DETEKCIJA ANOMALIJA U BAZAMA PODATAKA PRIMENOM ALGORITAMA VEŠTAČKE INTELIGENCIJE

Lazar Kopanja[1]

Rezime: U eri digitalizacije baze podataka predstavljaju centralnu ulogu u pouzdanom upravljanju informacijama u brojnim domenima, uključujući medicinu, finansije i industriju. Međutim, s obzirom na rastuću složenost i obim podataka, identifikacija anomalija postaje sve veći izazov. Ovaj rad se bavi analizom primene veštačke inteligencije u procesu detekcije anomalija u bazama podataka, sa posebnim osvrtom na tehnike mašinskog učenja. Razmatrani su pristupi nadziranog, nenadziranog i polunadziranog učenja, kao i njihove primene u realnim sistemima. Fokus je stavljen na biblioteke otvorenog koda u Python programskom jeziku, uključujući scikit-learn, PyOD i TensorFlow, koje omogućavaju razvoj efikasnih i skalabilnih rešenja. Takođe, opisani su konkretni algoritmi: KNN, One-Class SVM, Isolation Forest, autoenkoderi, LOF i DBSCAN uz prateće kod primere. Cilj rada je da pruži teorijski uvid i ponudi praktične smernice za savremene metode detekcije anomalija, doprinoseći time automatizaciji kontrole podataka u kompleksnim informacionim sistemima.

Ključne reči: veštačka inteligencija, algoritam, baze podataka, detekcija anomalija, Python

ANOMALY DETECTION IN DATABASES THROUGH ARTIFICIAL INTELLIGENCE ALGORITHMS

Abstract: In the era of digitalization, databases play a central role in the reliable management of information across various domains, including medicine, finance, and industry. However, with the increasing complexity and volume of data, detecting anomalies is becoming an ever more challenging task. This paper focuses on the application of artificial intelligence in the process of anomaly detection in databases, with particular emphasis on machine learning techniques. It explores supervised, unsupervised, and semi-supervised learning approaches, as well as their implementation in real-world systems. Special attention is given to open-source libraries in the Python programming language, such as scikit-learn, PyOD, and TensorFlow, which support the development of efficient and scalable solutions. Furthermore, specific algorithms are described—KNN, One-Class SVM, Isolation Forest, autoencoders, LOF, and DBSCAN—accompanied by example code. The aim of this work is to provide a theoretical overview and offer practical guidelines on modern anomaly detection methods, thereby contributing to the automation of data monitoring in complex information systems.

Keywords: Artificial intelligence, Algorithm, Databases, Anomaly detection, Python

1. UVOD

U savremenom digitalnom dobu, baze podataka predstavljaju ključnu infrastrukturu za organizovano skladištenje i obradu masovnih i raznovrsnih skupova informacija. Obezbeđenje integriteta, tačnosti i konzistentnosti podataka od suštinske je važnosti za donošenje validnih odluka u različitim oblastima primene, uključujući poslovne sisteme, medicinsku dijagnostiku, finansijsko upravljanje i druge kompleksne domene. Međutim, usled eksponencijalnog rasta obima i heterogenosti podataka, detekcija nepravilnosti, grešaka i potencijalnih bezbednosnih incidenata – poznatih kao anomalije – postaje izuzetno izazovna. Detekcija anomalija u kontekstu baza podataka podrazumeva identifikaciju zapisa koji značajno odstupaju od ustaljenih obrazaca ponašanja i distribucije. U tom smislu, sve je izraženija potreba za primenom efikasnih, skalabilnih i automatizovanih metoda detekcije anomalija, posebno u okruženjima obeleženim visokom dinamikom i kompleksnošću podataka, gde konvencionalni pristupi ručnog nadzora postaju nedovoljni [1].

