**RUN-TIME EFFICIENCY ASSESSMENT FOR THE SELECTION OF OPTIMIZED DATA STRUCTURES FOR JAVA PROGRAMS**

**R.N.G.J.H. Nawarathna\*, P.M.P.C. Gunathilake**

*Department of Computer Science and Statistics, University of Peradeniya, Peradeniya,*

*Sri Lanka.*

*\*s16417@sci.pdn.ac.lk*

Data structures help programmers a lot in terms of organizing, storing, and handling data more efficiently. The set of built-in optimized data structures has many advantages in Java because they can be used depending on the programmer's needs. These include data structures like Arrays, Lists, Maps, etc. This research work intends to make those programmers' needs accessible more efficiently by providing guidance or assistance by comparing and analyzing the run-time behaviour of different Java data structures with a machine learning approach. We have taken the List, and the Map interface from the Java Collection interface into account to carry out the research. By having the selected data structures benchmarked to get the run-time behaviour in the sense of how a data structure in a particular program manages to handle the memory and the run-time, we would be able to get an idea of how that data structure would perform in different scenarios. Java data structures have properties like growing policies; the data structure grows as it gets filled with elements and dynamically increases and decreases the storage to handle the input data efficiently. With the collected data of the run-time behaviour of those selected data structures, we can extract the information of so-called data-structure properties. We could differentiate those data structures in terms of their run-time behaviour by analyzing the data. Furthermore, have an assisting report to suggest better data structures to use, particularly in programs like Java enterprise applications, to improve performance and ultimately give a better experience to the end-user. After analyzing the collected data through the benchmarking process, one finding was that for the behaviour of the memory usage of each data structure, there's no need for any machine learning algorithm to classify the data structures. Because, just by analyzing data, we could figure out which data structures are better in each scenario. But, for the runtime behaviour, it's different in each iteration of the amount of data processed through the data structures. Classification algorithm from the library XGBoost helped to create machine learning models to get the suggestions for each situation by analyzing the amount of data and the runtime.

**Keywords**: Java, Data Structures, Runtime Behaviour, Efficiency Assessment