

SISTEM BAZAT PE INTELIGENȚĂ ARTIFICIALĂ PENTRU ANALIZA AVANSATĂ A DATELOR DE DETECTARE A FISURILOR ÎN CONDUCTE DE TRANSPORT FLUIDE

Teză de doctorat – Rezumat

pentru obținerea titlului științific de doctor la
Universitatea Politehnică Timișoara
în domeniul de doctorat Ingineria Sistemelor

autor ing. Adrian-Ioan ARGEȘANU

conducător științific Prof. univ. dr. ing. Gheorghe-Daniel ANDREESCU

Decembrie 2023

Conductele reprezintă mijlocele principale pentru transportul în siguranță și eficient a produselor petroliere la presiuni mari pe distanțe lungi. Un accent principal este pus pe siguranța și disponibilitatea sistemului de conducte. Din perspectiva integrității, conductele pot fi tratate ca vase sub presiune. Orice anomalie sau defect trebuie detectate și identificate înainte ca acestea să aibă un efect dăunător asupra integrității conductei sau a întregului sistem [3]. Prin urmare, integritatea conductelor trebuie evaluată frecvent de operatorii acestora. Cu peste 90% din conducte aflate sub nivelul solului [1], localizarea potențialelor defecte reprezintă o provocare majoră.

Tipologia defectelor ce pot afecta în timp condiția conductelor este foarte variată. Acestea pot fi cauzate de diferiți factori, precum parametrii de funcționare ai conductei, condițiile ambientale sau chiar procesul de fabricație a acesteia. În general însă, defectele pot fi grupate în: coroziune și pierderi de metal, fisuri și defecte similare fisurilor, laminari și alte defecte la mijlocul peretelui metalic, și anomalii geometrice. În timp ce pierderea de metal este de departe cel mai răspândit fenomen, fisurarea este adesea cea mai îngrijorătoare amenințare la adresa siguranței conductelor, datorită potențialului său de a duce la defectare bruscă.

Fisurarea sub tensiune-corozivă (SCC) se dezvoltă în prezența simultană a tensiunii de tracțiune și a atacului de coroziune. Teoretic, aceasta poate apărea în orice punct în care tensiunea locală depășește valoarea tensiunii admisibile la tracțiune a materialului. SCC extern se dezvoltă de obicei sub forma unor fisuri foarte fine, aproape nedetectabile cu ochiului liber. Orientate de obicei longitudinal, acestea apar des în așa-numitele "colonii". Deoarece SCC se dezvoltă inițial lent, coloniile de fisuri pot exista în conducte timp de mulți ani fără a avea un impact asupra funcționării acesteia. Alungirea și adâncirea acestor fisuri duce în timp însă la interconectarea lor. Odată ajunse la dimensiune critică, conducta fie se va scurge, fie se va rupe. Datorită faptului că nu este posibilă o predicție fiabilă a apariției SCC, operatorii de conducte recurg la inspecții regulate.

Metodele convenționale de testare a integrității conductelor sunt efectuate "în conductă" folosind dispozitive inteligente de inspecție (PIGs). PIG-urile includ hardware electronic și senzori care colectează diverse forme de date în timpul călătoriei prin conductă. Având în vedere structura fisurilor, o tehnică de inspecție bazată pe ultrasunete care utilizează transductoare piezoelectrice este recomandată [4]. Modificări în structura cristalină a metalului (precum fisurile exemplificate în Figura 1) duc la reflectarea undei de ultrasunet. Prin efectul piezoelectric direct, aceste reflexii sunt traduse în impulsuri electrice, care ulterior sunt măsurate, procesate și analizate. Caracteristicile semnalului împreună cu momentul înregistrării

acestui, corelate cu datele înregistrate de senzorii adiacenți, furnizează informații despre geometria și poziția defectului.

UltraScan CDP (dispozitivul de inspecție „in conductă” (ILI) cu senzori piezoelectrice dezvoltat de Process & Pipeline Services, ilustrat în Figura 1) a fost proiectat pentru a detecta și măsura fisuri și colonii de fisuri orientate longitudinal, cu lungimi începând de la 25 mm și adâncimi începând cu 1 mm, în conducte care transportă medii lichide, la viteze de inspecție de până la 5 m/s [2]. Numărul de transductoare este proportional cu diametrul conductei inspectate, putând atinge câteva sute.



Figura 1. UltraScan CDP [2] și diferite tipuri de fisuri [11]

Datorită cantității mari de date înregistrate în timpul inspecției conductelor ce pot exceda o sută de kilometri în lungime, una dintre principalele limitări istorice de performanță ale dispozitivelor ILI a fost capacitatea de stocarea a datelor. Au fost făcute mai multe abordări cu ajutorul a diferiți algoritmi de compresie a datelor; UltraScan CDP în speța folosește algoritmul ALOK (Amplitude Time Locus Curves), care oferă o comprimare cu pierderi, dar cu rate foarte ridicate de compresie. Pe lângă compresie, datele înregistrate sunt scanate în timp real, iar secvențele de înregistrări care nu prezintă structuri de interes eliminate. La final, datele sunt stocate în fișiere binare.

Strategia de stocare utilizată de PIG-ul de detecție a fisurilor al Process & Pipeline Services are ca rezultat o reprezentare foarte rară a peretelui conductei pentru analize ulterioare. Agregarea tuturor valorilor compresionate înregistrate de un senzor într-un singur ciclu de inspecție sub forma unui A-Scan duce la o reprezentare 2D a amplitudinilor raportate la timp. Agregarea tuturor A-Scan-urilor înregistrate de un anumit senzor sub forma unui B-Scan duce la o reprezentare 3D a amplitudinilor raportate la timp și distanță. Suprapunerea tuturor B-Scan-urilor înregistrate de toți senzorii rezultă într-o vizualizare 4D, care aplatizată pe axa azimutului duce la o reprezentare 3D C-Scan. Vederea C-Scan prezintă întreaga circumferință a conductei. Exemple de C-Scan și B-Scan sunt ilustrate în Figura 2; amplitudinile sunt codate în culoare falsă.