[1] Visoka tehnička škola strukovnih studija u Novom Sadu, Školska 1, Novi Sad, e-mail: kopanja.l@vtsns.edu.rs

Tradicionalni pristupi otkrivanju nepravilnosti često se pokazuju kao nedovoljni u savremenim, složenim i dinamičnim okruženjima. Zbog toga se sve veća pažnja posvećuje primeni algoritama veštačke inteligencije, posebno onih iz oblasti mašinskog učenja, koji omogućavaju automatizovano i efikasno prepoznavanje anomalija. Cilj ovog rada je da predstavi teorijski pregled različitih AI tehnika za detekciju anomalija u bazama podataka, uz prateće primere implementacije u programskom jeziku Python, ukazujući na potencijalne primene u oblastima kao što su medicina, finansije, sajber bezbednost i industrijska automatizacija. Rad se fokusira na pregled metoda koje su dostupne kao biblioteke otvorenog koda i koje se lako mogu integrisati u postojeće informacione sisteme.

2. VEŠTAČKA INTELIGENCIJA I DETEKCIJA ANOMALIJA

Veštačka inteligencija obuhvata skup tehnika i algoritama koji omogućavaju računarima da analiziraju podatke, uče iz njih i samostalno donose zaključke, bez potrebe za unapred definisanim pravilima. U domenu detekcije anomalija, AI pokazuje posebno veliki potencijal, budući da su anomalije retke, heterogene i teško predvidive [2]. AI pristupi detekciji anomalija uključuju nadzirano učenje, koje koristi označene podatke, nenadzirano učenje, koje otkriva obrasce u neoznačenim podacima, i polunadzirano učenje, koje kombinuje malu količinu označenih podataka sa neoznačenima.

Detekcija anomalija pomoću veštačke inteligencije nalazi primenu u različitim oblastima kao što su: *medicina* – identifikacija neobičnih obrazaca u medicinskim slikama, laboratorijskim rezultatima, kao i EKG i EEG signalima [3]; *finansije* – detekcija prevara u finansijskim transakcijama, neobičnih tokova novca i krađe identiteta [4]; *sajber bezbednost* – prepoznavanje pokušaja upada, zloupotrebe privilegija i nepravilnog ponašanja korisnika [5]; *industrijska automatizacija* – identifikacija kvarova u senzorima i odstupanja u proizvodnim procesima [6].

U brojnim praktičnim sistemima realne implementacije algoritama veštačke inteligencije obuhvataju širok spektar domena. U medicinskim primenama koriste se autoenkoderi i One-Class SVM algoritmi za detekciju oboljenja na osnovu EKG signala (npr. PhysioNet Challenge sistemi) [7]. U finansijskim transakcijama prevare u realnom vremenu se detektuju koristeći Isolation Forest i autoenkodere (npr. sistemi MasterCard Fraud Detection, PayPal) [8]. Otkrivanje anomalija u mrežnom saobraćaju i log fajlovima često se realizuje primenom nenadziranih metoda mašinskog učenja, pri čemu se kao standardni skupovi podataka za obuku i evaluaciju koriste KDD99 i UNSW-NB15. Ovi pristupi se sve češće integrišu u moderne SIEM alate radi automatizovane analize bezbednosnih događaja [9].

3. ALGORITMI ZA DETEKCIJU ANOMALIJA

3.1. Python biblioteke

U programskom jeziku Python postoje biblioteke koje omogućavaju razvijanje, evaluaciju i integraciju modela za detekciju anomalija u šire softverske sisteme. Najčešće se koriste sledeće biblioteke: **scikit-learn** – obuhvata implementacije metoda kao što su KNN, One-Class SVM, Isolation Forest, DBSCAN i LOF; **PyOD** – specijalizovana biblioteka za otkrivanje anomalija koja integriše veliki broj modela, uključujući i različite varijante autoenkodera; **TensorFlow / Keras** – koristi se za kreiranje i obučavanje neuronskih mreža, posebno dubokih autoenkodera; **pandas** i **NumPy** – omogućavaju efikasnu obradu, analizu i pripremu podataka za modele; **matplotlib** i **seaborn** – služe za vizuelizaciju rezultata, identifikaciju obrazaca i interpretaciju modela.

