HOLISUN

MUSHNOMICS - Proiect de cercetare

RAPORT ȘTIINȚIFIC ȘI TEHNIC 2022

Referință 5929/25.11.2022

Manager de proiect: ***Rudolf Erdei***

Istoricul versiunilor

Versiune	Autor	Modificări
1.0	Rudolf Erdei	Versiunea inițială

Cuprins

Parteneri

Figura 1: Logo-ul proiectului MUSHNOMICS

(a) Holisun SRL(Holisun, Romania) Coordinator

(b) Pilze-Nagy Ltd (PILZE, Hungary)

(c) Department of Plant and Environmental Sciences, University of Copenhagen (UCPH, Denmark)

(d) University College Dublin (UCD, Ireland)

Figura 2: Partenerii proiectului *MUSHNOMICS*

Sumar

Acest raport prezintă activitatea Holisun în anul al doilea din cadrul proiectului *MUSHNOMICS* . Activitatea este împărțită pe diferite capitole, discutându-se metodologiile de lucru utilizate precum și inovațiile ce au fost introduse în componentele dezvoltate. Se prezintă rolul Holisun în cadrul proiectului precum și rezultatele obținute. Structura și conținutul capitolelor este precum urmează:

În Capitolul 1 conține informații despre proiect și despre conținutul prezentului document. Capitolul 2 vorbește despre obiectivele ce au stat la baza activităților din perioada de referință. În Capitolul 3 se arată pe scurt care sunt activitățile planificate pentru perioada de raportare, iar Capitolul 4 conține activitățile efectuate în perioada de referință.

Capitolul 5 prezintă realizările tehnice, implementarea ce a avut loc în cadrul proiectului. Capitolul 6 prezintă livrabilele ce au fost elaborate în cadrul proiectului și artefactele rezultate. În Capitolul 7 se prezintă activitățile de diseminare și exploatare în cadrul cărora rezultatele din cadrul proiectului au fost prezentate. În final, în Capitolul 8 se discută concluziile documentului privind desfășurarea proiectului.

1 Introducere

MUSHNOMICS își propune să dezvolte o platformă integrată pentru eficientizarea producției de ciuperci, precum și eficientizarea întregului lanț de producție și aprovizionare, eficientă din punct de vedere al costurilor, capabilă să monitorizeze sănătatea culturilor. În general, proiectul aspiră să culmineze cu producerea unei platforme radicale care va fi o paradigmă schimbătoare a modului în care inovația tehnologică bazată pe inteligența artificială poate deveni un instrument accesibil, accesibil tuturor și ușor de utilizat, aplicabil tuturor formelor de agricultură în medii controlate.

Prezentul raport oferă o imagine de ansamblu asupra cadrului operațional și a designului aplicației *MUSHNOMICS* , responsabilă cu interacțiunea care are loc între platformă și utilizatorul final. Aplicația va extinde și completa funcționalitățile Platformei digitale *MUSHNOMICS* , făcând-o o soluție completă pentru utilizarea acesteia.

Raportul descrie pe scurt proiectarea și implementarea platformei de date *MUSHNOMICS* cu cele două componente ale ei: sistemul de ingestie și prelucrare a datelor, precum și interfața cu utilizatorul. În cele din urmă, raportul clarifică amplasarea arhitecturală a componentelor digitale care deservesc soluția generală *MUSHNOMICS* .

2 Obiective planificate în perioada raportată

În perioada de raportare, Holisun a efectuat activități de cercetare, inovare și implementare, necesare pentru buna desfășurare a proiectului *MUSHNOMICS* . Având în vedere rolul tehnic al Holisun, implementarea a ocupat un rol central în activitatea companiei în cadrul proiectului, pe toată durata desfășurării acestuia.

Scopul Holisun în cadrul proiectului a fost, printre altele, și realizarea interfeței grafice cu utilizatorul, interfață ce va fi folosită de către toți utilizatorii sistemului, indiferent de gradul lor de înțelegere. Astfel, obiectivele planificate pentru perioada de raportare, sunt:

- O1 - proiectarea arhitecturii aplicației;
- O2 - proiectarea entităților din cadrul sistemului, astfel încât să poată fi înțelese cât mai ușor de către oricare utilizator și performanța sistemului să fie la un nivel înalt;
- O3 - formalizarea cerințelor sistemului, astfel încât acesta să fie cât mai accesibil oricărei clase de utilizatori;
- O4 - implementarea interfeței utilizatorului, ținând cont de toate cerințele elicitate, atât funcționale cât și non-funcționale;
- O5 - implementarea componentei de import date și prelucrarea acestora în vederea generării modelelor de învățare automată;
- O6 - testarea platformei cu ajutorul uneltelor disponibile.
- O7 - monitorizarea progresului și coordonarea echipei de proiect

3 Activități planificate

În perioada 01.12.2021 - 01.12.2022 au fost planificate următoarele activități:

- Cercetare pentru stabilirea nevoilor și cerințelor platformei *MUSHNOMICS* ;
- Cercetarea și proiectarea arhitecturii sistemului *MUSHNOMICS* ;
- Implementarea componentei de import date ce va fi folosită de către clienții platformei;
- Implementarea funcționalităților de conexiune la date pentru modulul de interfață cu utilizatorul;
- Proiectarea testării componentelor dezvoltate;
- Întâlniri trimestriale de progres.

4 Activități efectuate

În perioada 01.12.2021 - 01.12.2022 au fost efectuate următoarele activități:

- Cercetare pentru stabilirea nevoilor și cerințelor platformei *MUSHNOMICS* ;
- Cercetarea și proiectarea arhitecturii sistemului *MUSHNOMICS* ;
- Implementarea componentei de import date ce va fi folosită de către clienții platformei;
- Implementarea funcționalităților de conexiune la date pentru modulul de interfață cu utilizatorul;
- Proiectarea testării componentelor dezvoltate;
- 4 întâlniri trimestriale de progres, în datele: 24.02.2022, 17.05.2022, 25.07.2022, 13.09.2022.

4.1 Devieri de la planificare

În perioada raportata nu au fost devieri de la planificare, sub nici un aspect.

