Mesterséges intelligencia alapú alkalmazások tesztelése

Testing approaches for AI-based applications

Dr. HORNYÁK Olivér
egyetemi docens

Miskolci Egyetem,
Informatikai Intézet,
3515 Miskolc-Egyetemváros, Egyetem út 1., oliver.hornyak@uni-miskolc.hu

Abstract

Modern artificial intelligence systems—especially deep-learning and generative models—contain numerous non-deterministic components. Stochastic learning, parallel execution, data augmentation, and GPU-dependent floating-point operations affect testability. This can lead to variance in outputs, intermittently failing (unstable) tests, and results that are difficult to reproduce. The paper examines the sources and mitigation of non-deterministic behavior and proposes stable metrics and measurement protocols.

Keywords: Artificial intelligence, software testing, reproducibility of results, performance evaluation, software metrics

Kivonat

A modern mesterséges intelligencia rendszerek – különösen a mélytanuló és generatív modellek – számos nem-determinista komponenst tartalmaznak. A sztochasztikus tanulás, a párhuzamos futtatás, az adat-augmentáció, GPU-függő lebegőpontos műveletek a tesztelhetőséget befolyásolják. Ez a kimenetek varianciájához, nem stabilan viselkedő tesztekhez és nehezen reprodukálható eredményekhez vezethet. A cikk a nem-determinista viselkedés forrásaival, kezelésével, stabil metrikák és mérési protokollok kialakításával foglalkozik.

Kulcsszavak: mesterséges intelligencia, szoftvertesztelés, reprodukálhatóság, kiértékelés, szoftvermetrika

1. BEVEZETÉS

A mesterséges intelligencia (MI) alapú alkalmazások rohamos terjedése új minőségbiztosítási és tesztelési paradigmákat követel meg. Az MI-rendszerek sajátossága, hogy viselkedésüket a tanítóadat és a futtatási környezet együtt alakítja; a funkcionalitás jelentős része nem explicit kódszabályokból, hanem statisztikai mintázatokból „tanulódik ki". Emiatt a klasszikus tesztelési megközelítések – amelyek stabil specifikációra és determinisztikus viselkedésre épülnek – önmagukban nem elegendőek. A szakirodalom az ilyen rendszerek körül felhalmozódó rejtett technikai problémára (pl. adat-/modell-összefonódás, metrikafüggőség) is figyelmeztet, amely a karbantartási költségeket és a hibakockázatot egyaránt növeli [1].
Az ipari gyakorlat azt mutatja, hogy az MI-fejlesztés életciklusát (adatgyűjtés, kísérletezés, tanítás, integráció, telepítés, monitorozás) a hagyományos szoftvermérnöki folyamatokba kell illeszteni, de ezek a folyamatok több ponton módosulnak: nő az adatelőkészítésre fordított erőfeszítés, a hibák jelentős része adat- és konfigurációs eredetű, és új kompetenciák válnak kulcsfontosságúvá [2]. Empirikus vizsgálatok szerint a legnagyobb ipari nehézségek a tesztadatok előállításánál, a tesztvégrehajtásnál és az eredmények értelmezésénél jelentkeznek; mindezt módszertani és eszköztámogatási hiányok súlyosbítják [9].
Az MI-rendszerek tesztelésének egyik alapvető nehézsége az, hogy a várható helyes viselkedés sok esetben nem írható le egzakt specifikációval. Ezért előtérbe kerülnek a tulajdonságalapú és relációalapú módszerek, különösen a metamorf tesztelés (MT), amelyben előre megfogalmazott metamorf relációk

(pl. invarianciák, monotonitás, ekvivalens transzformációk) alapján következtetünk a kimenetek konzisztenciájára [4].

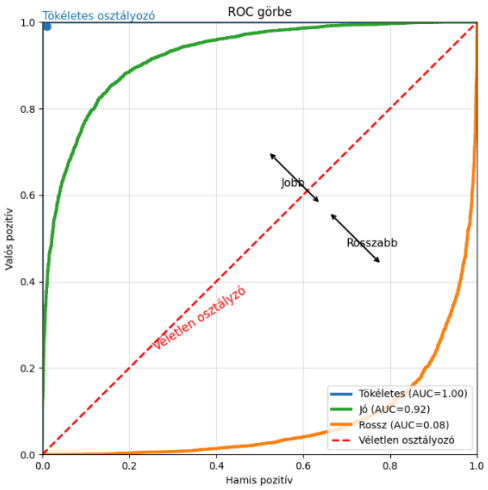
A mélytanulás sajátos kockázatait vizsgáló kutatások új kritériumokat és tesztelési megközelítéseket vezettek be. A DeepXplore neuron lefedettség- (neuron coverage) és többmodell-differenciára építő fehér-dobozos technika, míg a DeepTest valósághű képfeltételek (pl. köd, eső, fényviszonyok) generálásával keresi a speciális eseteket autonóm vezetési rendszerekben. Emellett a DeepGauge több-finomsági osztályba tartozó fedettségi mutatókat javasol (réteg-, neuronszintű és aktivációs statisztikák), a kombinatorikus tesztelés pedig a neurális állapottér robbanását csökkenti a tesztesetekkel való lefedettséget figyelembe vevő input-konstrukcióval [5–8]. Ezek a módszerek ugyan nem helyettesítik a klasszikus funkcionális és nem-funkcionális teszteket, de kifejezetten az MI-specifikus hibák (robosztusság, generalizációs hiányosságok, érzékenység) feltárására fókuszálnak.