U Tabeli 1 su prikazane prednosti i ograničenja navedenih Python biblioteka za detekciju anomalija.

***Tabela 1**. Pregled Python biblioteka za detekciju anomalija*

Biblioteka	Prednosti	Ograničenja
scikit-learn	- Dobro dokumentovana i stabilna - Podržava osnovne algoritme (One-Class SVM, Isolation Forest, LOF, DBSCAN) - Laka integracija	- Ograničena podrška za duboke modele - Nije optimizovana za vremenske serije
PyOD	- Fokusirana isključivo na detekciju anomalija - Više od 40 modela - Jednostavna za korišćenje i testiranje	- Manje efikasna za velike skupove podataka - Ograničenja kod kompleksnih neuronskih mreža
TensorFlow / Keras	- Fleksibilna i moćna za duboke modele (autoenkoderi, GAN-ovi) - Podrška za GPU - Idealna za složene obrasce	- Veća složenost implementacije - Duže vreme treniranja
pandas, NumPy	- Osnovni alati za obradu podataka - Neophodni za pripremu ulaza - Brzi i efikasni za numeričke operacije	- Ne pružaju metode za detekciju anomalija direktno
matplotlib, seaborn	- Omogućavaju vizualizaciju rezultata - seaborn nudi napredne statističke prikaze i estetiku	- Samo za prikaz - ne vrše detekciju anomalija

3.2. Algoritmi za detekciju anomalija

3.2.1. K najbližih suseda (KNN)

KNN algoritam u kontekstu detekcije anomalija koristi udaljenost do K najbližih suseda kao meru za procenu verovatnoće anomalije. Ako je udaljenost instance do njenih suseda velika, verovatno je reč o anomaliji. KNN je jednostavan i intuitivan, ali zahteva značajnu procesorsku snagu pri velikim skupovima podataka [10]. Sledeći kod za detekciju anomalija pomoću K najbližih suseda (KNN) generiše skup od 100 nasumičnih tačaka u 2D prostoru:

```
from sklearn.neighbors import NearestNeighbors
import numpy as np
X = np.random.rand(100, 2)
nbrs = NearestNeighbors(n_neighbors=5).fit(X)
distances, _ = nbrs.kneighbors(X)
anomaly_scores = distances.mean(axis=1)
```

Model se trenira tako da za svaku tačku može pronaći 5 najbližih suseda. Pronalazi rastojanja do 5 najbližih suseda za svaku tačku. Računa prosečnu udaljenost do suseda, što se koristi kao skor anomalije – veći skor znači veća verovatnoća da je tačka anomalna.

3.2.2. One-Class SVM

One-Class SVM je algoritam koji uči graničnu površinu oko normalnih podataka i detektuje tačke izvan te granice kao anomalije. Pogodan je za situacije gde postoji veliki broj normalnih, ali malo ili nimalo označenih anomalnih primera, zbog čega se često koristi u nenadziranom kontekstu [11]. Kod za One-Class SVM pri čemu je promenljiva X prethodno definisana i odgovarajuće formatirana:

```
from sklearn.svm import OneClassSVM
model = OneClassSVM(kernel='rbf', nu=0.1)
model.fit(X)
y_pred = model.predict(X)
```

U ovom kodu se inicijalizuje model sa *RBF* kernelom (koristi Radial Basis Function što je uobičajeno za nelinearnu separaciju podataka) i parametrom *nu* koji predstavlja gornju granicu za udeo anomalija koje model toleriše (u ovom slučaju do 10%).

