
PREDIKCIA ČASOVÝCH RADOV POMOCOU METÓD STROJOVÉHO UČENIA V OBLASTI POČÍTAČOVEJ BEZPEČNOSTI

Alex Gajdoš¹

Školiteľ^a: RNDr. Richard Staňa

¹Ústav informatiky, Prírodovedecká fakulta UPJŠ, Jesenná 5, 040 01 Košice

Práca sa zameriava na predikciu časových radov v oblasti informačnej bezpečnosti, s cieľom poskytnúť sieťovým administrátorom lepší prehľad o aktuálnom stave siete a pomôcť im predvídať, ako sa tento stav bude meniť v budúcnosti. Sústreďuje sa na použitie metód strojového učenia, ako sú SARIMA, LightGBM, XGBoost, SVR a Prophet a porovnáva výsledky týchto metód s výsledkami prác, ktoré použili iné štatistické metódy a neurónové siete. Úloha predikcie je v tejto oblasti veľmi náročná z viacerých dôvodov, ako je nedostatok kvalitných dát a ich zložitosť v interpretácii. Cieľom práce je nájsť dobrý prístup v predikcii časových radov v oblasti informačnej bezpečnosti a tým pomôcť komunitě v ďalšom výskume v tejto oblasti. Výsledky by mohli pomôcť sieťovým administrátorom a výskumníkom zlepšiť predikciu a poskytnúť lepší prehľad o aktuálnom stave a budúcich trendoch v oblasti sieťového bezpečnostného situačného povedomia.

MONITORING OF BODY POSTURE DURING COMPUTER USE

Bc. Jana Kovaličová

Školiteľ: RNDr. Juraj Šebej, PhD.

Ústav informatiky, Prírodovedecká fakulta UPJŠ, Jesenná 5, 04001 Košice

We live in a time where more and more jobs require either full or partial computer work. Many people are starting to experience various medical conditions, which are directly related to the sedentary type of work. In recent years, quite a few sitting posture monitoring systems have been developed to prevent back pain related to the bad sitting posture. In this work, we focus on proper sitting and posture when working with a computer and we review some available systems for posture monitoring. We have a look at different technologies and methods used, to help us design our own posture monitoring system. We then describe a process of designing a network of IoT sensors that is attached to the back of the chair and used to monitor a posture of a computer user. The data we collect are then processed and annotated so that it can be used for the artificial intelligence models. We use unsupervised and supervised machine learning algorithms and compare the results of those algorithms.

Literatúra:

1. M. Taieb-Maimon, J. Cwikel, B. Shapira, and I. Orenstein, “The effectiveness of a training method using self-modeling webcam photos for reducing musculoskeletal risk among office workers using computers,” *Appl. Ergon.*, vol. 43, no. 2, pp. 376–385, Mar. 2012, doi: 10.1016/J.APERGO.2011.05.015.
2. A. Kulikajevas, R. Maskeliunas, and R. Damaševičius, “Detection of sitting posture using hierarchical image composition and deep learning,” *PeerJ Comput. Sci.*, vol. 7, pp. 1–20, Mar. 2021, doi: 10.7717/PEERJ-CS.442/TABLE-4.
3. J. Roh, H.-J. Park, K. J. Lee, J. Hyeong, S. Kim, and B. Lee, “Sitting Posture Monitoring System Based on a Low-Cost Load Cell Using Machine Learning,” doi: 10.3390/s18010208.
4. F. Luna-Perejón, J. M. Montes-Sánchez, L. Durán-López, A. Vazquez-Baeza, I. Beasley-Bohórquez, and J. L. Sevillano-Ramos, “IoT device for sitting posture classification using artificial neural networks,” *Electron.*, vol. 10, no. 15, pp. 1–15, 2021, doi: 10.3390/electronics10151825.
5. S. Matuska, M. Paralic, and R. Hudec, “A Smart System for Sitting Posture Detection Based on Force Sensors and Mobile Application,” *Mob. Inf. Syst.*, vol. 2020, 2020, doi: 10.1155/2020/6625797.
6. B. Mahesh, “Machine Learning Algorithms-A Review Machine Learning Algorithms-A Review View project Self Flowing Generator View project Batta Mahesh Independent Researcher Machine Learning Algorithms-A Review,” *Int. J. Sci. Res.*, 2018, doi: 10.21275/ART20203995.