5 Realizări tehnice

5.1 Proiectarea infrastructurii de date

În prezent, companiile sunt din ce în ce mai dependente de analiza datelor. Aceasta presupune urmărirea anumitor parametri, înțelegerea corelației dintre acestea, analiza statisticilor referitoare la datele rezultate precum și generarea de predicții. Predicțiile sunt importante în contextul diferitelor optimizări posibile pentru a îmbunătăți calitatea și cantitatea de produse pe care compania dorește să le comercializeze, precum și pentru reducerea costurilor de producție.

Aceste aspecte, odată rezolvate, duc la anumite cunoștințe derivate din procesul de predicție, procesul de analiza și corelare a datelor și parametrilor. Aceste cunoștințe pot fi făcute publice de către companie, însă în general acestea sunt private, considerate a fi secrete de companie. De asemenea, datele din care rezultă aceste cunoștințe pot fi secrete de companie, întrucât datele pot releva parametrii de funcționare ale anumitor instalații, ce ar ajuta concurența, limitând astfel veniturile companiei.

Datorită situațiilor și aspectelor discutate, infrastructura de date *MUSHNOMICS* va avea disponibile două moduri de utilizare, în funcție de opțiunea utilizatorului (care în situația noastră este compania-client). Una dintre opțiuni va fi centrată pe clienții care nu dispun încă de volum de date care ar afecta secretele de firmă, iar a doua opțiune va fi pentru companiile ce doresc să-și păstreze datele complet private. Astfel, opțiunile sunt:

1. **Centralizarea tuturor datelor într-o platformă cloud** și generarea de modele pe baza datelor;
2. **Generarea locală de modele** și centralizarea modelelor parțiale (numită și **Învățare Federativă**).

Întrucât avantajele și dezavantajele sunt destul de balansate, iar modalitatea ideală de a lucra cu fiecare companie în parte poate fi diferită, am ales proiectarea infrastructurii astfel încât să poate funcționa în ambele moduri, atât centralizat cât și federativ.

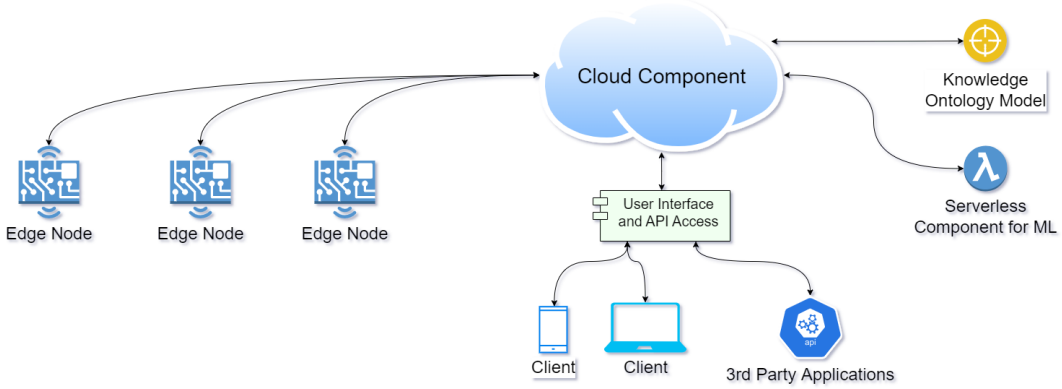

Figura 3: Arhitectura de nivel înalt pentru platforma *MUSHNOMICS*

În Figura 3 se prezintă propunerea pentru arhitectura de nivel înalt a platformei *MUSHNOMICS* , o arhitectură scalabilă ce va putea deservi un potențial de mii de clienți. Fiecare client își va putea defini propriul proiect, cu propriile modalități de ingerare a datelor, astfel încât platforma va oferi flexibilitatea necesară lucrului cu arhitecturi de orice fel.

Pentru proiectarea acestei arhitecturi, au fost luate în calcul următoarele cerințe funcționale și non-funcționale:

- Clientul își va putea conecta infrastructura proprie de date, fie că e vorba de un sistem local (învățare federativă), sau doar de senzori (învățare centralizată);
- Componenta Cloud va putea acomoda atât învățarea centralizată cât și cea federativă;
- Cunoștințele obținute vor putea fi formalizate de către utilizatori într-un model ontologic, disponibil prin componenta UI;
- Procesele de învățare automată vor fi disponibile prin funcții lambda (componente *serverless*) pentru reducerea costurilor asociate platformelor cloud;
- Clienții vor avea acces la o interfață grafică interactivă, atât prin intermediul browserului, cât și prin intermediul dispozitivelor mobile, pentru urmărirea parametrilor și obținerea diferitelor estimări;
- Clienții vor putea accesa platforma și cu ajutorul aplicațiilor terțe, prin intermediul API-ului pus la dispoziție;
- Clienții vor putea pune la dispoziție datele / modelele de învățare automată, pentru a putea fi analizate de către cercetători din domeniu și a contribui astfel la progres științific.

Deși este o arhitectură de nivel înalt, se pot deja observa avantajele utilizării ei, prin flexibilitatea și extensibilitatea ce le oferă sistemului rezultat. Această platformă, chiar aplicată altor domenii diferite de cea de față, va oferi cercetătorilor și companiilor o posibilitate de colaborare cu beneficii mutuale, scăzând astfel discrepanța dintre cercetarea fundamentală și aplicarea ei în producție.

5.2 Proiectarea arhitecturii pentru modulul de import și prelucrare a datelor

Componenta Cloud este punctul central în arhitectura *MUSHNOMICS* deoarece este centrul operațional al acestuia. Componenta software instalată are multiple roluri, printre care:

- Definirea entităților de tip *Proiect* sau *Aplicație* folosite de utilizatori;
- Definirea setărilor per proiect/aplicație, ce vor controla modul de funcționare și utilizare particularizată a platformei;
- Modalitatea de import a datelor (pentru modul respectiv de lucru) precum și locația acestora;
- Modalitatea de generare a modelelor de învățare automată (tot în funcție de setările de proiect);

- Modalitatea de interacțiune cu modelele de învățare automată generate;
- Controlul funcționării nodurilor, în contextul nevoilor și opțiunilor exprimate de utilizator.

