Dr hab. Bartosz Świderski, Warszawa, 17.08.2022

Instytut Informatyki Technicznej

Katedra Sztucznej Inteligencji

Szkoła Główna Gospodarstwa Wiejskiego w Warszawie,

Ul. Nowoursynowska 159, 02-776 Warszawa

[bartosz\_swiderski@sggw.edu.pl](mailto:bartosz_swiderski@sggw.edu.pl)

Recenzja rozprawy doktorskiej mgra inż. Kazimierza Krosmana pt. „Ewaluacja systemów wbudowanych poprzez monitorowanie programowe”

promotor rozprawy:

prof. dr hab. inż. Janusz Sosnowski, Wydział Elektroniki i Technik Informacyjnych, Politechnika Warszawska

promotor pomocniczy:

dr inż. Piotr Gawkowski, Wydział Elektroniki i Technik Informacyjnych, Politechnika Warszawska

zlecona pismem z dnia 6 lipca 2022 r., PSP/zlecenie: 504/04291/1030/44000000 przez Przewodniczącego Rady Naukowej Dyscypliny Informatyka Techniczna i Telekomunikacja PW, dr hab. Jarosłąwa Arabasa, prof. PW

Wstęp

Tematyka recenzowanej rozprawy doktorskiej koncentruje się wokół automatyzacji monitorowania systemów komputerowych ze szczególnym uwzględnienie systemów wbudowanych. Jako wykorzystywane źródła informacji Autor rozważa:

1. ślad instrukcji (przebieg wykonania instrukcji jako zestaw adresów instrukcji)
2. zawartość logów programowych
3. inne sygnały towarzyszące wykonywaniu danego programu ujęte w szereg czasowy (np. zużycie energii)

Autorsko opracowane algorytmy dotyczą:

1. wykrywania anomalii
2. analizy szeregów czasowych poprzez model generujący graf przejść pomiędzy zidentyfikowanymi stanami wraz z łączącymi je relacjami
3. moduł korelacji logów tekstowych z szeregiem czasowym umożliwiającym poszerzoną analizę badanego programu.

Praca zawiera tezę: „Analiza i monitorowanie pracy systemów wbudowanych wymaga stworzenia odpowiednich modeli dekompozycji, agregacji i korelacji danych z procesów monitorowania” Przy czym w pracy dodatkowo wyróżniono następujące cele badawcze:

1. identyfikacja problemów testowania i weryfikacji systemów wbudowanych oraz ich powiązanie z różnymi perspektywami obserwacji (monitorowania).
2. analiza specyfiki monitorowanych danych oraz opracowanie obiektowych modeli ich reprezentacji.
3. opracowanie algorytmów eksploracji oraz korelacji danych bazujących na stworzonych modelach.
4. opracowanie narzędzi wspomagających tworzenie modeli i ich analizę.
5. weryfikacja opracowanej metodyki badań na przykładzie zrealizowanych systemów wbudowanych.

Przeprowadzone i opisane przez Autora badania są spójne z przyjętymi celami, potencjalnie mogą prowadzić do rozwoju użytecznych narzędzi automatycznego monitoringu oprogramowania (dotyczy to również systemów wbudowanych).

Charakterystyka zawartości pracy:

Praca składa się z sześciu rozdziałów oraz bibliografii wraz z załącznikami (opisami zastosowanych systemów agregacji i analizy śladu oraz detekcji stanów i korelacji logów). Całość liczy 169 stron, bibliografia zawiera 95 pozycji generalnie odpowiednio dobranych i poprawnie przytaczanych (trafnie przeprowadzona analiza źródeł).

Rozdział pierwszy zawiera wprowadzenie, przegląd literatury, omówienie struktury pracy. W rozdziale tym Autor także określa tezę oraz cele pracy.

Drugi dotyczy monitorowania przebiegu wykonywania programu poprzez analizę śladu instrukcji. W rozdziale tym Autor prezentuje autorski algorytm detekcji anomalii.

W rozdziale trzecim przedstawiono interesujące podejście do analizy szeregu czasowego towarzyszącego wykonywaniu programu (np. charakterystyk poboru energii). Opisywane podejście zdolne jest identyfikować stany systemu a także możliwe przejście pomiędzy nimi. Finalnym efektem jest graf opisujący zachowanie monitorowanego programu wraz ze zidentyfikowanymi stanami oraz możliwymi przejściami pomiędzy nimi.

