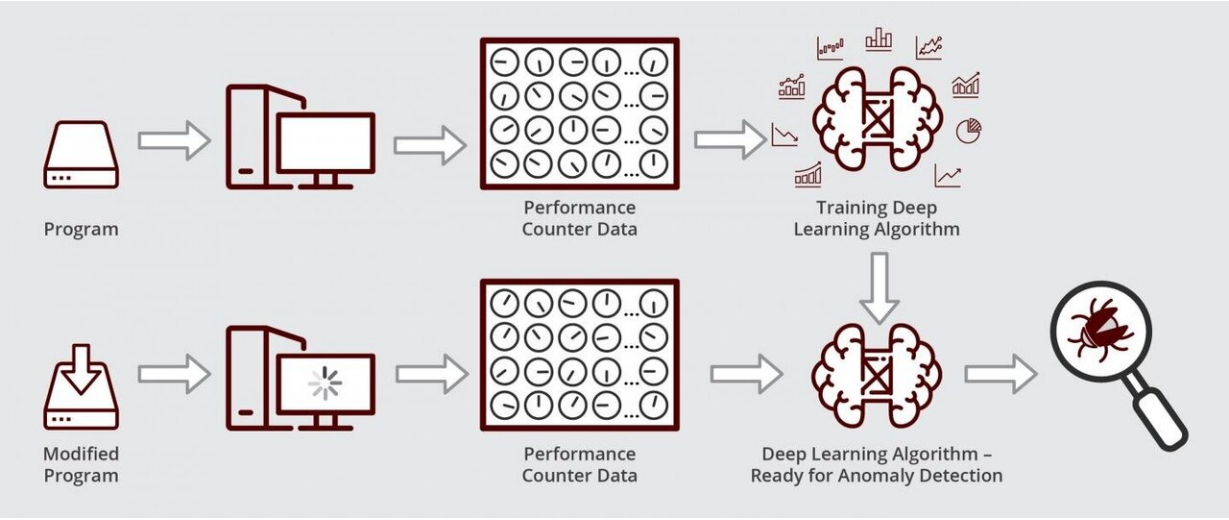
# Computer scientists design a tool to identify the source of errors caused by software updates

February 11 2020, by Vandana Suresh and Stephanie Jones




Schematic illustrating how Muzahid's deep learning algorithm works. The algorithm is ready for anomaly detection after it is first trained on performance counter data from a bug-free version of a program. Credit: Texas A&M Engineering

We've all shared the frustration—software updates that are intended to make our applications run faster inadvertently end up doing just the opposite. These bugs, dubbed in the computer science field as performance regressions, are time-consuming to fix since locating software errors normally requires substantial human intervention.

To overcome this obstacle, researchers at Texas A&M University, in collaboration with computer scientists at Intel Labs, have now developed a complete automated way of identifying the source of errors caused by software updates. Their algorithm, based on a specialized form of machine learning called deep learning, is not only turnkey, but also quick, finding performance bugs in a matter of a few hours instead of days.

"Updating software can sometimes turn on you when errors creep in and cause slowdowns. This problem is even more exaggerated for companies that use large-scale software systems that are continuously evolving," said Dr. Abdullah Muzahid, assistant professor in the Department of Computer Science and Engineering. "We have designed a convenient tool for diagnosing performance regressions that is compatible with a whole range of software and programming languages, expanding its usefulness tremendously."

The researchers described their findings in the 32nd edition of Advances in Neural Information Processing Systems from the proceedings of the Neural Information Processing Systems conference in December.

To pinpoint the source of errors within software, debuggers often check the status of performance counters within the central processing unit. These counters are lines of code that monitor how the program is being executed on the computer's hardware in the memory, for example. So, when the software runs, counters keep track of the number of times it accesses certain memory locations, the time it stays there and when it exits, among other things. Hence, when the software's behavior goes awry, counters are again used for diagnostics.

"Performance counters give an idea of the execution health of the program," said Muzahid. "So, if some program is not running as it is supposed to, these counters will usually have the telltale sign of

anomalous behavior."

However, newer desktops and servers have hundreds of performance counters, making it virtually impossible to keep track of all of their statuses manually and then look for aberrant patterns that are indicative of a performance error. That is where Muzahid's machine learning comes in.

By using deep learning, the researchers were able to monitor data coming from a large number of the counters simultaneously by reducing the size of the data, which is similar to compressing a high-resolution image to a fraction of its original size by changing its format. In the lower dimensional data, their algorithm could then look for patterns that deviate from normal.

When their algorithm was ready, the researchers tested if it could find and diagnose a performance bug in a commercially available data management software used by companies to keep track of their numbers and figures. First, they trained their algorithm to recognize normal counter data by running an older, glitch-free version of the data management software. Next, they ran their algorithm on an updated version of the software with the performance regression. They found that their algorithm located and diagnosed the bug within a few hours. Muzahid said this type of analysis could take a considerable amount of time if done manually.

In addition to diagnosing performance regressions in software, Muzahid noted that their deep learning algorithm has potential uses in other areas of research as well, such as developing the technology needed for autonomous driving.

"The basic idea is once again the same, that is being able to detect an anomalous pattern," said Muzahid. "Self-driving cars must be able to

detect whether a car or a human is in front of it and then act accordingly. So, it's again a form of anomaly detection and the good news is that is what our algorithm is already designed to do."

Other contributors to the research include Dr. Mejbah Alam, Dr. Justin Gottschlich, Dr. Nesime Tatbul, Dr. Javier Turek and Dr. Timothy Mattson from Intel Labs.

Provided by Texas A&M University

Citation: Computer scientists design a tool to identify the source of errors caused by software updates (2020, February 11) retrieved 20 April 2024 from https://techxplore.com/news/2020-02-scientists-tool-source-errors-software.html