5.2.1 Cerințe funcționale și non-funcționale ale arhitecturii

În cadrul proiectului *MUSHNOMICS* , cerințele funcționale și non-funcționale sunt obținute după sesiuni de brainstorming în care au fost incluși și potențiali beneficiari ai proiectului, precum și alte părți interesate. După generarea inițială a ideii, structurarea cerințelor s-a făcut prin utilizarea metodologiei *MoSCoW* [1], care prioritizează funcțiile cu cele mai multe beneficii pentru rezultatul final și, de asemenea, folosind metoda *ATAM* [6]. Evaluarea așteptărilor utilizatorilor și cererea pieței pentru o soluție centrată pe confidențialitate au fost de asemenea luate în considerație la proiectarea acestor cerințe.

Cerințele funcționale sunt formulate ca funcționalități de platformă de nivel înalt. Fiecare element a fost apoi împărțit în cerințe mai specifice și atomice necesare pentru fazele de dezvoltare, testare și validare. Dintre acestea, cele mai importante sunt:

- **FR1** - Abilitatea de a utiliza mai multe metodologii *Distributed Learning*(DL), inclusiv *Învățare Federativă*, dar și a oferi suport pentru metodele clasice (centralizate) de Machine Learning (ML);
- **FR2** - Capacitatea sistemului de a oferi atât suport de cercetare [2], cât și metode automate de calitate [12];
- **FR3** - Capacitatea sistemului de a găzdui mai mulți producători agricoli într-un mod sigur, protejat, și cu date private, dar și de a permite crearea consorțiilor de producători agricoli;
- **FR4** - Capacitatea sistemului de a-și furniza serviciile într-un mod centrat pe confidențialitate;
- **FR5** - Abilitatea de a reține cunoștințele descoperite, per proiect;
- **FR6** - Abilitatea de a conecta la sistem aplicații/integrări client de 3^{rd}.

Un extras din **Cerințele non-funcționale**, ca aspecte de performanță și securitate de care are nevoie platforma pentru a oferi servicii de calitate, sunt:

Conform Figurii 3, sistemul este compus din cel puțin trei componente principale, componente dezvoltate într-un mod de sine stătător și flexibil, astfel încât să se poată utiliza și în alte sisteme similare. Rolurile și responsabilitățile celor trei mari componente ale sistemului sunt:

- **NFR1** - *Securitate*, *Scalabilitate* și *Stabilitatea* sistemului rezultat;
- **NFR2** - *Reziliența platformei*, ca abilitate de a *gestiona și recupera mai multe clase de eșec* din cadrul sistemului;
- **NFR3** - *Cerințe hardware cu specificații reduse* pentru Componenta Cloud (folosind Componentele Serverless);
- **NFR4** - *Performanta ridicata* și *API-uri flexibile* pentru integrări de 3^{rd};
- **NFR5** - *Portabilitate/Accesibilitate*, cu posibilitatea ca platforma să fie accesată din diverse sisteme de operare, inclusiv mobil;
- **NFR6** - *Flexibilitate*, deoarece platforma ar trebui utilizată atât în scopuri științifice, cât și în scopuri de producție.

5.2.2 Arhitectura sistemului de învățare distribuită

Metodologiile DL au un avantaj uriaș deoarece acestea utilizează nodurile (Edge-ul) pentru generarea unui model parțial. Dar, pentru ca *Model Federativ* să poată fi obținut, modelele în sine trebuie să fie *compatibile* [7]. Modul obișnuit de federare a modelelor este prin realizarea mediei valorilor parametrilor pe toate modelele parțiale.

Caracteristicile modelului sunt trimise ca metadate, astfel încât Componenta Cloud poate estima un factor de importanță pentru acesta, factor care va fi tradus în ponderea modelului în medie. Aceste operațiuni asigură robustețea modelului și rezistența inițială împotriva problemelor care pot apărea, inclusiv eșecurile bizantine ale nodului de margine [17].

Compatibilitatea modelului se obține în doi pași:

1. **Caracteristicile construcției modelului** (cum ar fi numărul de straturi ascunse și/sau numărul de tensori pe strat, pentru rețelele neuronale) și un model inițial sunt obținute în partea de *cercetare* a metodologiei de lucru. Aceasta oferă cele mai bune setări pentru modelul care urmează să fie construit. Aceste caracteristici, care sunt unice pe proiect, sunt apoi trimise la Nodurile Edge;
2. Nodurile Edge utilizează caracteristicile primite și datele locale, pentru a genera un **model parțial** care va fi încărcat în Componenta Cloud. Folosind aceleași setări, media modelelor se poate face într-un mod mai simplu.

În cazul Învățării Federative, Componenta Cloud nu stochează permanent modelele parțiale (deoarece acest lucru ar prezenta unele riscuri de confidențialitate), mai degrabă folosește modelele parțiale pentru a actualiza modelul global și apoi renunță la modelul parțial utilizat.

Arhitectura de nivel înalt este prezentată în Figura 3. Putem observa multiplele *Edge Nodes*, care furnizează intrările necesare de proiect către *Componenta Cloud* și primesc, de asemenea, parametrii modelului și alte setări ale platformei (cum ar fi frecvența de citire a senzorului). Componenta *Cloud* centralizează fie datele complete (pentru metodologiile clasice ML), fie modelele parțiale rezultate, pentru a oferi analize, stocare de date/model, păstrare a cunoștințelor și o conductă avansată de *crearea modelelor* care poate fie lucrează într-o manieră clasică, sau într-o manieră descentralizată (federată). Accesul la platformă poate fi oferit fie prin intermediul *Web UI*, fie printr-un *API*.

Edge Nodes sunt responsabile pentru dispozitivele locale de colectare a datelor, centralizarea și preprocesarea datelor de la senzori. În funcție de metodologie, datele vor fi utilizate fie pentru generarea/îmbunătățirea unui model local, fie pentru încărcarea acestuia în *Componenta Cloud*, prin intermediul `Modulul de export de date`. Dacă este necesar, *Edge Nodes* poate avea o interfață de administrare locală, care va prezenta și statistici locale și *Node Health*.

Componenta **Cloud Platform** este responsabilă pentru agregarea fie a datelor, fie a modelelor parțiale, generând schema inițială a modelului, oferind suport pentru *Edge Nodes Management*, *System Overview* și *Knowledge Management*.

Componenta **Platform UI** este responsabilă de gestionarea comunicațiilor cu clienții, prin intermediul `Web UI Interface`, astfel încât modulele din cadrul acestuoa sunt concentrate exclusiv pe această sarcină.