Czwarty rozdział prezentuje algorytm „korelacji logów” czyli podejście umożliwiające przyporządkowanie logów do poszczególnych stanów, finalnie ułatwiając wychwycenie precyzyjniejszych opisów zdarzań.

Ocena rozprawy doktorskiej z uwzględnieniem wytycznych „Informacje i zalecenia dla recenzentów rozpraw doktorskich Politechnika Warszawska”:

Recenzowana praca dotyczy zagadnienia automatycznego wspomagania monitoringu i diagnostyki monitorowania programowego systemów. Autor w pracy przedstawił trzy grupy (moduły) rozwiązań.

Pierwsza grupa odnosi się do identyfikacji anomalii. Warto to podkreślić, że zagadnienie to samo w sobie w tym przypadku jest dość trudne do uniwersalnego scharakteryzowania. Niemal dowolny, pozornie anomalny przebieg wykonywania instrukcji, może być w gruncie rzeczy zachowaniem zgodnym z celem programu. Pewnym przykładem może tu być ciąg symulacji, gdzie po przekroczeniu zadanego kwantylu zmiennej losowej wchodzimy w inny zestaw instrukcji. Z drugiej strony jednak intuicyjnie zrozumiała jest pragmatyczna potrzeba i celowość wykrywania anomalnego zachowania programu. Samo zaś wykrywanie anomalii jako problem posiada szerokie spektrum wypracowanych już rozwiązań i dość gruntowne podwaliny teoretyczne. Autor prezentuje w pracy, oryginalne rozwiązanie polegające na wykorzystaniu algorytmu DBSACN oraz autorsko dobranej metryki odległości (opartą o różnicę symetryczne zbiorów). W tym miejscu naturalnym wydaje się pytanie o zastosowanie „dedykowanych” metod wykrywania anomalii (niekoniecznie opartych o wspomnianą metrykę odległości) takich jak np.: One-Class SVM, Local Outlier Factor, Isolation Forest, czy wreszcie metody wykorzystujące głębokie uczenie (np. wykorzystujące residua w autoencoderze). Podobnie sama metryka odległości w gruncie rzeczy działająca na histogramach przedstawia interesującą ideę porównania dwóch rozkładów. Jednak warto zaznaczyć, że istnieje tu także wiele metod badających odległość czy dywergencje pomiędzy rozkładami (przykładami mogą być: statystyka Kołmogorowa-Smirnowa, szereg miar entropijnych, czy wreszcie odległość Wassersteina - oparta o teorię transportu miary, miara ciesząca się ostatnio dość dużym zainteresowaniem stosowana np. przy niektórych sieciach głębokich w odniesieniu do wielowymiarowych rozkładów). Niewątpliwie porównanie wypracowanych przez Autora metod z niektórymi znanymi już rozwiązaniami uatrakcyjniłoby pracę.

Druga grupa wypracowanych rozwiązań dotyczy analizy szeregu czasowego towarzyszącemu wykonywaniu danego programu. Przykładam może tu być np. szereg czasowy charakteryzujący pobór energii w czasie. Opracowany algorytm umożliwia identyfikację odpowiednich (podyktowanych metryką) stanów w jakich znajduje się program wraz z digrafem przejść pomiędzy nimi. Szczególną rolę w algorytmie odgrywa stan bazowy (jako stan powracający), na podstawie którego identyfikowane są poszczególne cykle. W wyniku pracy algorytmu otrzymujemy pewnego rodzaju graf wiedzy opisujący behawioralny profil działania badanego programu. Może on posłużyć zarówno do badania zależności między zidentyfikowanymi stanami, wykrywania anomalii jak również do pewnych optymalizacji (np. zużycia energii). Wypracowanym rozwiązaniem nie jest zatem konkretny model szeregu czasowego (jakie występują np. na gruncie ekonometrii) a raczej hierarchiczny obiektowy model „zjawiska” charakteryzujący zmiany stanów w czasie z uwzględnieniem identyfikacji klas poszczególnych cykli. Jednym z kluczowych komponentów jest tutaj funkcja określająca podobieństwo pomiędzy segmentami/grupami. Autor podaje przykład takiej funkcji, pozostawiając jednak pewne pole również do rozbudowy czytelnikowi. I chociaż przytoczona funkcje może nie optymalna (np. porównywanie wartości skrajnych, porównanie całek oznaczonych pod sygnałami potencjalnie niewycentrowanymi, etc.) jednak pokazuje pewną ideę działającą w określonych warunkach. Ważną podkreślenia jest również możliwość zastosowania technik grafowych dla tak opisanego zjawiska (przebiegu wykonywania programu), ze szczególnym uwzględnieniem narzędzi używanych w sieciach społecznościowych (w tym także wykrywania anomalii).