METÓDY SAMOKONTROLOVANÉHO UČENIA PRE ANALÝZU MOLEKULÁRNYCH VLASTNOSTÍ

Bc. Samuel Baran

Školiteľ: RNDr. Juraj Šebej, PhD.

Ústav informatiky, Prírodovedecká fakulta, UPJŠ, Jesenná 5, 040 01 Košice

Vďaka zvýšenej popularite metód strojového učenia aplikovaných na úlohy predikcie molekulárnych vlastností zaznamenala chemoinformatika v posledných desaťročiach výrazný posun. Zároveň došlo k rozšíreniu datasetov a viacero nových modelov dosiahlo v úlohách predikcie molekulárnych vlastností sľubné výsledky.

Strojové učenie aplikované v oblasti spracovania molekúl však aj naďalej čelí viacerým výzvam. Molekulárne datasety sú oveľa menšie ako datasety z iných oblastí, keďže získavanie takýchto dát je vzhľadom na potrebu špecializovaných zariadení a dohľadu odborníkov náročné, no súčasne je priestor chemických zlúčenín enormne veľký. Výrazný nepomer množstva dostupných dát a celkovej veľkosti priestoru príkladov znemožňuje efektívne využitie metód učenia s učiteľom, pretože modely tréňované na malých datasetoch zle generalizujú a sú náchylné na preučenie. Preto sa vývoj v posledných rokoch zamerával aj na oblasť samokontrolovaného učenia, ktorého prednosťou je primárne využitie neanotovaných dát na vytvorenie skrytých reprezentácií vstupov, ktoré poskytujú lepšiu východiskovú pozíciu pre modely učenia s učiteľom. Príkladmi takýchto metód v chemickej doméne sú MolCLR a ChemBERTa.

V tejto práci predstavíme metódu samokontrolovaného učenia, ktorá vznikla úpravou nedávno publikovanej metódy DINO pôvodne navrhutej na spracovanie obrazu tak, aby ju bolo možné aplikovať na grafové reprezentácie molekúl. Ako kostrovú architektúru sme namiesto visual transformerov použili grafovú konvolučnú neurónovú sieť a navrhli sme transformácie grafových reprezentácií molekúl tak, aby zachovávali pôvodnú ideu algoritmu DINO.

Naše experimenty ukázali, že aj napriek limitovanej predtréningovej fáze nami natrénovaný model prekonal existujúce modely učenia s učiteľom na väčšine datasetov a tiež zaznamenal oproti kontrolovanému tréningu grafovej konvolučnej siete lepšie výsledky na piatich zo šiestich datasetov. Zároveň sa nám podarilo prekonať metódu ChemBERTa na všetkých dostupných datasetoch a v porovnaní s ostatnými samokontrolovanými metódami naša predtréningová stratégia dosiahla porovnateľné výsledky.

Literatúra:

1. WU, Zhenqin, et al. MoleculeNet: a benchmark for molecular machine learning. *Chemical science*, 2018, 9.2: 513-530.
2. CARON, Mathilde, et al. Emerging properties in self-supervised vision transformers. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021. p. 9650-9660.

GRAFY ÚTOKOV V KYBERNETICKEJ BEZPEČNOSTI

Bc. Vladimír Homola

Školiteľ: doc. RNDr. JUDr. Pavol Sokol, PhD.