Arhitectura detaliată este prezentată în Figura 4, unde putem distinge componentele principale responsabile de fiecare dintre modurile de lucru prezentate anterior (vezi Secțiunea 5.2.2).

Această arhitectură se regăsește în articolul publicat în cadrul proiectului: *Security Centric Scalable Architecture for Distributed Learning and Knowledge Preservation* [5]

5.2.3 Arhitectura componentelor Cloud

Componenta Cloud este compusă din mai multe subsisteme care oferă servicii importante pentru platformă. Aceste subsisteme au roluri foarte specializate și specifice, fiecare fiind compus din unul sau mai multe module care vor coopera pentru a implementa pe deplin cerințele desemnate. O scurtă descriere a subsistemelor existente:

- **Edge Communication Manager** - implică toate modulele care primesc sau trimit informații către/de la componentele edge și este dirijat de *Modulul de control al accesului*. Modulele oferă, de asemenea, mecanismele de criptare/decriptare necesare pe care platforma le va folosi;
- **Multi-Paradigm Data Ingestion Pipeline** - gestionează *importul* de date/modele parțiale în platformă, verificând *validitatea datelor* și calculând *calitatea modelului* (vezi Secțiunea 8);
- **Persistența datelor** - evaluează dacă datele primite pot/trebuie să fie persistente și acționează în baza acestei decizii. De asemenea, transmite modelele primite către metodologia Învățării Federative, care va declanșa *Actualizarea generală a modelului*;
- **Knowledge Preservation** - cunoștințele adunate vor îmbunătăți/actualiza cunoștințele deja existente (modelul

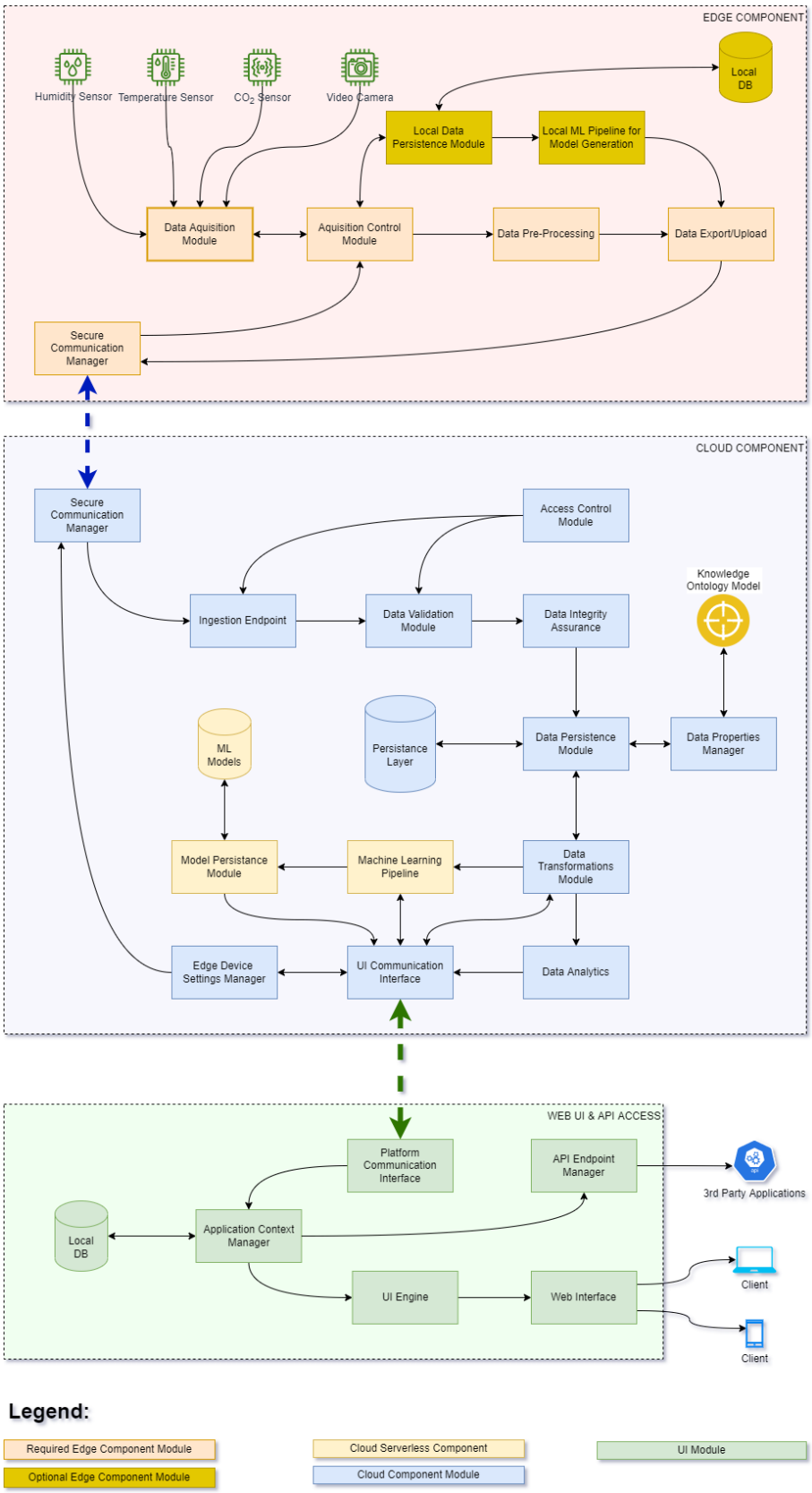

Figura 4: Arhitectura detaliată pentru platforma *MUSHNOMICS*

ontologic). Aceste module vor păstra cunoștințele pe bază de proiect, permițând publicarea sau partajarea ontologiilor;

- **Multi-Paradigm Machine Learning Pipeline** - are mai multe metodologii care pot fi selectate per proiect, care folosesc datele pentru a genera modele sau folosesc modele parțiale pentru a actualiza un model general.

5.2.4 Validarea Arhitecturii

Validarea arhitecturii a fost realizată prin utilizarea metodologiilor *Software Architecture Analysis Method* (SAAM) [15] și *Active Reviews for Intermediate Designs* (ARID) [3], care au fost utilizate la întâlnirile de revizuire a arhitecturii cu părțile interesate implicate.