3.2.3. Isolation Forest

Isolation Forest se, za razliku od tradicionalnih pristupa koji pokušavaju da modeluju normalne obrasce ponašanja, zasniva na principu izolacije [12]. Algoritam gradi više nasumičnih stabala za izolaciju (isolation trees) tako što kod svakog čvora slučajno bira atribut i tačku preseka i na taj način rekurzivno deli skup podataka. Budući da su anomalije retke i različite, one se obično mogu izolovati u manjem broju podela (imaju manju dubinu u stablima, tzv. plitka stabla). Nasuprot tome, normalni podaci, koji se nalaze u gušćim regionima, zahtevaju veći broj podela i samim tim imaju veću prosečnu dubinu (dublja stabla). Kod za korišćenje Isolation Forest algoritma u scikit-learn biblioteci:

```
from sklearn.ensemble import IsolationForest
model = IsolationForest(contamination=0.1, random_state=13)
model.fit(X)
scores = model.decision_function(X)
predictions = model.predict(X)
```

Parametar *contamination* označava očekivani udeo anomalija u skupu podataka (10%). Parametar *random_state* je postavljen na fiksnu vrednost radi obebedjivanja reproduktivnosti rezultata. Model se obučava nad podacima *X* bez potrebe za etiketiranim instancama, jer algoritam pripada domenu nenadziranog učenja. Funkcija *decision_function* vraća vrednosti koje predstavljaju skor anomalije, pri čemu viši rezultati ukazuju na veću verovatnoću da podatak nije anomalija. Funkcija *predict* klasifikuje instance tako da vrednost 1 označava normalne podatke, dok -1 označava anomalije.

3.2.4. Autoenkoderi

Autoenkoder predstavlja arhitekturu veštačkih neuronskih mreža koja se sastoji od dva međusobno povezana segmenta – deo za kodiranje i deo za dekodiranje. Kodirajući deo pretvara ulazne podatke u kompaktnu unutrašnju reprezentaciju smanjene dimenzionalnosti, dok dekodirajući deo nastoji da iz te reprezentacije rekonstruše originalni ulaz. Tokom procesa učenja, mreža pokušava da minimizira razliku između originalnog i rekonstruisanog podatka. Ako se za određeni podatak uoči značajno veća greška rekonstrukcije u poređenju sa ostalima, to može ukazivati na anomaliju. Zbog ove osobine, autoenkoderi su naročito efikasni pri identifikaciji složenih, nelinearnih odstupanja u podacima visoke složenosti [13].

3.2.5. Local Outlier Factor (LOF)

Algoritam Local Outlier Factor (LOF) koristi se za nenadzirano prepoznavanje anomalija tako što upoređuje gustinu tačke sa gustinom njenih najbližih suseda. Ako neka tačka pripada oblasti sa znatno manjom gustinom u poređenju sa okolinom, verovatnije je da se radi o anomaliji [14]. Gustina se procenjuje na osnovu udaljenosti do unapred definisanog broja suseda (npr. 20 najbližih tačaka):

```
from sklearn.neighbors import LocalOutlierFactor
model = LocalOutlierFactor(n_neighbors=20)
y_pred = model.fit_predict(X)
```

Funkcija *fit_predict* odmah trenira model i vraća predikcije, vrednost -1 za *y_pred* označava detektovanu anomaliju, a vrednost 1 normalnu tačku.

3.2.6. DBSCAN

DBSCAN je algoritam za klasterizaciju koji identifikuje regije sa visokom gustinom podataka, dok tačke koje ne pripadaju nijednoj regiji klasifikuje kao šum, odnosno anomalije [15].

```
from sklearn.cluster import DBSCAN
```

```
model = DBSCAN(eps=0.3, min_samples=5)
labels = model.fit_predict(X)
```

Parametar *eps* predstavlja maksimalnu udaljenost između tačaka koje se mogu smatrati delom istog klastera. Manja vrednost podrazumeva strože kriterijume za klasterizaciju. Parametar *min_samples* predstavlja minimalan broj tačaka koji je potreban da bi se formirao klaster. Funkcija *fit_predict* trenira model i vraća predikcije, tako da promenljiva *labels* sadrži klastere (brojevi) ili vrednost -1 za šum, tj. Anomalije.