2. OSZTÁLYOZÁS JELLEGŰ FELADATOK

Az MI alkalmazási területeinek egy jelentős része osztályozás jellegű [10]. Az osztályozók elkülönítő képessége azt méri, mennyire rangsorolja a modell „feljebb” a valódi pozitív eseteket a negatívaknál, függetlenül attól, hol húzzuk meg a döntési küszöböt. A legelterjedtebb küszöbfüggetlen mutató [15–16] a ROC-görbe alatti terület (ROC–AUC), amely a véletlenül választott pozitív magasabb pontszámot kap, mint egy véletlenül választott negatív esemény valószínűségeként is értelmezhető; széles körben használható általános szeparációs mérőszámként.

Erősen kiegyensúlyozatlan adatoknál – amikor a pozitív osztály ritka – informatívabb a precízió–visszahívás térben mért PR–AUC, azaz átlagos precízió, mert közvetlenül a pozitív osztály teljesítményére koncentrál, és a találati arány–visszahívás kompromisszumát ragadja meg; a szakirodalom kimutatja, hogy ilyen helyzetekben a PR–görbe és az abból származtatott terület jobban tükrözi a gyakorlati hasznosságot, mint a ROC–AUC.

Ha egyetlen küszöbhöz szeretnénk összefoglalószámot, a Matthews-korrelációs együttható (MCC) és a kiegyensúlyozott pontosság kevésbé torzít az uralkodó osztály irányába, mert a teljes konfúziós mátrix információját, illetve az osztályonkénti találati arányt egyformán veszik figyelembe; több vizsgálat szerint az MCC különösen robusztus alternatíva az egyszerű pontossággal vagy az F1-gyel szemben.



1. ábra. *ROC-görbe*

2.1 KALIBRÁCIÓ

Az osztályozásban a kalibráció azt írja le, mennyire felelnek meg a modell által kiadott valószínűségek a tényleges gyakoriságoknak. A kalibráció nem azonos az elkülönítő képességgel: előfordulhat, hogy egy jól szeparáló modell súlyosan túl- vagy alulmagabiztos. A leggyakrabban használt pontozófüggvények például a Brier-pontszám és a negatív log-valószínűség (log-loss) kifejezetten jutalmazzák a helyes valószínűségeket: elméleti értelemben akkor és csak akkor minimálisak, ha a becslések megegyeznek az igaz valószínűségekkel, ezért stabil alapot adnak a kalibráció számszerűsítésére [11][12][13].

A kalibráció diagnosztikájához a kalibrációs görbe (reliability diagram) nyújt szemléletes képet: a becsült valószínűségek átlagait vetjük össze a megfigyelt találati arányokkal, és azt várjuk, hogy a görbe a

főátló közelében fusson. A görbe összegző, skálázható megfelelője az Expected Calibration Error (ECE), amely a binelt eltérések súlyozott átlaga; a modern mélyhálók tipikusan túlmagabiztosak, amit az ECE és a reliability diagram együtt jól feltár, különösen akkor, ha konfidenciaintervallumokat is közlünk [14]. A pontozófüggvények és a grafikus/aggregált mérők együttese gyakorlatban is bevált: a Brier/log-loss numerikus stabilitást és döntéselméleti megfelelést ad, a görbe és az ECE pedig a hibatípus (alul- vagy túlkalibráltság) természetét világítja meg [11][12][14].

2.2 TIPIKUS MI FELADATOK

Az alábbi táblázat gyors áttekintést ad a leggyakoribb MI-feladatokról, és összerendezi őket a megfelelő tanulási paradigmákkal, kimenettípusokkal, tipikus bemenetekkel. Látható, hogy a feladatok nagy száma miatt általánosan tesztelési metodika helyett feladatspecifikus megközelítés szükséges.