Maparea dintre componentele și cerințele platformei a fost analizată și discutată amănunțit, disecând orice preocupări ridicate de către părțile interesate și atenuând orice problemă potențială. Aceste metodologii conduc la un proiect incremental pentru arhitectura finală, având cel puțin 4 trepte până la finalizarea arhitecturii. La sfârșitul procesului de validare, s-a dovedit că arhitectura finală suportă toate cerințele pe care le-a cerut beneficiarii reali.

Tabela 1: Corelația dintre cerințele funcționale și modulele din arhitectură

Cerințe Funcţionale	**Module care sunt utilizate pentru implementarea funcționalității**
FR1	Module ML din Componentele Edge, pipeline-ul de integrare a modelului în Componenta Cloud
FR2	Pipeline flexibil de generare a modelelor în componenta cloud, componentă UI, comunicarea cu componentele Edge
FR3	Modul de control al accesului, Modulul de persistență model
FR4	Metodologia de învățare federată în cadrul pipeline-ului de învățare automată
FR5	Modul de control al accesului, modul de persistență model, model de ontologie a cunoștințelor
FR6	Modul de control al accesului, API Endpoint Manager

5.3 Metodologii de lucru cu datele

5.3.1 Centralizarea datelor în cloud

Centralizarea datelor este standardul curent în industriile care conțin componente cu învățare automată. Metodologia din prezent este bazată pe o bază de date ce poate fi interogată și care conține toate datele disponibile pentru acest proces. Sistemul va aplica algoritmi de prelucrare a datelor, cu scopul atât de a elimina datele incorecte sau lipsă, dar și cu scopul de a extrage statistici și diferite concluzii din aceste date (transformarea în informație utilă).

Învățarea într-un mod centralizat oferă algoritmilor cantitatea maxim disponibilă de date, care se prelucrează în diferite moduri: eliminarea eventualului zgomot de diferite naturi, eliminarea sau înlocuirea/repararea datelor corupte, gruparea, generarea de coloane noi/compozite și orice alte prelucrări ar mai fi nevoie (în funcție de setul de date disponibil).

Probleme intervin în această metodologie în momentul în care datele sunt slabe calitativ (conțin mult zgomot, multe valori nule, probleme la citirea senzorilor, etc). În această situație nu prea sunt modalități multe de a rezolva această problemă. Metodologii de lucru cum ar fi învățarea colaborativă [8, 9] și învățarea ce ține cont de context [14] reușesc să îmbunătățească calitatea modelului generat, prin utilizarea fie a datelor din alte surse, fie prin utilizarea altor parametrii ce vor întregi contextul.

Avantajele centralizării datelor sunt multiple:

- Centralizarea oferă context disponibil oricând. Datele fiind disponibile, algoritmul poate pur și simplu încărca și utiliza aceste date;

- Existența datelor din alte locații cu o puternică corelație. Aceste surse vor întregi peisajul setului de date, menținând în același timp și panta de variație (gradientul). Algoritmul poate alege să folosească automat sau implicit aceste date;
- Se pot utiliza oricare algoritmi de învățare automată. Fiecare dintre aceștia sunt concepuți pentru a fi utilizați în acest fel, așadar utilizatorul poate chiar să facă multiple teste/comparații pentru a alege cel mai potrivit algoritm;
- Se pot utiliza cu succes algoritmii de tip *ensemble*, în care rezultatul se supune la votul a cel puțin trei algoritmi diferiți.

În schimb, această metodologie are și unele dezavantaje majore, care împing cercetarea spre alte direcții atunci când vine vorba de sistemele de producție:

- Expunerea datelor. Centralizarea presupune ca toate datele, de la toți clienții, să fie fizic încărcate în cloud. Oricine are acces la baza de date va avea implicit acces la toți parametrii, datele și cunoștințele companiei. De aceea, această modalitate de lucru cu datele este extrem de vulnerabilă atacurilor de orice fel;
- Vulnerabilitatea la atacuri, care se datorează faptului că toate datele sunt centralizate și gata de a fi sustrase în momentul compromiterii sistemului;
- Conexiunea permanentă și rapidă la internet este vitală. Având în vedere cantitatea enormă de date ce se pot genera într-un astfel de sistem, acestea trebuie transferate și stocate, ceea ce pune o presiune suplimentară atât pe conexiunea de internet cât și pe cerințele pentru componenta cloud.

Deși metodologia aceasta este una consacrată, se pot observa diferite dezavantaje care pot fi considerate majore și care ne dau direcția spre cercetare suplimentară, atât fundamentală cât și aplicativă.

5.3.2 Crearea de modele parțiale, Învățare Federativă

Învățarea Federativă [10, 16] este una dintre metodologiile noi care îmbunătățesc dramatic funcționarea învățării automate. Deși departe încă de a deveni un standard, asta datorită nivelului foarte mic alt TRL-ului (estimativ 3-4), învățarea federativă este deja o potențială alternativă pentru centralizarea datelor.

Această modalitate elimină complet nevoia de a centraliza datele. Sistemul local al clientului va stoca datele temporar, fiind necesare doar până la generarea unui model parțial, ce poate fi privit și ca o actualizare pentru un model anterior. Aceste modele parțiale sau actualizări se pot apoi încărca pe platforma cloud pentru a putea fi generat un model generalizat ce va putea fi ulterior utilizat de către personal din cercetare sau alte companii din industrie.

Modul de funcționare este ușor diferit față de învățarea clasică. Obținerea unui model presupune cel puțin doi pași. Primul pas implică generarea unui model local, folosindu-se hardware-ul utilizatorului. Etapa aceasta implică toți pașii de prelucrare a datelor cum s-a arătat în secțiunea precedentă. Al doilea pas presupune încărcarea modelelor parțiale în platforma cloud și generarea unui model ce va combina toate aceste modele parțiale.