Ústav informatiky, Prírodovedecká fakulta UPJŠ, Jesenná 5, 040 01 Košice

V práci sa venujeme spracovaniu bezpečnostných údajov, analýze kybernetických útokov, technikám modelovania útokov, konkrétne generovaniu ich grafov. Bližšie sa venujeme nástroju pre generovanie grafov útokov – MulVAL. Tento nástroj je už pre dnešné potreby zastaralý a navyše nie je jednoduché ho na dnešných zariadeniach spustiť. Po preštudovaní aktuálneho stavu poznania pre tento nástroj sme sa rozhodli, že pôvodné jadro nástroja, ktoré používa logické programovanie, je to čo robí tento nástroj efektívnym. Preto sme sa rozhodli ho zachovať a zamerali sme sa na prerobenie vstupu a výstupu nástroja. Na vstupe je hlavnou zmenou, že zraniteľnosti sa sťahujú z externej databázy zraniteľností a dajú sa priebežne aktualizovať. Na výstupe sa graf exportuje do grafovej databázy, kde sa dá následne prezerať, poprípade na ňom vykonať ďalšia analýza.

Literatúra:

1. ZENG, Jianping, et al.: Survey of attack graph analysis methods from the perspective of data and knowledge processing. Security and Communication Networks, 2019.
2. WANG, Lingyu; JAJODIA, Sushil; SINGHAL, Anoop. Network Security Metrics. Springer, 2017.
3. KAYNAR, Kerem. A taxonomy for attack graph generation and usage in network security. Journal of Information Security and Applications, 2016, 29: 27-56.
4. OU, Xinming; GOVINDAVAJHALA, Sudhakar; APPEL, Andrew W. MulVAL: A Logic-based Network Security Analyzer. In: USENIX Security Symposium. 2005. p. 8-8.

RICEOV – SIFFOV ALGORITMUS A VYJADROVACIA SILA JEHO VÝSTUPOV

Autor¹ : Bc. František Kurimský

Autor² : Bc. Dominik Džama

Školiteľ: doc. RNDr. Ondrej Krídlo, PhD

*Adresa : Ústav informatiky, Prírodovedecká fakulta, Univerzita Pavla Jozefa Šafárika v Košiciach, Jesenná 5,
040 01 Košice*

V predloženej práci experimentálne poukazujeme na postačujúcu vyjadrovaciu silu informácie obsiahnutej v množine formálnych konceptov získaných tzv. Riceovým-Siffovým algoritmom. Spomínaný algoritmus vyhľadáva formálne koncepty v objektovo-atribútovej tabuľke pomocou Jaccardovej vzdialenosti, pričom spája vždy dva najbližšie koncepty. Počet získaných konceptov je zhruba dvojnásobok počtu objektov (resp menšieho z rozmerov vstupnej tabuľky). Spomínaný počet je rapidne menší než počet všetkých formálnych konceptov, ktorých je vo všeobecnosti exponenciálne veľa vzhľadom k rozmerom vstupnej tabuľky. Takto redukovanú množinu sme podrobili schopnosti pokrývania vstupnej tabuľky. Výsledky našej štúdie poukazujú na to, že i takáto extrémne redukovaná množina výsledkov má štatisticky rovnakú vyjadrovaciu silu ako plná množina všetkých formálnych konceptov.

DIZAJN A IMPLEMENTÁCIA KOMPILÁTORA JAZYKA HASKELL

Matej Nižník

Školiteľ: doc. RNDr. Ondrej Kridlo PhD.

Ústav informatiky, Prírodovedecká fakulta UPJŠ, Jesenná 5, 040 01 Košice

Cieľom práce je implementovať základy kompilátora štandardu Haskell2010, skúmajúc aktuálne používané techniky pri návrhu kompilátorov pre funkcionálne, nestriktne vykonávané a silne staticky typované jazyky.