1. táblázat – Tipikus MI feladatok

Feladat	Tanulási paradigma	Kimenet típusa	Tipikus bemenet/modalitás
Bináris / többosztályos osztályozás	Felügyelt	Címke (diszkrét)	Táblázatos, szöveg, kép
Regresszió	Felügyelt	Valós érték	Táblázatos, idősortáblázat
Rangsorolás / információkeresés	Felügyelt / tanár-nélküli jelzéssel	Rendezett lista	Keresési lekérdezés + dokumentum jellemzők
Anomáliadetektálás	Felügyelt / félig felügyelt / felügyelet nélküli	Bináris/score	Idősor, hálózati napló, szenzor
Klaszterezés	Felügyelet nélküli	Csoportcímke (utólag)	Táblázatos, szöveg, kép
Dimenziócsökkentés / reprezentáció	Felügyelet nélküli / önfelügyelt	Beágyazás	Kép, szöveg, multimodális
Objektumdetekció (kép)	Felügyelt	Doboz + címke	Kép
Szemantikus szegmentálás (kép)	Felügyelt	Pixelcímke térkép	Kép
Gépi fordítás (NLP)	Felügyelt / önfelügyelt	Szöveg	Párhuzamos korpusz
Kérdés–válasz / olvasásértés	Felügyelt	Szöveg / span	Korpuszból kivonat
Beszédfelismerés (ASR)	Felügyelt	Szöveg	Hang
Ajánlórendszerek	Felügyelt / implicit	Top-N lista / score	Kliens × tétel logok
Megerősítéses tanulás (vezérlés)	RL	Politika / akció	Állapot-átmenetek
Idősor-előrejelzés	Felügyelt	Idősor	Univariáns/multivariáns TS
Multimodális tanulás	Felügyelt / önfelügyelt	Különféle	Kép+szöveg, kép+hang

3. ÜZEMELTETÉSI METRIKÁK MI-SZOLGÁLTATÁSOKHOZ

Egy MI-rendszernek nemcsak okosnak, hanem folyamatosan elérhetőnek, gyorsnak és megfizethetőnek is kell lennie. Az üzemeltetési metrikák azt figyelik, hogyan teljesít a szolgáltatás mint egész: milyen gyorsan válaszol, mennyi hibát termel, mennyire stabil a terhelés alatt, és mindezt milyen erőforrás- és pénzköltséggel éri el. Ezek a mutatók kiegészítik a modell pontosságát: lehet egy modell kiváló offline, ha közben a felhasználó a lassúság vagy a gyakori hibák miatt elpártol.

A sebességet legjobban a késleltetés percentilisei írják le [22][23] A p95 például azt jelenti, hogy a kérések 95%-a ezen idő alatt választ kap; a felhasználói élményt jellemzően nem az átlag, hanem a kiegészítő halmaz (p95–p99) határozza meg, mert a ritka, de nagyon lassú válaszok is frusztrálók. Nagy forgalomban az apró késleltetések összeadódnak, ezért fontos az olyan tervezés, amely nem növeli feleslegesen a párhuzamos hívások számát, és értelmes időkorlátokkal, újrapróbálásokkal kezeli a lassulásokat.

A hibaarány önmagában kevés, ezért érdemes külön-külön követni a klienshibákat (rossz kérés), a szerverhibákat (belső hiba), a modell-szolgáltatási problémákat (időtúllépés, nincs betöltött modell), valamint az adatminőségi gondokat (hiányzó vagy késve érkező jellemzők). A pontos diagnózis kulcsa a metrikák szegmentálása végpont, régió, ügyféltípus és modellverzió szerint, mert a problémák gyakran csak az egyik szeletben jelentkeznek.

Hogy mindez ne csak mérés, hanem vállalás is legyen, szolgáltatási célokat megfogalmazni. Egy egyszerű példa: „a p95 késleltetés 120 ms alatt marad, a sikeres válaszok aránya pedig legalább 99,9%”. MI-szolgáltatásnál ezek mellé kerülnek modell-szintű célok is, például hogy a valós felhasználói visszajelzés

alapján a pozitív jelzések pontossága egy napon belül bizonyos szint felett legyen. Az ilyen célok segítenek abban, hogy a kiadások és kísérletek ne rontsák le észrevétlenül a szolgáltatás minőségét.

A költség és a kapacitás külön figyelmet érdemel. Praktikus a költséget egységes formában jelenteni (költség/1000 kérés vagy költség/előrejelzés), és mellette mérni a GPU/CPU kihasználtságot, a batch-méretet, a párhuzamosságot és a gyorsítótár találati arányát. Ezek együtt mutatják meg, hol szivárog el a pénz: sokszor nem is a modell „drága”, hanem a kis batch-ek, a gyakori modell-betöltések (cold start) vagy a rosszul méretezett skálázás.