După ce datele au fost analizate, un raport este trimis operatorului de conducte, care porcede la excavarea secțiunilor afectate în vederea remedierii defecțiunilor detectate. Costurile financiare uriașe ale excavărilor impun o analiză a datelor cu erori minime. În timp ce reflexiile înregistrate conțin informații despre natura și geometria fenomenelor întâlnite, detectarea, clasificarea și dimensionarea precisă a defectelor folosind unde ultrasonice ghidate rămân o

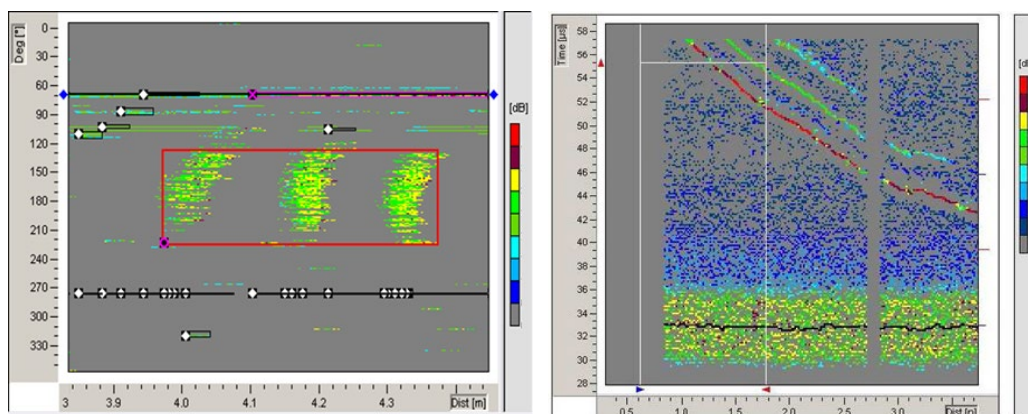


Figura 2. Vizualizare C-Scan (stânga) și B-Scan (dreapta) - exemple

provocare majoră în curs de investigare. Datorită complexității caracteristicilor propagării undelor, analiza datelor implică efort manual semnificativ din partea experților în domeniu (SME).

Pentru a reduce acest efort, se folosesc algoritmi ori de câte ori este posibil. Rezultatele algoritmilor sunt rareori publicate direct în raportul destinat operatorilor; acestea sunt în mare parte verificate manual de către SME. În cadrul analizei datelor înregistrate de UltraScan CDP, unul dintre principalele puncte de interes este automatizarea proiectată pentru detectarea și evidențierea zonelor cu potențiale defecte asemănătoare fisurilor. Din considerente de siguranță, acest algoritm este ajustat pentru a nu omite niciun defect semnificativ, făcându-l astfel extrem de sensibil la schimbări în caracteristicile conductei – cum ar fi detaliile de fabricație, parametrii mediului transportat în conductă, și curățenia peretelui acesteia – de exemplu prezența depunerilor de ceara. Împreună cu provocările generate de raritatea datelor înregistrate, această configurație duce la un număr mare de zone de interes – numite și regiuni de interes (ROI-uri) – create în jurul unor reflectori neinvazivi, cum ar fi suprafața aspră a conductei, ceara și alte depuneri ne-metalice, impurități în mediul transportat, zgomot electronic, interferențe între senzori, etc.

Figura 3 prezintă un rezumat al etapelor de evaluare. Datele sunt întâi descărcate, apoi procesate de către algoritmii automatizați. Rezultatele automatizărilor sunt evaluate manual de către SME ("Data Analysis" în figură). Rezultatele analizei manuale sunt compilate sub forma de rapoarte, listări și statistici ("Client Deliverables" în figură) și trimise operatorului conductei.

Cu cât sunt create mai multe ROI-uri non-defect, cu atât crește timpul necesar pentru analiza de către experți. Pentru a permite specialiștilor să se concentreze pe deciziile "de înaltă competență", numărul de ROI-uri non-defect trebuie redus. Deoarece considerentele de siguranță interzic ajustarea mai agresivă a automatizării, accentul este mutat pe prefiltrarea rezultatelor acesteia înainte de evaluarea manuală.

Obiectivul principal al acestei lucrări a fost dezvoltarea și operaționalizarea unui sistem bazat pe inteligență artificială, care să asiste experții în distingerea ROI-urilor defect de cele non-defect în datele de detectare a fisurilor înregistrate de PIG-ul UltraScan CDP dezvoltat de Process & Pipeline Services.

Antrenarea și validarea unui astfel de sistem necesită în primul rând acces de tip batch de mare viteză la structurile binare 4D puternic comprimate. Pentru a satisface această cerință, mai întâi a fost concepută și dezvoltată o **soluție de stocare a datelor NoSQL**. Bazele de date NoSQL, cunoscute și sub denumirile de "Not only SQL", "non-relaționale" și "non-SQL", au existat încă de la sfârșitul anilor 1960 [7], dar au fost acceptate în mod general doar la începutul

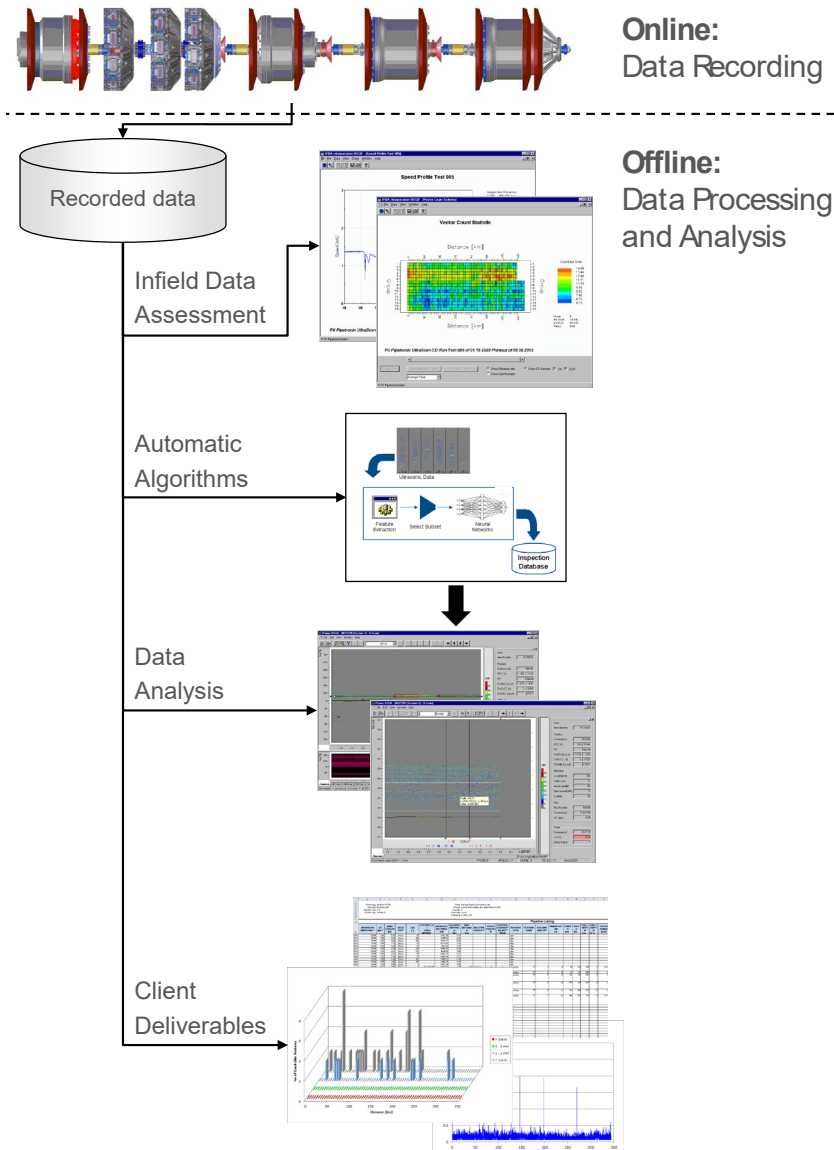


Figura 3. Fluxul datelor [11]

secolului XXI, servind aplicații de big data și web în timp real [8].