4. PERFORMANSE ALGORITAMA

Za objektivno poređenje performansi algoritama za detekciju anomalija koriste se metrike poput preciznosti, odziva, F1 skora i AUC-ROC vrednosti, koje ocenjuju tačnost i sposobnost modela da razlikuje anomalije od normalnih podataka. Evaluacija se najčešće sprovodi na unapred označenim skupovima podataka (npr. KDD99, NAB), pri čemu rezultati zavise od karakteristika podataka, predobrade i podešavanja modela. AI algoritmi za detekciju anomalija mogu se integrisati u DBMS sisteme putem offline analize, real-time API integracija ili direktnog uvođenja modela unutar baze (npr. PL/Python u PostgreSQL-u). Ova integracija omogućava pravovremeno prepoznavanje sumnjivih aktivnosti, automatske reakcije i adaptivno učenje modela u skladu s promenama u podacima.

Razmatrani algoritmi za otkrivanje anomalija koriste različite pristupe, a njihov izbor zavisi od vrste podataka i ciljeva analize. KNN je jednostavan i lako se implementira, ali slabo funkcioniše u prostorima sa mnogo dimenzija, gde udaljenosti gube značaj. One-Class SVM ima snažnu teorijsku osnovu i dobre rezultate na strukturiranim podacima, ali teško skalira zbog osetljivosti na parametre i velike računarske zahtevnosti. Isolation Forest je brz, efikasan i ne zavisi od distribucije podataka, što ga čini pogodnim za obradu velikih skupova i rad u realnom vremenu. Autoenkoderi omogućavaju detekciju složenih i nelinearnih anomalija pomoću neuronskih mreža, ali zahtevaju mnogo podataka i pažljivo podešavanje arhitekture. LOF koristi lokalnu gustinu za otkrivanje anomalija u manjim okruženjima, ali može biti nestabilan u prisustvu šuma i osetljiv na broj suseda. DBSCAN, iako namenjen klasterizaciji, može efikasno identifikovati anomalije kao tačke koje ne pripadaju nijednom klasteru; prednost mu je što ne zahteva unapred poznat broj klastera, ali izbor parametara zahteva dodatno podešavanje. U Tabeli 2 je dat pregled algoritama za detekciju anomalija. Podaci prikazani u Tabeli 2 predstavljaju uopštene orijentacione karakteristike algoritama i mogu varirati u zavisnosti od konkretne implementacije, vrste podataka, veličine uzorka, kao i izbora parametara tokom obuke i testiranja.

***Tabela 2**. Algoritmi za detekciju anomalija*

Algoritam	Tip učenja	Skalabilnost	Pogodnost za visoke dimenzije	Potrebni parametri	Detekcija lokalnih anomalija	Otporan na šum
KNN	Nenadzirano	Slaba	Ne	n_neighbors	Delimično	Ne
One-Class SVM	Nadzirano	Slaba	Delimično	kernel, nu	Ne	Ne
Isolation Forest	Nenadzirano	Dobra	Da	random_state, contamination	Ne	Da
Autoenkoder	Nadzirano/ Nenadzirano	Dobra	Da	Arhitektura, epohe, loss	Ne	Delimično
LOF	Nenadzirano	Srednja	Delimično	n_neighbors	Da	Ne
DBSCAN	Nenadzirano	Srednja	Ne	eps, min_samples	Da	Da

5. ZAKLJUČAK

Savremeni informacioni sistemi suočeni s ogromnim količinama podataka zahtevaju pouzdane metode za otkrivanje neuobičajenih ponašanja i potencijalnih grešaka. U tom

kontekstu, veštačka inteligencija, a posebno tehnike mašinskog učenja, sve više se nameću kao nezaobilazna rešenja za detekciju anomalija u bazama podataka. Zahvaljujući dostupnosti biblioteka otvorenog koda u programskom jeziku Python moguće je razviti efikasne modele koji se mogu prilagoditi potrebama korisnika u oblastima poput medicine, finansija, sajber bezbednosti i industrije.