A valós üzleti sikerességet az egyéb minőségi mutatók jelzik. Ajánlórendszernél ilyen a kattintási arány vagy a konverzió, keresésnél a találati oldalak mélysége és a visszapattanás.

Az adatok és a modell frissessége üzem közben folyamatosan romolhat, ha változik a környezet. Emiatt érdemes önállóan mérni, mennyire frissek a jellemzők (adatkor), mekkora a különbség a tanításkor és a kiszolgáláskor használt be-/kimenetek között és kimutatható-e eloszláseltolódás a forgalomban. Ilyenkor segítenek a reprodukálható, statisztikai összehasonlító tesztek; nem az a cél, hogy minden apró eltérést riasztás kövessen, hanem hogy a lényeges változásokat időben észrevegyük és újratanítással, utókalibrálással reagáljunk.

Végül érdemes figyelni a modell bizonytalanságára is. Ha egy időszakban hirtelen megemelkedik a bizonytalan előrejelzések aránya, az gyakran a bejövő adatok változását jelzi. Ilyenkor hasznos lehet ideiglenes védelmi lépés (például alacsonyabb kockázatú szabályok alkalmazása), miközben elindul a tényleges okok kivizsgálása és a modell frissítése.

az üzemeltetési metrikák akkor segítenek a legtöbbet, ha egyszerre fedik le a sebességet, a hibákat, a rendelkezésre állást, a költséget és a való életbeli minőséget; ha a célokat előre rögzítjük és mérhetően követjük; és ha minden fontos mutatót értelmes szeletekre bontva, a felhasználói élmény szempontjából vizsgálunk.

4. SPECIFIKÁCIÓS NEHÉZSÉGEK

A helyes viselkedés tipikusan kontextusfüggő, ezért általánosan nem adható meg teljes, zárt specifikációval. Egyrészt a működési környezet nyitott és dinamikus: a releváns tényezők (adatdisztribúció, jogi előírások, felhasználói preferenciák, ellenfél-modellek) időben változnak, így a rögzített követelmények gyorsan elavulnak. Másrészt a helyesség normatív komponenseket is tartalmaz (pl. méltányosság, ártalomminimalizálás, társadalmi elvárások), amelyek csak részlegesen formalizálhatók. Harmadrészt a rendszerek nagy állapottérrel és nemlineáris kölcsönhatásokkal rendelkeznek; ez a kombinatorikus komplexitás a teljes körű leírást gyakorlati és elméleti korlátok (pl. eldönthetetlenség) miatt is ellehetetleníti.

A mérnöki gyakorlat ennek megfelelően rétegzett megközelítést alkalmaz. A kritikus tulajdonságokat szigorú invariánsok és formális tulajdonságok (időlogikai korlátok, biztonsági limitációk) rögzítik, míg a teljesítményjellegű célok szolgáltatási szintű mutatókkal (percentilis késleltetés, hibaarány, hamis riasztás) kerülnek kalibrálásra. A nem determinisztikus vagy adatvezérelt komponenseket futásidejű teszteknek kell alávetni.

Különösen fontos az emberi szerep beépítése a folyamatba. A „human-in-the-loop” nem a gép gyengeségének, hanem a társadalmi beágyazottság elismerésének jele. A homályos, nagy kockázatú vagy normatív megfontolást igénylő szituációkban legyen világos, hogyan és mikor lép közbe emberi döntéshozó, hogyan dokumentáljuk a döntési indokokat, és miként tanul a rendszer a visszajelzésekből.

IRODALMI HIVATKOZÁSOK

[1] Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J.-F., Dennison, D. Hidden Technical Debt in Machine Learning Systems. Advances in Neural Information Processing Systems 28, Curran Associates, Inc., 2015, 2503–2511.

[2] Amershi, S., Begel, A., Bird, C., DeLine, R., Gall, H., Kamar, E., Nagappan, N., Nushi, B., Zimmermann, T. Software Engineering for Machine Learning: A Case Study. ICSE-SEIP ’19: Proceedings of the 41st International Conference on Software Engineering: Software Engineering in Practice, IEEE, 2019, 291–300.

[3] Zhang, J. M., Harman, M., Ma, L., Liu, Y. Machine Learning Testing: Survey, Landscapes and Horizons. IEEE Transactions on Software Engineering, IEEE, 2022, 48(1), 1–36.