Aceste sisteme de stocare reprezintă o alternativă la bazele de date relaționale tradiționale și nu necesită o schemă fixă. Acest lucru înseamnă la rândul său, că bazele de date NoSQL pot stoca date structurate, semi-structurate, nestructurate și polimorfe, făcându-le deosebit de eficiente pentru date rare.

Sistemele NoSQL au fost concepute pentru a servi volume mari de date, tranzacționând scalarea tradițională (verticală) cu o abordare de scalare (orizontală) pentru a satisface cerințele de latență odată cu creșterea traficului.

Am efectuat un studiu comparativ cuprinzător al tehnologiilor NoSQL existente în contextul cerințelor de stocare și tranzacționare a datelor ale Process & Pipeline Services – ingestie one-time, prelucrare în batch și evaluare interactivă. HBase a fost identificat ca fiind candidatul cel mai potrivit pentru datele 4D ale UltraScan CDP. HBase este o bază de date distribuită, orientată pe coloane, modelată după Google BigTable [9]. Proiectată pentru stocarea cantităților extrem de mari de date într-o structură similară unei table, HBase oferă acces rapid și aleatoriu la datele găzduite, alături de diverse alte caracteristici și avantaje:

- Funcționare optimă în cazul schemelor variabile, unde structura înregistrărilor diferă,
- Citiri și scrieri strict consistente, oferind suport pentru interogări în timp real,
- Scalabilitate, cu partiționare automată și configurabilă la nivelul tabelelor.

Pentru stocarea datelor, am dezvoltat o schema variabilă în HBase – "un A-Scan per rând" (Figura 4). Conformă cu recomandarea „scheme de stocare înalte-dar-înguste” din literatura de specialitate, "un A-Scan per rând" a prezentat cel mai bun compromis pentru operațiunile de scanare („range-scans”) de diferite lungimi, ale datelor de inspecție rare ale UltraScan CDP.

RowKey			A-Scan	
Dataset UID	Odometer	Sensor UID	Meta Information	Recorded Values

Figura 4. Schema HBase: un A-Scan per rând

Pentru a remedia principala limitare de performanta a HBase, am dezvoltat o soluție software pentru accelerarea încărcării de date în mod batch în cluster-urile HBase optimizate pentru operațiuni de tip „range-scans”. Prevenind congestia la nivel de regiune (Region) și server de regiuni (Region Server), Pre-Shard și Shard-Allocation (Figura 5) garantează stadiul optimizat pentru interogarea datelor al tabelului chiar înainte de emiterea primei cerere de scriere, cu costul unei sarcini minime de calcul, fără să interfereze cu datele sau designul cheilor de rând existente.

Primul pas, numit Pre-Shard, pre-aloca regiuni în tabelul existent. Pre-Shard evaluează setul de date care urmează să fie ingerat și, luând în considerare structura cheii de rând, derivă intervalul cheilor pe care îl va ocupa în tabelul HBase. Pre-Shard calculează punctele de partiționare în conformitate cu schema de partiționare (RegionSplitPolicy) configurată la nivelul tabelului și parametrii acesteia, dar permite și specificarea unei scheme de partiționare diferite la runtime, pentru adaptarea granularității regiunilor și a volumului de date stocat per

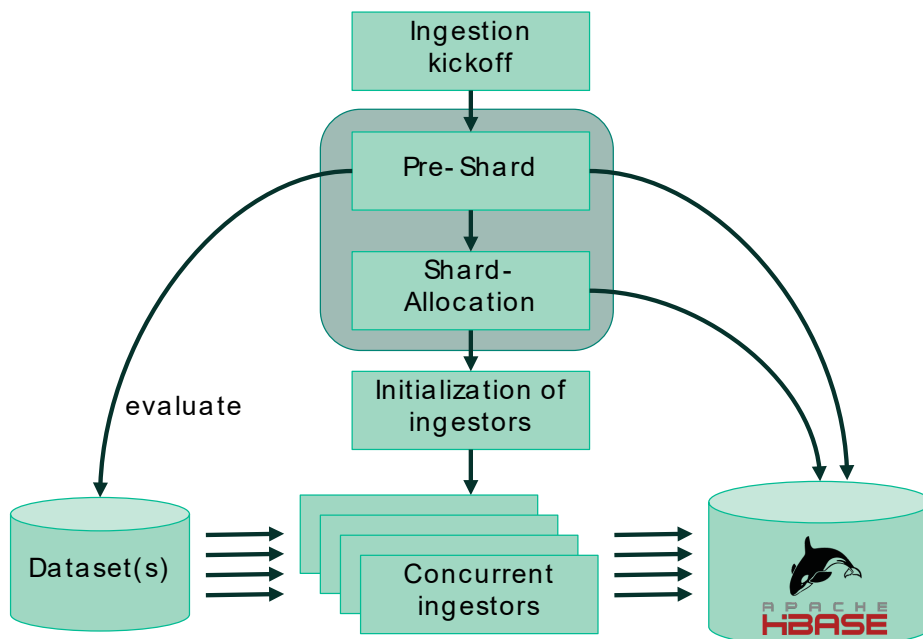


Figura 5. Integrarea Pre-Shard și Shard-Allocation

regiune în funcție de caracteristicile setului de date ingerat.

În HBase, balancer-ul ar detecta în cele din urmă regiunile alocate de Pre-Shard, iar avându-le pe toate situate pe un singur nod, ar alege un subset dintre acestea pentru mutarea lor pe alte noduri. Deoarece această automatizare nu ia în considerare vecinătatea intervalului cheilor de rând și efectele acesteia asupra încărcării în mod batch sau operațiilor de scanare, este probabil să apară hotspot-uri la nivelul serverelor de regiuni atât în timpul încărcării cât și a interogării ulterioare. Shard-Allocation a fost dezvoltat pentru a aborda acest neajuns. Utilitarul interoghează toate regiunile nou alocate și serverele de regiunii active de la HMaster și alocă partițiile nodurilor în maniera round-robin, astfel încât două regiuni cu intervale vecine de chei să nu fie niciodată situate pe același nod. Cu etapa a doua finalizată, operațiunea de ingerare poate fi reluată normal.

