machine learning toolbox. The Journal of Machine Learning Research, 11, 1799-1802. 15

[4] Saha, R. K., Lyu, Y., Yoshida, H., & Prasad, M. R. (2017, October). Elixir: Effective object-oriented program repair. In 2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE) (pp. 648-659). IEEE.

[5] Peña, J. M., Gutiérrez, P. A., Hervás-Martinez, C., Six, J., Plant, R. E., & López-Granados, F. (2014). Object-based image classification of summer crops with machine learning methods. Remote sensing, 6(6), 5019-5041. 22

[6] Brodsky, A., Luo, J., & Nash, H. (2008, December). CoReJava: Learning functions expressed as object-oriented programs. In 2008 Seventh International Conference on Machine Learning and Applications (pp. 368-375). IEEE. 21

[7] Singh, A., Bhatia, R., & Singhrova, A. (2018). Taxonomy of machine learning algorithms in software fault prediction using object oriented metrics. Procedia computer science, 132, 993-1001

[8] Silva, F. L. D., & Costa, A. H. R. (2018, July). Object-oriented curriculum generation for reinforcement learning. In Proceedings of the 17th international conference on autonomous agents and multiagent systems (pp. 1026-1034).

[9] Kulkami, S., & Cavazos, J. (2012, October). Mitigating the compiler optimization phase-ordering problem using machine learning. In Proceedings of the ACM international conference on Object oriented programming systems languages and applications (pp. 147-162).

[10] Wang, S., Wen, 23 Lin, B., Liu, Y., Bissyandé, T. F., & Mao, X. (2023). Pre-Implementation Method Name Prediction for Object-Oriented Programming. ACM Transactions on Software Engineering and Methodology.

© 2020 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).