Avantajele acestei metode sunt:

- Ne-expunerea datelor prin lipsa centralizării, face ca impactul unei posibile breșe de securitate din sistem să se diminueze. Chiar dacă datele în sine se pot interpreta, la modelele salvate nu există o astfel de posibilitate, deoarece nu se cunoaște contextul modelului (cum ar fi algoritmul folosit, parametrii acestuia). Astfel atacurile la platformă nu mai aduc pericolul expunerii datelor clienților;
- Limitarea cantității de date transferate este, probabil, una dintre cele mai importante avantaje ale unui astfel de sistem. Banda necesară pentru transferul datelor poate avea, în cele mai deficitare scenarii, o creștere exponențială raportată la numărul de senzori din sistem. În schimb, în cazul învățării federative, creșterea este logaritmică, indiferent de dimensiunea setului de date nefiind necesară decât transferarea modelului rezultat;
- Cerințe hardware mai mici pe cloud; sunt o consecință directă a faptului că nu trebuie să transferăm și să menținem întreaga bază de date. De asemenea, marea parte a antrenării modelelor se face pe nodurile clienților,

ceea ce înseamnă un efort mult redus pentru componenta cloud. De aceea, cerințele hardware pentru sistemul cloud sunt mult mai mici, limitate doar la stocarea modelelor și la puterea de procesare necesară pentru federalizarea acestor modele;

- Îmbunătățirea securității datelor din cadrul sistemului, deoarece datele sunt menținute la client. Astfel, fiecare client este direct răspunzător de securitatea proprie a secretelor de companie. Atacurile asupra platformei nu vor avea ca și consecință expunerea datelor clienților, scăzând astfel drastic impactul breșelor de securitate.

Dezavantajele acestei metode sunt:

- Cantitatea limitată de date unui singur nod, practic doar ce senzorii acelui nod vor înregistra. Aceasta oferă putere limitată modelului generat, acesta fiind specific instanței locale a aplicației, generalizarea acestuia fiind extrem de limitată sau chiar non-existentă;
- Modul de prelucrare a datelor lipsă este diferit, deoarece datele nule sau corupte nu se pot pur și simplu elimina. Importanța fiecărei înregistrări crește, așadar este nevoie de alte strategii de prelucrare a datelor, astfel încât să fie pregătite pentru procesul de învățare automată și generarea unui model;
- Sistemul rezultat este mai complex. Rolul fiecărui nod nu mai este doar acela de a împinge datele într-un sistem de tip cloud, ci acela de a prelucra local datele, a genera și valida un model de învățare automată, precum și de a trimite în cloud acest model. O modalitate de a clasifica modelele în contextul calității lor, este necesar pentru a crește validitatea și viabilitatea întregului sistem. De asemenea, cerințele hardware pentru noduri sunt mai mari, datorită prelucrărilor suplimentare ce se aplică.

5.3.3 Formalizarea cunoștințelor

Interacțiunea dintre interfață și sistemul de învățare automată presupune și setarea unor limite ale valorilor parametrilor din sistem, precum și alți parametrii de funcționare ai platformei [4, 11]. Aceste valori limită (sau valori ideale) fac parte din noțiunea de rețetă de creștere, formalizată în cadrul platformei *MUSHNOMICS* . Toate aceste informații extrase din cadrul datelor se pun în baza de cunoștințe, o bază ce va fi specifică fiecărui client în parte. Formalizarea acestora implică următoarele noțiuni, ce sunt implementare în cadrul sistemului, sub formă ontologică:

- **Container 1 Compost**
  - Materie prima pentru compost - acestea pot fi atât deșeuri alimentare cât și frunze, lemne, paie, etc.;
  - Rețetă compost - raportul de materie primă pentru optimizarea creșterii ciupercilor;
  - Prelucrare compost - procesul de amestecare și combinare a materiei primei după rețetă;
  - Compostul rezultat - amestecul de materie primă folosit ca substrat și îngrășământ pentru ciuperci;
- **Container 2 Producția de ciuperci**
  - Substratul - combinația de ingrediente folosite ca mediu de creștere pentru ciuperci;
  - Miceliu - pentru însămânțarea substratului și demararea procesului de creștere și producție a ciupercilor;
  - Senzori Smart - senzori de monitorizarea mediului înconjurător de creștere pentru optimizarea și automatizarea procesului de producție;
- **Container 3 Valorizarea substratului**
  - Compost - acest amestec se poate folosi pentru o altă generație de ciuperci;
  - Levigatul - lichidul care se formează în urma compostului trebuie procesat, egalizat și pasteurizat pentru a avea un impact mic asupra mediului.