Compararea directă a performanței de ingestie în configurația de testare a arătat o scădere de 76% în numărul evenimentelor de blocare-și-așteptare atunci când Pre-Shard și Shard-Allocation au fost activate (Figura 6), echivalând cu de 4,2 ori mai multe înregistrări ingerate în medie între aceste evenimente, și o reducere medie de 47% a duratei de ingerare. [12] Funcțiile Pre-Shard și Shard-Allocation constituie o soluție aplicabilă tuturor instanțelor HBase care stochează date de tipul seriilor temporale.

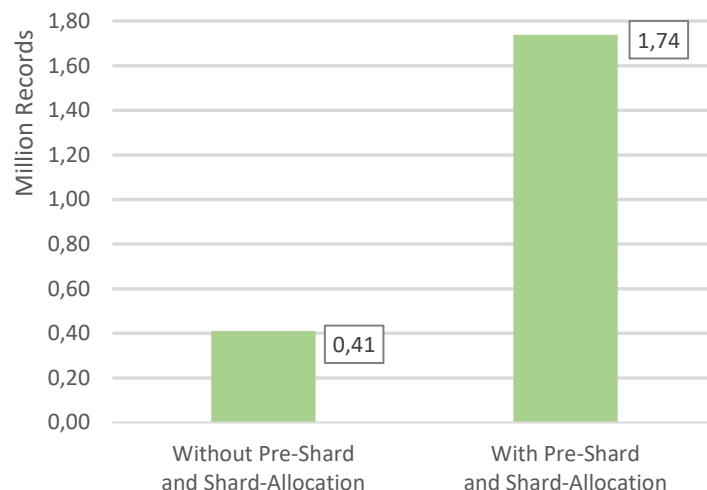


Figura 6. Numărul de înregistrări ingerate în medie între evenimentele blocare-și-așteptare

Cu datele istorice ale Process & Pipeline Services încărcate în baza de date NoSQL, am întors atenția spre prototiparea și operaționalizarea algoritmului, și am introdus **platforma personalizată pentru gestionarea întregului ciclu de viață al modelelor de învățare automată**. Oferind nativ capabilități de inferență offline, platforma propusă completează sistemele precum Bighead și Michelangelo. [13]

Platforma este construită în Python și se bazează pe tehnologii open-source precum Docker, Keras și altele. Proiectate în principal – dar nu limitate – să ruleze în Amazon Web Services (AWS), componentele individuale sunt oferite sub formă de containere Docker. Fiecare container Docker, denumit serviciu, încapsulează o singură funcționalitate, pe care o pune la dispoziție prin intermediul unui server HTTP. Datorită abordării containerizate, serviciile pot fi rulate atât pe hardware local, cât și în configurații virtualizate. Serviciile pot fi agregate dinamic pentru a aborda provocările în continuă schimbare ale diferitelor probleme de învățare automată. O configurație multi-serviciu este denumită „mediu”; mediile sunt orchestrate prin intermediul Docker Swarm.

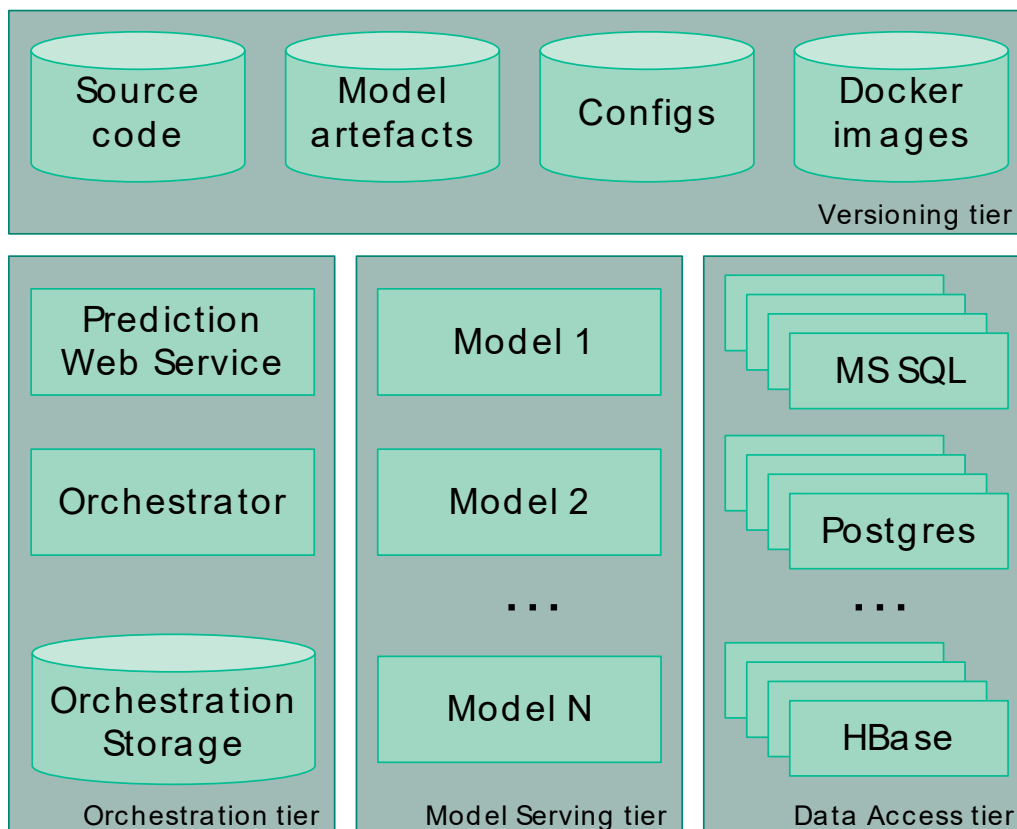


Figura 7. Arhitectura sistemului

Din punct de vedere arhitectural, distingem între următoarele layer-e majore, conform ilustrației din Figura 7: Orchestrare (Orchestration), Servire model (Model Serving), Acces la date (Data Access) și Versionare (Versioning).

Layer-ul Orchestrare conține trei componente principale: Orchestrator – modulul de orchestrare a întregii platforme, Prediction Web Service – un serviciu web orientat către client, și Orchestration Storage – sistemul central de stocare și configurare al platformei.