Trzeci moduł rozwiązań opisywanych w pracy dotyczy korelacji tekstowych logów ze zidentyfikowanym wcześniej (moduł 2) zdarzeniami. Moduł ten szczególnie przydatny jest w sytuacji niezsynchronizowania obu zegarów podsystemów (listy obiektów zdarzeń i systemu logowania). Idea tego rozwiązania polega na powiązaniu znaczników czasowych grup (sekwencji logów) logów tekstowych (zdarzeń) ze znacznikami czasowymi zidentyfikowanych wcześniej cykli. Podczas generacji poszczególnych grup logów Autor używa ważonej miary podobieństwa dla reprezentacji *bag of words* poszczególnych logów. Finalnie algorytm zwraca przesunięcie pomiędzy osiami czasowymi zdarzeń i logów oraz sekwencje przyporządkowanych do siebie zdarzeń z logami.

Warto również podkreślić „nieinwazyjność” proponowanych metod co jest niewątpliwą zaletą na tle szerzej stosowanych/publikowanych metod (np. w stosunku do metod wykorzystujących wstrzykiwanie znaczników przy synchronizacji logów).

Jako oryginalny dorobek Autora rozprawy oraz główny wkład w rozwój dyscypliny naukowej wymienić można:

1. Opracowanie algorytmu detekcji anomalii na podstawie analizy śladu instrukcji wraz z implementacją algorytmu w systemie *TraceAnalyzer* i weryfikacją na przykładzie wybranego systemu wbudowanego (lokalizacja samochodów ciężarowych).

2. Opracowanie hierarchiczno-obiektowego algorytmu analizy szeregu czasowego wraz z reprezentacją w postaci grafu skierowanego dla zidentyfikowanych stanów istotnych przejść pomiędzy nimi. Implementacja algorytmy w systemie *StateEventAnalyzer* i jego weryfikacja na przykładzie holtera EKG.

3. Opracowanie algorytmu korelacji logów tekstowych z szeregiem czasowym i jego implementacja w systemie *StateEventAnalyzer* (rozszerzenie) i jego weryfikacja na przykładzie holtera EKG.

Dodatkowo bibliografia przytacza sześć publikacji Autora, z których cztery [1-4] są ściśle powiązane z rozważaniami zawartymi w pracy doktorskiej a dwie wcześniejsze [5,6] dotyczą innych perspektyw monitowania systemów zorientowanych na aspekty wydajnościowe i niezawodnościowe.

1. K. Krosman and J. Sosnowski, "Correlating Time Series Signals and Event Logs in Embedded Systems," *Sensors,* vol. 21, p. 7128, 1 2021.

2. K. Krosman, J. Sosnowski and P. Gawkowski, "Object oriented time series exploration: Applied to power consumption analysis of embedded systems," *Expert Systems with Applications,* vol. 184, p. 115531, 12 2021.

3. K. Krosman, „Instruction trace analysis and enhanced debugging in embedded systems,” w *Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2018*, 2018.

4. K. Krosman i J. Sosnowski, „ESD problems in embedded systems,” w *Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2019*, 2019.

5. K. Krosman i J. Sosnowski, „Fault tolerance techniques for embedded telemetry system: case study,” w *Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2016*, 2016.

6. K. Krosman i J. Sosnowski, „Exploring disk performance benchmarks,” w *Photonics Applications in Astronomy, Communications, Industry, and High Energy Physics Experiments 2017*, 2017.

Dwie z powyższych prac posiadają Impact Factor oraz wysoką punktację ministerialną (Expert Systems with Applications – IF = 5.43, 140 pkt, Sensors – IF = 3.64, 100 pkt ). Ponadto warte jest również podkreślenia ponad 12-letnie doświadczenie zawodowe Autora skupiające się głównie wokół otoczenia opisywanego obszaru. To pozwoliło mu zweryfikować opracowaną metodykę monitorowania na danych z rzeczywistych systemów wbudowanych (holtery EKG), opracowane narzędzia wykorzystał do optymalizacji tych urządzeń w swojej pracy zawodowej.