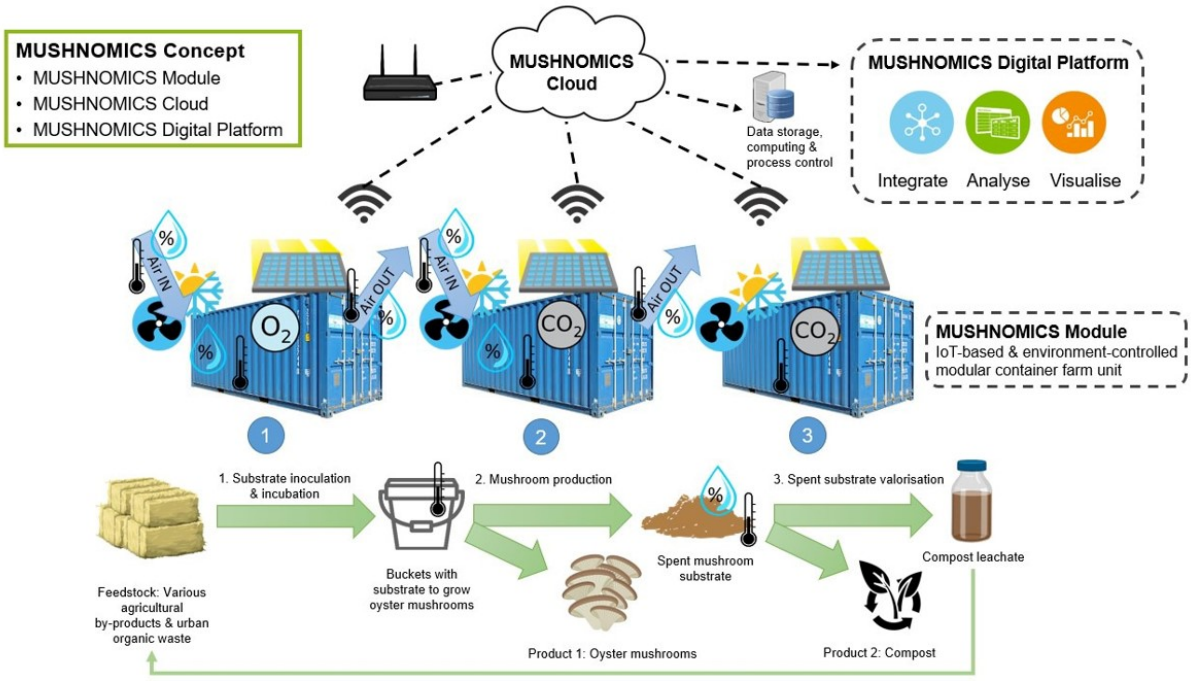

Figura 5: Modulul MUSHNOMICS bazat pe IoT: concept și abordare

5.4 Interfața cu utilizatorul

Implementarea interfeței cu utilizatorul a început prin stabilirea tipologiei persoanei care va utiliza platforma (eng.: *Persona*) [13]. Pentru proiectul *MUSHNOMICS* s-au definit 2 tipuri de utilizatori care vor utiliza platforma, și anume: *cercetători* și *companiile*. Următoarele caracteristici ale celor două tipuri de *Persona* care vor utiliza platforma:

Cercetător Persona

- are între 35 și 50 de ani;
- trăiește în mediu urban, în zona universității de care aparține;
- este cercetător post-doctoral într-un domeniu agricol;
- dispune de suficient timp liber pentru studiu;
- pasionat de moduri de hrănire sănătoase;
- cu un interes sincer pentru protecția mediului și sustenabilitate;
- își dorește ocazia de a-și cultiva singur hrana;
- pasionat de studiu individual și de descoperirea individuală.

Companie Persona

Interfața companiei cu aplicația este întotdeauna o persoană fizică, care în cazul acesta poate fi ori un tehnician/operator, sau un inginer agricol. Caracteristicile acestei *persona* sunt:

- are între 25 și 40 de ani;
- absolvent de studii universitare în domeniul agricol;
- trăiește în mediu rural, sau rural mic, aproape de ferma la care își desfășoară activitatea;
- pasionat de sustenabilitate și optimizarea lanțului de producție agricolă;
- are o viziune de viitor referitoare la evoluția agriculturii pe plan mondial și dorește să participe activ la evoluția științei agricole.

Informații care trebuie să existe pe **în interfața de utilizare pentru utilizatorii finali**:

- are cel puțin o instalație activă și nu poate accesa date decât dacă le deține;
- poate vedea statistici despre propriile date/instalații;
- poate vedea ieșirea modelului pe propriile date;
- poate genera un model pe baza propriilor date;
- poate folosi un model generic generat de oameni de știință;
- poate sprijini consorții de companii (partajare de date/modele);
- setarea proiectului/instalației cu intrări;
- e-mailuri pentru diverse notificări;
- notificări în browser.

Informații care trebuie să existe în interfața de utilizare pentru **cercetători**:

- Nu are o instanță instalată, dar poate accesa toate seturile de date publice (care sunt etichetate ca publice de către utilizatorii finali);
- Poate accesa modelele de învățare federativă ce au fost puse la dispoziție de utilizatorii finali;
- poate vedea analize avansate asupra datelor;
- poate face estimări asupra datelor (regresia liniară/logistică asupra datelor viitoare);
- poate proiecta și testa diverși algoritmi de învățare automată, pentru a vedea care din ei sunt mai potriviți;
- Dispune de un tablou de bord personalizabil;

6 Deliverables and outputs

6.1 Livrabile

In perioada raportata am furnizat livrabilele din tabela 2.

Tabela 2: Livrabilele elaborate în perioada de raportare

Nr. livrabil	Termen	Livrabil	Status livrabil
D2.1	M12	Raport despre algoritmi de ultimă generație	Livrat în M12
D4.1	M18	Arhitectura și funcționalitățile platformei digitale	Livrat în M18

7 Diseminare și exploatare

7.1 Activități de diseminare

Proiectul a fost diseminat în următoarele moduri:

- pe pagina web: https://research.holisun.com/ro/proiecte/agriculture-4-0/mushnomics-ro, având un număr de 180 de vizitatori lunari
- pe contul de LinkedIn: https://www.linkedin.com/company/holisun, cu 400 de adepți
- pe contul de LinkedIn al proiectului https://www.linkedin.com/company/mushnomics-project/, cu 26 de urmăritori
- pe pagina de Facebook: https://www.facebook.com/Holisun.IT/, având 1881 de urmăritori
- pe contul de Twitter al proiectului: https://twitter.com/mushnomics, având 15 de urmăritori.

Au fost desfășurate o serie de activități de diseminare în cadrul unor evenimente de afaceri, expoziții și evenimente de brokeraj sau networking, listate în Tabelul 3.

Tabela 3: Lista de activități de diseminare

Nume	Data	Link	Participanți	Rezultate
IPEC 2022 AI & Sustainability	08-09/03/2022	https://innoelectro-expo-with-hybrid-b2b.b2match.io	Rudolf Erdei	Prezentare *MUSHNOMICS*
ISE Open Innovation Challenge 2022	10-18/05/2022	https://ise-congress-open-challenge.b2match.io/	Rudolf Erdei, Daniela Delinschi	Prezentare *MUSHNOMICS*
Applied Artificial Intelligence Conference 2022	25/05/2022	https://aaic2022.b2match.io/	Rudolf Erdei	Prezentare *MUSHNOMICS*
B2B meetings CONNECTO 2022 Mostar	26-28/07/2022	https://connecto.ba/	Daniela Delinschi	Prezentare *MUSHNOMICS*
Co-Matching Virtual Meetings 2022	25/08/2022	https://co-matching-2022.b2match.io/	Rudolf Erdei	Prezentare *MUSHNOMICS*

7.1.1 Alte activități de diseminare

Proiectul a mai fost diseminat prin următoarele canale:

Tabela 4: Lista canale de diseminare

Canal	Adresa	Indicatori
Pagina web	https://mushnomics.org/	Vizitatori: 365
LinkedIn	https://www.linkedin.com/company/mushnomics-project/	Postări: 19
Twitter	https://twitter.com/mushnomics	Tweet-uri: 36

7.2 Exploatare

Exploatarea rezultatelor implică înțelegerea pieței, pentru a se putea proiecta un plan de afaceri potrivit acesteia. Pentru aceasta, s-a luat în considerare abstractizarea (sau generalizarea) conceptelor ce există în cadrul platformei, pentru a putea fi adaptată ușor și altor piețe ce utilizează sisteme similare, cum ar fi piața auto, logistica, microclimatele, sistemele smart-home, și multe altele, piețe în care aplicațiile IoT sunt în centrul atenției.