Orchestration Storage este folosit pentru configurare, sincronizare, checkpointing și păstrare a evidenței. O singură instanță rulează per mediu; serviciul este accesibil doar în layer-ul Orchestrare. Toate configurațiile de inferență a modelelor (URL-ul serviciului, dimensiunea lotului de inferență, timeout-ul serviciului, etc.), candidații de inferență (informații de catalogare, precum și datele meta), mesajele de progres, notificările de eroare și rezultatele inferenței sunt persistate în aceasta baza de date.

Prediction Web Service găzduiește interfețele de programare a aplicațiilor (API-urile) cu care clientul interacționează atunci când emite cereri de inferență și monitorizează progresul acestora.

Aționând ca unitatea centrală de orchestrare pentru întreg mediul, Orchestrator-ul este, de asemenea, responsabil de monitorizarea și gestionarea infrastructurii și serviciilor. Front-end-ul integrat al Orchestrator oferă vizualizări pentru diferite niveluri de monitorizare, depanare și interpretare a datelor și modelelor de învățare automată, inclusiv: Celery Flower, Portainer, Kibana, panoul "Modele implementate", panoul "Ultimele job-uri de inferență", Redash și pgweb.

Nivelul Servire model găzduiește toate modelele implementate. Fiecare serviciu rulează un singur model încapsulat într-un server HTTP. Toate serviciile acestui nivel moștenesc

funcționalitatea cadru și API-ul de la o „imagine de bază” Docker. Imaginea de bază face abstracție de use-case-ul modelului și framework-ul de învățare automată utilizat. Imaginile derivate injectează dependențe și runtime-uri personalizate pentru o flexibilitate maximă.

Imaginea de bază oferă scheletul pentru (re-)antrenarea modelului și servirea predicțiilor. Sunt furnizate prototipuri pentru filtre pre-inferența (orice candidat care nu trece aceste verificări va fi exclus din procesul de inferența), precum și pentru protecții post-predicție (pentru cazurile în care rezultatul inferenței este „pe baza celor mai bune eforturi” și ar trebui interpretat cu prudență de către solicitant), care pot fi suprascrise sau supraîncărcate în modelele derivate.

Layer-ul Acces la data facilitează ingestia datelor în platformă. Conectori individuali – numiți DAC – pot fi codificați în diferite limbaje de programare sau scripting și configurați în funcție de mediul de stocare pentru care sunt destinați. API-urile variază în funcție de scenariul de utilizare, dar includ întotdeauna un punct final pentru verificarea stării de funcționare.

Nivelul Versionare este un singleton disponibil în toate mediile. Întreaga dezvoltare, implementarea serviciilor și orchestrarea acestora depind de disponibilitatea acestui nivel.

Versionarea se aplică pe mai multe sub-niveluri pentru toate componentele, începând de la codul sursă, peste parametrii și artefactele modelelor, până la imaginile Docker. Acestea sunt stocate în diverse sisteme, inclusiv: Git, Docker Registry și AWS S3. Am folosit un sistem de identificare universal unic pentru a cataloga și modela relațiile dintre artefactele stocate.

Pentru a automatiza majoritatea etapelor de dezvoltare și roll-out a modelelor, am implementat o interfață de linie de comandă (CLI) dedicată, cu contexte personalizabile. CLI-ul este folosit pentru coding, construirea imaginilor Docker, testarea codului și imaginilor, antrenarea și validarea modelelor, implementarea prin procese Jenkins de Continuous Deployment și managementul ciclului de viață. Comenzile CLI sunt disponibile oriunde în copia locală a codului sursă.

CLI-ul nostru facilitează, dar nu impune eforturi regulate de integrare și testare. Pentru a asigura stabilitatea codului după fiecare commit, am implementat un sistem personalizat CI/CD. Am configurat fluxuri de lucru Jenkins pentru a compila, testa și implementa codul, imaginile și serviciile. Fluxurile de lucru Jenkins valorifică funcționalitatea CLI ori de câte ori este posibil – de exemplu prin intermediul comenzilor `build`, `test` și `up`, iar la rândul său, CLI-ul valorifică fluxul de implementare Jenkins pentru rularea serviciilor în configurații distribuite [20].

Pentru a maximiza uptime-ul și funcționarea eficientă a platformei și pentru a superviza performanța modelelor de învățare automată, am proiectat o soluție de monitorizare cu mai multe niveluri. În comparație cu sistemele software tradiționale, sistemele de învățare automată adaugă o serie de considerații specifice [17]. Spre deosebire de sistemele software tradiționale, comportamentul unui sistem de învățare automată este guvernat nu numai de regulile specificate în cod, ci și de comportamentul modelului învățat din date. Noutatea abordării prezentate constă în modul în care componente individuale sunt selectate, stratificate, integrate și prezentate utilizatorilor. [19] *Supraveghez, captez și afișez*:

- Disponibilitatea hardware-ului și nivelul de utilizare a acestuia, ca parte a System Monitoring prin intermediul recoltorilor de metrice, Prometheus și Grafana,
- Disponibilitatea serviciilor individuale și starea acestora, ca parte a System Monitoring prin intermediul recoltorilor de metrice, Prometheus și Grafana,
- Performanța modelelor de învățare automată (monitorizarea candidaților și filtrelor pre-inferența, rezultatelor și protecțiilor post-predicție, buclele de feedback) ca o combinație de Data Monitoring și Prediction Monitoring prin intermediul Orchestration Storage, Redash și pgweb,

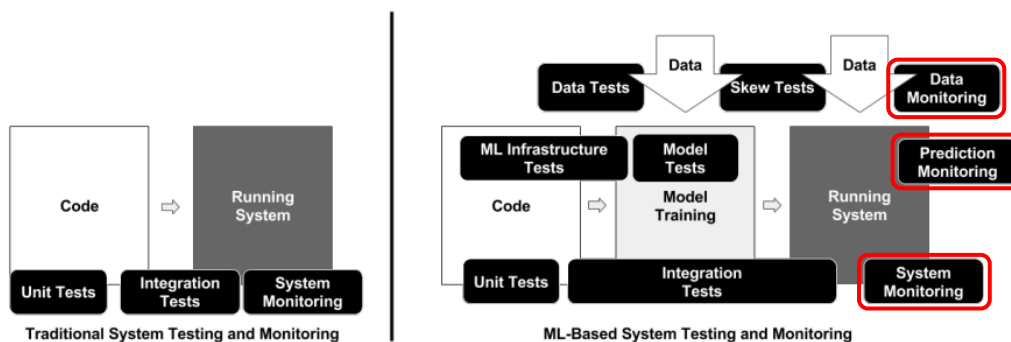


Figura 8. Monitorizarea codului tradițional (stânga) și a sistemelor de învățare automată (dreapta) [15]

- Jurnal de evenimente de la toate modelele și serviciile (auditabilitate și depanare) ca o combinație a tuturor celor trei zone de monitorizare evidențiate în Figura 8 prin intermediul stivei ELK (Elasticsearch, Logstash, Kibana).