Literatúra:

1. Programovací jazyk Haskell, <https://www.haskell.org/>
2. Hudak, P., Hughes, J., Jones, S. Peyton, Wadler, P.: A History of Haskell: Being Lazy With Class
3. Programovací jazyk Miranda, <https://www.cs.kent.ac.uk/people/staff/dat/miranda/>
4. Glasgow Haskell Compiler, <https://www.haskell.org/ghc/>
5. Kompilátor Duet, <https://github.com/chrisdone/duet>
6. Programovací jazyk PureScript, <https://www.purescript.org/>
7. Programovací jazyk Elm, <https://elm-lang.org/>
8. 2021 State of Haskell Survey Results, <https://taylor.fausak.me/2021/11/16/haskell-survey-results/>
9. Haskell 2010 Language Report, <https://www.haskell.org/onlinereport/haskell2010/>
10. ECMAScript 2021 language specification, <https://www.ecma-international.org/publications-and-standards/standards/ecma-262/>
11. Programovací jazyk TypeScript, <https://www.typescriptlang.org/>
12. Programovací jazyk CoffeeScript, <https://coffeescript.org/>
13. Programovací jazyk C—, <https://www.cs.tufts.edu/~nr/c--/index.html>
14. The LLVM Compiler Infrastructure, <https://www.llvm.org/>
15. Happy, The Parser Generator for Haskell, <https://www.haskell.org/happy/>
16. Alex: A lexical analyser generator for Haskell, <https://www.haskell.org/alex/>
17. Knižnica megaparsec, <https://github.com/mrkrp/megaparsec>
18. Knižnica attoparsec, <https://github.com/bgamari/attoparsec>
19. Knižnica trifecta, <https://github.com/ekmett/trifecta>
20. Knižnica flatparse, <https://github.com/AndrasKovacs/flatparse>
21. Hindley, J. Roger: The Principal Type-Scheme of an Object in Combinatory Logic. Transactions of the American Mathematical Society, 146: 29–60 (1969)
22. Clement, D.: A Simple Applicative Language: Mini-ML. LFP'86, ACM (1986)
23. Jones, S. Peyton: Type inference as constraint solving: how GHC's type inference engine actually works. Zurich keynote talk (2019)
24. Core to core pipeline, <https://gitlab.haskell.org/ghc/ghc/-/wikis/commentary/compiler/core-to-core-pipeline>
25. Jones, S. L Peyton: Implementing lazy functional languages on stock hardware: the Spineless Tagless G-machine. (1992)
26. GRIN Compiler, <https://grin-compiler.github.io/>
27. Barendregt, H.: Introduction to generalized type systems. Journal of Functional Programming, 1 (2): 125–154 (1991)

PARTIALLY NONDETERMINISTIC AUTOMATA - NONDETERMINISTIC CHOICE OF INITIAL STATES

Autor: Šimon Huraj

Školiteľ: RNDr. Juraj Šebej, PhD.

*Ústav informatiky, Prírodovedecká fakulta, Univerzita Pavla Jozefa Šafárika v Košiciach, Jesenná 5, 040 01
Košice*

We study nondeterministic automata, where the transitions are deterministic, and multiple states can be initial. This model is called a k -entry deterministic finite automata. We show the average state complexity of conversion from k -DFA to DFA. By the family, we call DFA without any initial state. Therefore, for a single n -state family, there are 2^n possible choices of initial states. In other words, one family represents 2^n k -entry deterministic finite state automata. For every family, we are interested in the number of distinct languages and state complexities of all the languages in the family. We also show that for every n , there exists a family such that all 2^n of its languages are distinct. We also provide various calculations on all families up to 5 states concerning state complexity and other properties. Computations are based on our theoretical results, which improve their complexity.

Literatúra:

1. Holzer, M., Salomaa, K. and Yu, S., On the State Complexity of k -Entry Deterministic Finite Automata, In: Journal of Automata, Languages and Combinatorics. Vol. 6, no. 4, p. 453-466, (2001)
2. Hopcroft, J. E., Motwani, R., Ullman, J. D., Introduction to automata theory, languages, and computation - international edition (2. ed). Addison-Wesley, (2003), ISBN 978-0201441246
3. Rabin, M. O., Scott, D., Finite automata and their decision problems, IBM J. Res. Dev., 3(2):114–125, (1959)