Numărul tot mai mare de aplicații IoT crește nevoia de securitate, ceea ce conduce piața. IoT va consta din miliarde de dispozitive digitale, servicii și alte obiecte fizice care au potențialul de a se conecta, interacționa și face schimb de informații fără probleme. Piața auto devine, de asemenea, un element important al IoT, conectat la lumea exterioară printr-un număr tot mai mare de tehnologii wireless. Deși beneficiile conexiunii îmbunătățite sunt benefice, de asemenea, deschide o mare de oportunități atacatori. Conform informațiilor privind securitatea cibernetică a anului 2020 de la Upstream Security, din 2018 până în 2019, a existat o creștere cu 99% a incidentelor de securitate cibernetică.

Astfel, activitățile desfășurate în cadrul prezentului proiect, și anume de proiectare și implementare a unui sistem rezilient la atacuri, distribuit, rapid și modern, asigură posibilitatea Holisun de a proiecta sisteme de ultimă generație.

8 Concluzii

Arhitectura propusă oferă flexibilitatea arhitecturală necesară pentru a integra mai multe paradigme de învățare automată, inclusiv învățarea federativă, care este și punctul principal al acestei cercetări. Platforma oferă, de asemenea, comunicații securizate, prin criptarea tuturor transferurilor de date între nodurile de margine și componenta centrală Cloud. Prin limitarea cantității de date transferate între componente și schimbarea modului în care învățarea este accentuată, lățimea de bandă necesară este limitată la minimum, astfel încât canalul de comunicare este o parte necritică a sistemului, spre deosebire de multe alte paradigme ML.

Dezavantajele includ incapacitatea sistemului de a efectua operațiuni în timp real/critice în timp. În unele cazuri specifice, acest aspect poate fi crucial. În cazul nostru de utilizare în agricultură, fereastra de timp pentru rezolvarea problemelor specifice pe care platforma le-ar putea prezice este destul de mare (ore sau poate chiar zile), astfel încât performanța în timp real nu este necesară în acest caz.

8.1 Activități viitoare

Activitățile viitoare se axează pe finalizarea lucrului și exploatarea rezultatelor. Mai exact, preconizăm activități referitoare la:

1. **Aducerea platformei la maturitate tehnologica TRL 6**. Pentru aceasta, toate funcționalitățile vor fi testate în condiții relevante de funcționare, atât în cadrul PILZE Kft, cât și împreună cu alți potențiali clienți ce vor fi cooptați;
2. **Aducerea Interfeței cu Utilizatorul la maturitate tehnologica TRL 6**. Datorită flexibilității, această interfață va putea fi implementată și altor tipuri de aplicații;
3. **Dezvoltarea unei noi linii de business** cu ajutorul platformei, ce va conține atât direcția curentă, în producția de ciuperci, cât și direcții alternative, cu ajutorul altor clienți din domeniul agricol.

Referințe

[1] Bebensee, T., Weerd, I.v.d., Brinkkemper, S.: Binary priority list for prioritizing software requirements. In: International working conference on requirements engineering: foundation for software quality. pp. 67–78. Springer (2010)

[2] Blockeel, H., Vanschoren, J.: Experiment databases: Towards an improved experimental methodology in machine learning. In: European Conference on Principles of Data Mining and Knowledge Discovery. pp. 6–17. Springer (2007)

[3] Clements, P.C.: Active reviews for intermediate designs. Tech. rep., CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST (2000)

[4] Delinschi, D., Erdei, R., Matei, O.: Ontology driven high performance messaging system for distributed software platforms. In: 2022 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR). pp. 1–6. IEEE (2022)

[5] Erdei, R., Delinschi, D., Matei, O.: Security centric scalable architecture for distributed learning and knowledge preservation. In: International Workshop on Soft Computing Models in Industrial and Environmental Applications. pp. 655–665. Springer (2023)

[6] Kazman, R., Klein, M., Barbacci, M., Longstaff, T., Lipson, H., Carriere, J.: The architecture tradeoff analysis method. In: Proceedings. fourth ieee international conference on engineering of complex computer systems (cat. no. 98ex193). pp. 68–78. IEEE (1998)

[7] Kühne, T.: On model compatibility with referees and contexts. Software & Systems Modeling **12**(3), 475–488 (2013)

[8] Laal, M., Ghodsi, S.M.: Benefits of collaborative learning. Procedia-social and behavioral sciences **31**, 486–490 (2012)

[9] Laal, M., Laal, M.: Collaborative learning: what is it? Procedia-Social and Behavioral Sciences **31**, 491–495 (2012)

[10] Li, L., Fan, Y., Tse, M., Lin, K.Y.: A review of applications in federated learning. Computers & Industrial Engineering **149**, 106854 (2020)

[11] Matei, O., Erdei, R., Delinschi, D.: Multimodal transportation overview and optimization ontology for a greener future. In: Computer Science On-line Conference. pp. 158–172. Springer (2021)

[12] Olson, R.S., Moore, J.H.: Tpot: A tree-based pipeline optimization tool for automating machine learning. In: Workshop on automatic machine learning. pp. 66–74. PMLR (2016)

[13] Pruitt, J., Adlin, T.: The persona lifecycle: keeping people in mind throughout product design. Elsevier (2010)

[14] Schilit, B., Adams, N., Want, R.: Context-aware computing applications. In: 1994 first workshop on mobile computing systems and applications. pp. 85–90. IEEE (1994)

[15] Vogel, C.: Saam (software architecture analysis method). Universität Karlsruhe p. 1 (2008)

[16] Yang, Q., Liu, Y., Cheng, Y., Kang, Y., Chen, T., Yu, H.: Federated learning. Synthesis Lectures on Artificial Intelligence and Machine Learning **13**(3), 1–207 (2019)

[17] Yin, D., Chen, Y., Kannan, R., Bartlett, P.: Byzantine-robust distributed learning: Towards optimal statistical rates. In: International Conference on Machine Learning. pp. 5650–5659. PMLR (2018)

info@holisun.com
http://www.holisun.com
+4-0262-226.778
Versiune document: 1.0 din 25.11.2022