Soluția proprie de monitorizare oferă astfel trei panouri de control, punând la dispoziție nivele diferite de detaliu pentru audiențe diferite. Grafana oferă stakeholder-ilor și personalului de mentenanță un panou de bord high-level pentru sănătatea aplicației. Folosind toate datele istorice de inferență, Redash poate fi utilizat pentru afișarea de statistici atât la nivel înalt, cât și detaliate, destinatarii fiind atât părțile interesate, cât și utilizatorii experți. Redash permite astfel urmărirea comportamentului fiecărui model de învățare automată în raport cu obiectivele sale de performanță atât pe moment cât și în timp. În cele din urmă, tabloul de control Kibana facilitează corelarea temporală a tuturor evenimentelor înregistrate în sistem, accelerând considerabil analiza situațiilor de eroare și activitățile de depanare, eliminând nevoia agregării manuale de informații din diferite surse.

Platforma proprie prezentată țintește să facă procesul de învățare automată perfect integrat, scalabil și consistent. Pentru use-case-ul Process & Pipeline Services:

- Captarea înregistrărilor de proveniență a datelor pentru toate experimentele și versionarea tuturor modelelor antrenate au asigurat reproductibilitatea și auditabilitatea în toate etapele ciclului de viață al modelului,
- Sistemul a oferit implementare în producție cu aproape zero efort: operațiunile de preprocesare a datelor au fost identice cu cele folosite în timpul antrenamentului, abordarea dockerizată a asigurat echivalența între runtime-ului de producție și cel folosit în dezvoltare, versionarea a asigurat că toate modelele antrenate și serviciile auxiliare erau disponibile imediat pentru operaționalizare,
- Platforma s-a dovedit scalabilă la seturi de date arbitrare de mari – implementată pe un EC2 AWS g3.4xlarge, modelul a furnizat predicții în mod batch pentru seturi de date depășind milioane de puncte de date pe set și TB în stocare necomprimată, utilizând până la 64 de conectori DAC în paralel, fără probleme de stabilitate sau epuizare a memoriei,
- Pentru monitorizarea modelului, am folosit Orchestration Storage și front-end-ul Orchestrator. Am creat grafice și statistici pentru rata de automatizare (numărul de predicții publicate față de totalul candidaților), distribuții ale scorului de inferență, filtrele pre-inferență și protecțiile post-predicție. [13]

Valorificând soluția proprie de stocare NoSQL HBase și platforma proprie pentru gestionarea întregului ciclu de viață al modelelor de învățare automată, am construit **soluția inovatoare pentru analiza datelor multidimensionale înregistrate de PIG-ul UltraScan**

CDP prin identificarea și clasificarea ROI-urilor non-defect. Aplicația a fost antrenată pe mai mult de 100 de inspecții efectuate de UltraScan CDP, agregând datele 4D înregistrate cu informațiile meta la nivel de inspecție pentru un set de peste 5 milioane de ROI-uri.

Modelul rezultat – un ansamblu de rețele neurale convoluționale (CNN) – s-a dovedit capabil să clasifice ROI-uri cu un grad extrem de ridicat de fidelitate. Figura 9 prezintă diagramele de distribuție a scorurilor pentru 5 inspecții eșantion. Diagramele prezentate sunt reprezentative pentru întregul set. Predicțiile pentru ROI-urile defect sunt reprezentate în albastru, iar cele pentru cele non-defect în magenta. ROI-ul defect cu scorul de predicție cel mai mic este evidențiat cu o săgeată albastră. Chiar dacă cele două distribuții prezintă o suprapunere semnificativă în zona scorurilor superioare, rezultatele indică faptul că modelul permite izolarea ROI-urilor non-defect pentru automatizare.

Prin clasificarea automată a zonelor de interes ca nepericuloase, am obținut o reducere de peste 10% a timpului total necesar pentru analiza seturilor de date UltraScan CDP. Măsurătorile au arătat că modelul automatizează de 3.5 ori mai multe ROI decât evaluarea manuală la rata maximă per unitate timp. Eliminarea muncii „de rutină” a permis experților să se