U ovom radu dat je teorijski pregled najzastupljenijih pristupa, uz ilustrativne primere koda u programskom jeziku Python. Prikazani algoritmi, KNN, One-Class SVM, Isolation Forest, autoenkoderi, LOF i DBSCAN, svaki sa svojim prednostima i ograničenjima, omogućavaju automatizaciju i inteligentnu analizu podataka koje bi bile teško ostvarive tradicionalnim pristupima. Ove tehnike su se pokazale kao moćni alati za identifikaciju podataka koji odstupaju od uobičajenih obrazaca. Njihova primena doprinosi ranom otkrivanju anomalija, povećanju bezbednosti sistema i unapređenju procesa donošenja odluka. Dalji razvoj i kombinovanje ovih tehnika sa naprednim neuronskim mrežama otvara mogućnosti za još precizniju i robusniju detekciju anomalija u sve složenijim digitalnim okruženjima.

6. LITERATURA

[1] Khnaisser, C., Hamrouni, H., Blumenthal, D. B., Dignös, A., & Gamper, J. (2024). Efficiently Labeling and Retrieving Temporal Anomalies in Relational Databases. Information Systems Frontiers, 1-25.

[2] Sallam, A., Fadolalkarim, D., Bertino, E., & Xiao, Q. (2016). Data and syntax centric anomaly detection for relational databases. Wiley interdisciplinary reviews: data mining and knowledge discovery, 6(6), 231-239.

[3] Wu, J. C. H., Liao, N. C., Yang, T. H., Hsieh, C. C., Huang, J. A., Pai, Y. W., ... & Lu, H. H. S. (2024). Deep-Learning-Based Automated Anomaly Detection of EEGs in Intensive Care Units. *Bioengineering*, *11*(5), 421.

[4] Le Fort, E. (2018). *A Comparative Study of Machine Learning Algorithms* (Doctoral dissertation).

[5] Kimanzi, R., Kimanga, P., Cherori, D., & Gikunda, P. K. (2024). Deep Learning Algorithms Used in Intrusion Detection Systems--A Review. *arXiv preprint arXiv:2402.17020*.

[6] Erhan, L., Ndubuaku, M., Di Mauro, M., Song, W., Chen, M., Fortino, G., ... & Liotta, A. (2021). Smart anomaly detection in sensor systems: A multi-perspective review. *Information Fusion*, *67*, 64-79.

[7] Zhang, S., Fang, Y., & Ren, Y. (2024). ECG autoencoder based on low-rank attention. Scientific Reports, 14(1), 12823.

[8] Gadimov, E., & Birihanu, E. (2025). Real-time suspicious detection framework for financial data streams. International Journal of Information Technology, 1-17.

[9] Moustafa, N., & Slay, J. (2016). The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. Information Security Journal: A Global Perspective, 25(1-3), 18-31.

[10] Goldstein, M., & Uchida, S. (2016). A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. PloS one, 11(4), e0152173.

[11] Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. Neural computation, 13(7), 1443-1471.

[12] Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008, December). Isolation forest. In 2008 eighth ieee international conference on data mining (pp. 413-422). IEEE.

[13] Sakurada, M., & Yairi, T. (2014, December). Anomaly detection using autoencoders with nonlinear dimensionality reduction. In Proceedings of the MLSDA 2014 2nd workshop on machine learning for sensory data analysis (pp. 4-11).

[14] Breunig, M. M., Kriegel, H. P., Ng, R. T., & Sander, J. (2000, May). LOF: identifying density-based local outliers. In Proceedings of the 2000 ACM SIGMOD international conference on Management of data (pp. 93-104).

[15] Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996, August). A density-based algorithm for discovering clusters in large spatial databases with noise. In kdd (Vol. 96, No. 34, pp. 226-231).