[4] Xie, X., Ho, J. W. K., Murphy, C., Kaiser, G., Xu, B., Chen, T. Y. Testing and Validating Machine Learning Classifiers by Metamorphic Testing. Journal of Systems and Software, Elsevier, 2011, 84(4), 544–558.

[5] Pei, K., Cao, Y., Yang, J., Jana, S. DeepXplore: Automated Whitebox Testing of Deep Learning Systems. SOSP ’17: Proceedings of the 26th ACM Symposium on Operating Systems Principles, ACM, 2017, 1–18.

[6] Tian, Y., Pei, K., Jana, S., Ray, B. DeepTest: Automated Testing of Deep-Neural-Network-driven Autonomous Cars. ICSE ’18: Proceedings of the 40th International Conference on Software Engineering, ACM, 2018, 303–314.
[7] Ma, L., Juefei-Xu, F., Zhang, F., Sun, J., Xue, M., Li, B., Chen, C., Su, T., Li, L., Liu, Y., Zhao, J., Wang, Y. DeepGauge: Multi-Granularity Testing Criteria for Deep Learning Systems. ASE ’18: Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering, ACM, 2018, 120–131.
[8] Ma, L., Zhang, F., Xue, M., Li, B., Liu, Y., Zhao, J., Wang, Y. Combinatorial Testing for Deep Learning Systems. arXiv, https://arxiv.org/abs/1806.07723 (Utolsó letöltés: 2025. 09.09).
[9] Li, S., Guo, J., Xia, X., Chen, H., Lo, D., Jin, Z. Testing Machine Learning Systems in Industry: An Empirical Study. ICSE-SEIP ’22: Proceedings of the 44th International Conference on Software Engineering: Software Engineering in Practice, IEEE/ACM, 2022, 263–272.
[10] Schröder, T., Schulz, M. Monitoring Machine Learning Models: A Categorization of Challenges and Methods. Data Science and Management, KeAi Publishing, 2022, 5(3), 105–116.
[11] Brier, G. W., Verification of Forecasts Expressed in Terms of Probability. Monthly Weather Review. American Meteorological Society, 1950, 78(1), 1–3.
[12] Gneiting, T., Raftery, A. E., Strictly Proper Scoring Rules, Prediction, and Estimation. Journal of the American Statistical Association. American Statistical Association, 2007, 102(477), 359–378.
[13] Niculescu-Mizil, A., Caruana, R., Predicting Good Probabilities with Supervised Learning. Proceedings of the 22nd International Conference on Machine Learning (ICML). ACM, 2005, 625–632.
[14] Guo, C., Pleiss, G., Sun, Y., Weinberger, K. Q., On Calibration of Modern Neural Networks. Proceedings of the 34th International Conference on Machine Learning (PMLR 70). PMLR, 2017, 1321–1330.
[15] Fawcett, T., An introduction to ROC analysis. Pattern Recognition Letters. Elsevier, 2006, 27(8), 861–874.
[16] Davis, J., Goadrich, M., The Relationship Between Precision-Recall and ROC Curves. Proceedings of the 23rd International Conference on Machine Learning (ICML). ACM, 2006, 233–240.
[17] Saito, T., Rehmsmeier, M., The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. PLOS ONE. Public Library of Science, 2015, 10(3), e0118432.
[18] Chicco, D., Jurman, G., The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genomics. BioMed Central, 2020, 21(1), 6.
[19] Velez, D. R., White, B. C., Motsinger, A. A., Bush, W. S., Ritchie, M. D., Williams, S. M., Moore, J. H., A balanced accuracy function for epistasis modeling in imbalanced datasets using multifactor dimensionality reduction. Genetic Epidemiology. Wiley, 2007, 31(4), 306–315.
[20] Massey Jr, F. J., The Kolmogorov–Smirnov Test for Goodness of Fit. Journal of the American Statistical Association. American Statistical Association, 1951, 46(253), 68–78.
[21] Hand, D. J., Till, R. J., A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classification Problems. Machine Learning. Kluwer Academic Publishers, 2001, 45(2), 171–186.
[22] Provost, F., Fawcett, T., Analysis and Visualization of Classifier Performance: Comparison under Imprecise Class and Cost Distributions. Proceedings of the Third International Conference on Knowledge Discovery and comparison under imprecise class and cost distributions
[22] Beyer, B., Jones, C., Petoff, J., Murphy, N. R. (szerk.), Site Reliability Engineering: How Google Runs Production Systems. O’Reilly, Sebastopol (CA), 2016.
[23] Dean, J., Barroso, L. A., The Tail at Scale. Communications of the ACM. ACM, 2013, 56(2), 74–80.