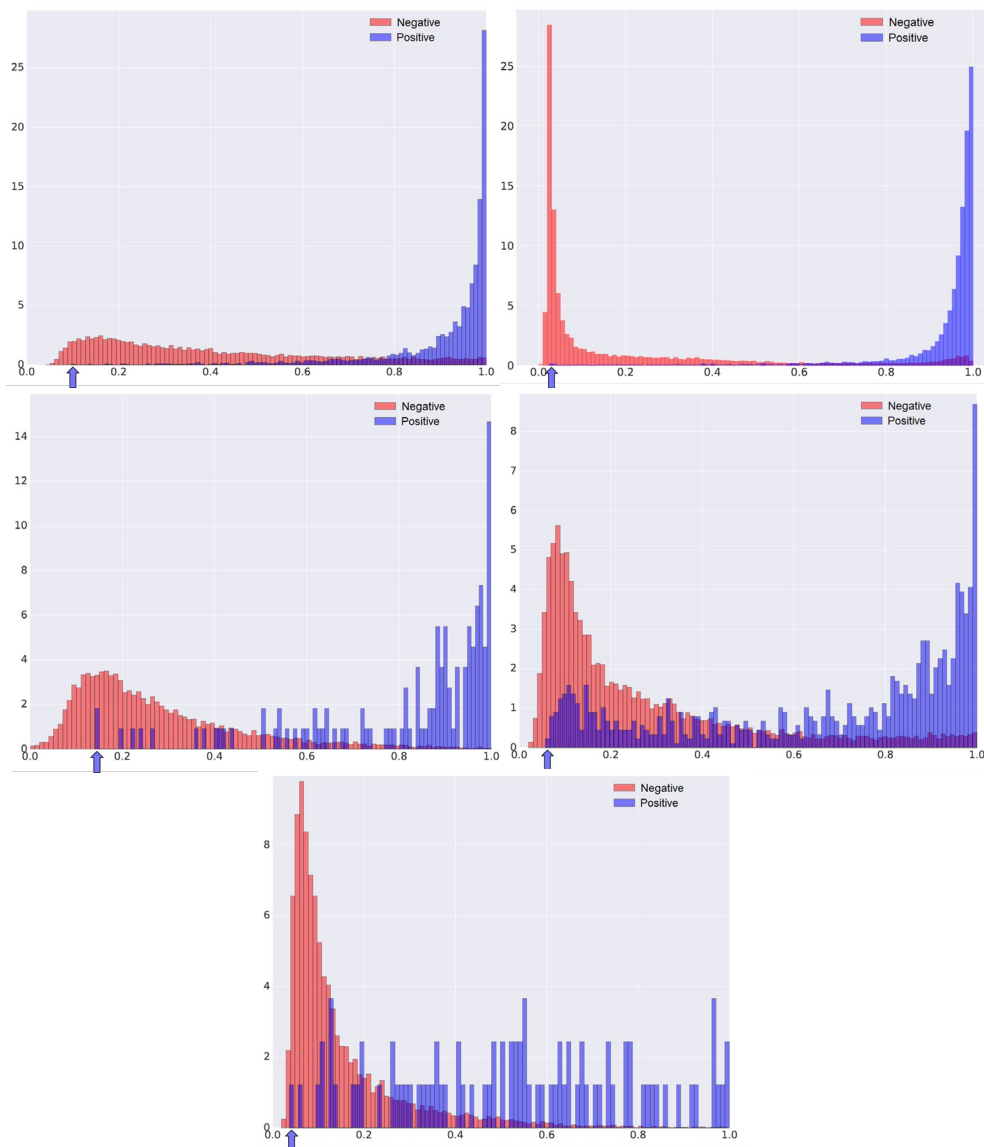


Figura 9. Distribuția scorurilor de inferență per inspecție

concentreze pe deciziile "de înaltă calificare" sau "dificile".

Pentru lansarea în producție, modelul a fost complementat cu o varietate de măsuri de protecție pentru a asigura detectarea derapajului datelor/conceptului și degradarea generală a performanței modelului înainte ca acestea să aibă un impact negativ. Am implementat proceduri de rollback, folosind Redash (Figura 10) pentru a monitoriza vizual performanța modelului și a activa bucla de feedback. Rezultatele prezentate în Figura 10 sunt pentru o inspecție. Jumătatea superioară a figurii prezintă procentul de automatizare per criteriu (partea stângă) și distribuția scorului de inferență (partea dreaptă). Personalul calificat poate evalua aceste informații pentru rate anormale de automatizare – prea mari sau prea scăzute, precum și segregarea populației.

Partea inferioară din Figura 10 prezintă un rezumat al filtrelor pre-inferență, urmat de un rezumat al protecțiilor post-predicție. Orice verificare pre-predicție care eșuează este imediat notată în protocolul asociat candidatului și candidatul este exclus din inferență. Măsurile de protecție post-predicție sunt executate după inferență. Scopul principal este de a notifica solicitantul dacă vreuna dintre predicții trebuie tratată ca "cel mai bun efort". În astfel de scenarii, rezultatul ar putea fi încă relevant pentru scenariile de augmentare sau QA, dar este eliminat pentru aplicațiile de automatizare critice pentru siguranță. Orice eșec post-predicție este notat în protocolul candidatului. Modelul prezentat aplică aceleași filtre pre-predicție și

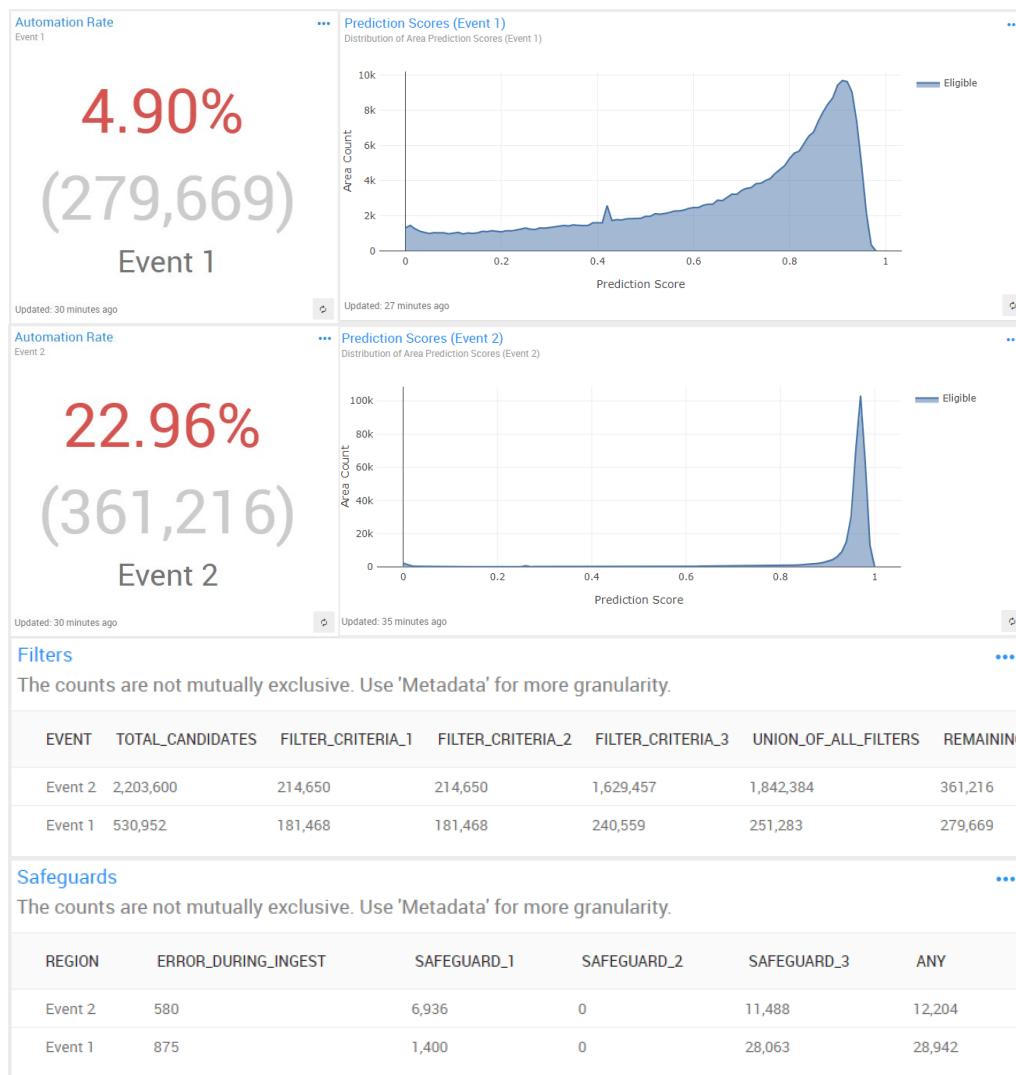


Figura 10. Panoul Redash

măsuri de siguranță post-inferență pentru ambele criterii.