Co do samej pracy - niestety jest ona napisana w nieco chaotyczny sposób. Niekiedy dają znać o sobie występujące literówki (przykładowo str. 5 ‘chraktezujące’, str 14 ‘wpisującego się’ – brak „w”, etc.). Ponadto niekiedy struktura pracy wydaje się nie do końca optymalna np. rozdział opisujący pseudokod powinien raczej znaleźć się przed pierwszym pseudokodem (czyli przed rozdziałem 2).

Zdarzają się również niedoskonałości w opisach merytorycznych np. str 27 – na schemacie blokowym (po START) powinno być „Agregacja N” (duże N). Zaś w samym schemacie blokowym: linia 5 – parametr S powinien być opisany (segment). Linia 7 zamiast “foreach (segment in streams)” powinno być: “foreach (segment in segments) do”. Dalej str 29 – widnieje odesłanie do nieistniejącego wzoru 2.14 (powinno być 2.7). Wreszcie sam dowód nierówności trójkąta (dalej strona 29) mógłby być przeprowadzony w sposób bardziej elegancji (nie zaczynając od tezy). Podobnie w algorytmie *add\_to\_similar\_state* na str. 52 warto by rozważyć jawnie przypadek podobieństwa danej grupy do kilku segmentów. Innym przykładem, gdzie pożądany wydałby się szerszy opis jest rysunek 10, str. 65 (scalanie stanów wobec braku scalania na poprzednim rysunku 9). Dużą część wątpliwości recenzent wyjaśnił bezpośrednio komunikując się z Autorem.

Na koniec uwaga ogólna. W pracy przetestowano skuteczność oryginalnie wypracowanych narzędzi na przykładzie wybranych problemów. Warto by było jednak pokusić się o przeprowadzenie szerszych eksperymentów. Najlepiej odwołujących się do opublikowanych benchmarków lub do badań symulacyjnych (ze znaną odpowiedzią / Ground Truth). Wówczas łatwiej jest zawrzeć syntetyczne podsumowania prezentowanych algorytmów (przykładowo dla wykrywania anomalii miary oparte na macierzy konfuzji, AUC, etc.), dając przy tym również odniesienia dla kolejnych badaczy. Możliwe byłoby przy tym przeprowadzenie eksperymentów w trybie „testowania” tzn. po wcześniejszych ustaleniu hiperparametrów (na innych danych „uczących”) dla danej klasy zagadnienia.

Powyższe krytyczne uwagi nie wpływają jednak znacząco na ocenę pracy. Wyraźna część z nich dotyczy warstwy redakcyjnej pracy. Odnosząc się zaś do poczynionych wcześniej uwag dotyczących możliwości przebadania alternatywnych metod, należy nadmienić, że uwagi te mają w dużej mierze charakter formalny. Nie umniejszają one istotnie osiągnięciom zawartym w pracy. Wskazując w pewnym sensie na samodzielne podejście Autora „od idei” a nie „od narzędzia” do „celu”. Autor rozwiązał postawione zagadnienia/cele, używając do tego generalnie odpowiednich metod, opierając się na pragmatycznych założeniach. Przedstawione rezultaty przemawiają za potencjalną przydatnością opisywanych metod w zagadnieniach automatycznego monitoringu / diagnostyki programów ze szczególnym uwzględnieniem systemów wbudowanych.

Finalnie, konkludując, stwierdzam, że recenzowana praca spełnia wymagania stawiane przez „art. 179 ust.2 Ustawy z dnia 3 lipca 2018 roku Przepisy wprowadzające ustawę – Prawo o szkolnictwie wyższym i nauce (Dz. U. z dnia 30 sierpnia 2018 r., poz. 1669) w związku art. 13 ust. 1 Ustawy z dnia 14 marca 2003 roku o stopniach naukowych i tytule naukowym oraz o stopniach i tytule w zakresie sztuki (Dz. U. z 2003 r. nr 65, poz. 595 z późn. zm.)  oraz Rozporządzenie Ministra Nauki i Szkolnictwa Wyższego z dnia 19 stycznia 2018 roku w sprawie szczegółowego trybu i warunków przeprowadzania czynności w przewodach doktorskich, postępowaniu habilitacyjnym oraz w postępowaniu o nadanie tytułu profesora” na stopień doktora i wnoszę i jej dopuszczenie do publicznej obrony. Ponadto z uwagi na oryginalność pracy oraz dorobek Autora sugeruję rozważyć wyróżnienie rozprawy.

Obraz zawierający tekst

Opis wygenerowany automatycznieZ poważaniem