Rezumând contribuțiile personale ale tezei, acestea includ, dar nu se limitează la:

- Un studiu comparativ al tehnologiilor NoSQL existente în contextul cerințelor de stocare și tranzacționare a datelor ale Process & Pipeline Services,
- O soluție de stocare NoSQL personalizată pentru datele UltraScan CDP 4D al Process & Pipeline Services,
- Un mecanism de ingestie accelerată aplicabil tuturor instanțelor HBase optimizate pentru operațiuni de scanare,
- O platformă nouă pentru gestionarea întregului ciclu de viață al modelelor de învățare automată, cu potențialul de a revoluționa modul în care modelele sunt antrenate și puse în producție. Platforma se concentrează pe automatizare, repetabilitate, reproductibilitate și observabilitate – aspecte discutate în această lucrare,
- O soluție avansată pentru analiza automată a datelor de detectare a fisurilor în conducte, înregistrate de UltraScan CDP. Modelul complex de învățare automată dezvoltat cu ajutorul componentelor și sistemelor introduse, generează rate semnificative de automatizare într-un mediu critic din punct de vedere al siguranței, permițând astfel experților să-și concentreze eforturile acolo unde contează cel mai mult.

În lucrare propunem elemente avansate de strictă noutate nemaiîntâlnite în literatura de specialitate după cunoștințele autorului.

BIBLIOGRAFIE

- [1] V. K. Varma, R. W. Tucker, Jr., A. P. Albright, "EMAT-Based Inspection of Natural Gas Pipelines for Stress Corrosion Cracks," Oak Ridge National Laboratory, Oak Ridge, Tennessee, USA, 2004.
- [2] "Crack capabilities overview." Process & Pipeline Services. Accessed April 2021. [Online]. Available: https://www.bakerhughes.com/sites/bakerhughes/files/2020-07/19004_BH_PPS_ILI_US_BRO_1912%20%28CRACK%20CAPAB%29.pdf
- [3] A. Barbian, M. Beller, "In-Line Inspection of High Pressure Transmission Pipelines: State-of-the-Art and Future Trends," *18th World Conference on Nondestructive Testing*, April 2012, Durban, South Africa.
- [4] W. Krieg, "A Novel EMAT Crack Detection and Coating Disbondment (RoCD2) ILI Technology," *Pipeline Technology Conference*, 2007, Hannover, Germany.
- [5] N. V. Chawla, "Data Mining for Imbalanced Datasets: An Overview," in Maimon O., Rokach L. (eds), *Data Mining and Knowledge Discovery Handbook*, Springer, Boston, MA, 2009.
- [6] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, A. Geiger, "Sparsity Invariant CNNs," *2017 International Conference on 3D Vision (3DV)*, Qingdao, China, 2017, pp. 11-20, doi: 10.1109/3DV.2017.00012.
- [7] N. Leavitt, "Will NoSQL Databases Live Up to Their Promise?," *Computer*, vol. 43, no. 2, pp. 12-14, Feb. 2010, doi: 10.1109/MC.2010.58.
- [8] "RDBMS dominate the database market, but NoSQL systems are catching up." DB-Engines.com. Accessed 25 November 2019. [Online]. Available: https://db-engines.com/en/blog_post/23
- [9] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, R. E. Gruber, "Bigtable: A Distributed Storage System for Structured Data," Google Inc., 2006. [Online]. Available: <https://research.google.com/archive/bigtable-osdi06.pdf>
- [10] D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, M. Young, D. Dennison, "Hidden Technical Debt in Machine Learning Systems," *NIPS*, 2015, pp. 2494-2502.
- [11] "UT Inline-Inspection Overview," Process & Pipeline Services, unpublished.
- [12] A.-I. Argesanu, G.-D. Andreescu, "Accelerating Data Ingress for Range-Scan Optimized HBase Instances," *2021 IEEE 15th International Symposium on Applied Computational Intelligence and Informatics (SACI)*, Timisoara, Romania, 2021, pp. 341-344, doi: 10.1109/SACI51354.2021.9465633.
- [13] A.-I. Argesanu, G.-D. Andreescu, "A Platform to Manage the End-to-End Lifecycle of Batch-Prediction Machine Learning Models," *2021 IEEE 15th International Symposium on Applied Computational Intelligence and Informatics (SACI)*, Timisoara, Romania, 2021, pp. 329-334, doi: 10.1109/SACI51354.2021.9465588.
- [14] L. Bellatreche, G. Chernishev, A. Corral, S. Ouchani, J. Vain, "Advances in Model and Data Engineering in the Digitalization Era," *MEDI 2021 International Workshops: DETECT, SIAS, CSMML, BIOC, HEDA*, Tallinn, Estonia, June 2021, doi: 10.1007/978-3-030-87657-9.
- [15] E. Breck, S. Cai, E. Nielsen, M. Salib, D. Sculley, "The ML test score: A rubric for ML production readiness and technical debt reduction," *2017 IEEE International Conference on Big Data (Big Data)*, Boston, MA, USA, 2017, pp. 1123-1132, doi: 10.1109/BigData.2017.8258038.
- [16] K. Simonyan, A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *3rd International Conference on Learning Representations (ICLR) 2015*, May 2015.

- [17] E. d. S. Nascimento, I. Ahmed, E. Oliveira, M. P. Palheta, I. Steinmacher, T. Conte, "Understanding Development Process of Machine Learning Systems: Challenges and Solutions," *2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, Porto de Galinhas, Brazil, 2019, pp. 1-6, doi: 10.1109/ESEM.2019.8870157.
- [18] S. Shankar, R. Garcia, J.M. Hellerstein, A.G. Parameswaran, "Operationalizing machine learning: An interview study," arXiv preprint arXiv:2209.09125, 2022.
- [19] A.-I. Argesanu, G.-D. Andreescu, "From Data to Decisions: The Importance of Monitoring ML Systems in Industrial Settings," *Acta Technica Napocensis - Series: Applied Mathematics, Mechanics, and Engineering*, vol. 66, no. 3, 2023. [Online]. Available: <https://atna-mam.utcluj.ro/index.php/Acta/article/view/2188>
- [20] A.-I. Argesanu, G.-D. Andreescu, "Streamlining Machine Learning Workflows in Industrial Applications with CLI's and CI/CD Pipelines," *Acta Technica Napocensis - Series: Applied Mathematics, Mechanics, and Engineering*, vol. 66, no. 3, 2023. [Online]. Available: <https://atna-mam.utcluj.ro/index.php/Acta/article/view/2189>