AMERICAN JOURNAL OF AI CYBER COMPUTING MANAGEMENT
E-ISSN:3069-0102
VOL.5, NO. 4(2025)

AUTONOMOUS MACHINE LEARNING MODELLING USING A TASK ONTOLOGY

Dr. P. Janaki Ramulu[1], Vasikarla Sridevi[2], Sujal Agarwal[3], Thummala Abhishekitha[4] ,Seelam Keshava Kumar Rao[5]

[1]*Professor, Department of CSE, Samskruti College Of Engineering And Technology, Kondapur (V), Ghatkesar (M), Medchal (D), Hyderabad, India.*

[2,3,4,5]*Student, Department of CSE, Samskruti College Of Engineering And Technology, Kondapur (V), Ghatkesar (M), Medchal (D), Hyderabad, India.*

Corresponding email ID: principal@samskruti.ac.in

ABSTRACT:

Recently, many researchers are intensely engaged in investigation on the artificial intelligence technology that recognizes, learns, inferences, and acts on external information in a wide range of fields by combining technologies of computing, big data and machine learning algorithms. The artificial intelligence technology is currently used in almost all industries, and many machine learning experts are working on integrating and standardizing various machine learning tools so that non-experts can easily apply them to their domain. The researchers are also studying an autonomous machine learning as well as ontology construction for standardizing the machine learning concepts. In this paper, we classify typical problem-solving steps for autonomous machine learning as tasks, and present a problem-solving process. We propose the modeling method of an autonomous machine learning using a process of the task execution on machine learning such as workflow. The proposed task ontology-based machine learning model de fines a task-based process grouping scheme of UML activities. And it will automatically generate and extend the machine learning models by transformation rules based on common elements and structures (relationships and processes between elements).

Keywords: UML, artificial intelligence, ontology

1 INTRODUCTION

Artificial intelligence technology has become one of the most essential tools in research and business context recently. Most of the machine learning frameworks are open-source, so the entry of barriers into machine learning are lowered. The typical machine learning frameworks include TensorFlow, Eras, Caffe, Scikit-learn, and Theano implemented in programming languages such as Python, Java, and R. In this respect, many machine learning experts are working on integrating and standardizing various tools so that machine learning nonexperts can easily apply them to their domains. On the other hand, an autonomous machine learning is still in its infancy, and some techniques provide the ability to reduce the unnecessary tasks that are progressively refined to prepare the model and improve its accuracy. The tools of autonomous machine learning provide an optimal algorithm for machine learning tasks and functions to determine the hyper-parameter setting through self-analysis. The typical tools include Auto sklearn, Auto-Weka, H2o Driverless AI and Google's Auto ML. In this paper, we describe a typical problem solving process for the machine learning as tasks, present their procedure, and propose the modeling method of an autonomous machine learning for using task execution processes. The modeling method of autonomous machine learning based on the task ontology define a structure based grouping method of the UML(Unified Modeling Language) activities and implement a function to automatically generate models based on common elements and structures. The purpose of the proposed autonomous machine learning model is to model

Received: 20-09-2025 | Accepted: 24-10-2025 | Published: 31-10-2025 | www.ajaccm.com

autonomous machine learning by reusing existing resources and producing new knowledge through relearning it.

Task ontologies

Ontology is defined in various fields depending on the field of applications. In the field of artificial intelligence, it is an explicit and formal specification of how objects and concepts described in the field of interest. In the Semantic Web, an ontology plays a very important role in processing, sharing, and reusing the knowledge for exchanging information between different databases. An ontology is also defined as an explicit description of concepts, attributes, constraints, and relationships between them on the domains.On the other hand, domain ontology can be defined as an 'explicit protocol for conceptualization' of the problem. A task ontology is defined as 'extracting and organizing the concepts and relations existing in the problem-solving process domain-independent'. In particular, a task ontology is a specification of the concept structure for the task execution process. Thus, the core concept is the subject of processing and the procedure of processing for a problem solving. In general, a person becomes a subject in a task ontology. However, in this paper, agents (programs) become subjects to perform the tasks.

Machine learning ontologies

ML schema is a top-level ontology that provides classes, properties, and constraints for machine learning algorithms, datasets and experimentation suggested by the W3C(ML Schema community group). It can be easily extended and refined, and can be mapped to other domain ontologies developed in the field of machine learning and data mining. MEX vocabulary has been designed to solve the share of provenance information in a lightweight form. The extended PROV ontology provides a model for representing, capturing, and sharing provision information on the Web. This can enable the use of analytical data and code so that another person can reuse the results. The code and the markup language are written in a single file, and processed to create a document.

Autonomous machine learning modeling:

Autonomous machine learning modeling is the work for standardization and abstraction to the core of the components base on the meta information of the machine learning. The model consists of the task and process and saves as the method library(API). It defines into small units for modular and systematization of its components. The defined components redefine as a UML-based metamodel for the consistency, traceability, reusability, and implementation-ready between tasks and the results. So the core class of the UML-based meta-model consist of tasks and processes. The Knowledge of the autonomous machine learning also describes a small task unit based on the MEXvocabulary. depicts a part of the knowledge of the object detection using the "YOLO" of the deep learning algorithm.

2 LITERATURESURVEY

TITLE:TensorFlow: A system for large-scale machine learning

AUTHOR: Mart´ın Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng

ABSTRACT: TensorFlow is a machine learning system that operates at large scale and in heterogeneous environments. TensorFlow uses dataflow graphs to represent computation, shared state, and the operations that mutate that state. It maps the nodes of a dataflow graph across many machines in a cluster, and within a machine across multiple computational devices, including multicore CPUs, generalpurpose GPUs, and custom-designed ASICs known as

Tensor Processing Units (TPUs). This architecture gives flexibility to the application developer: whereas in previous "parameter server" designs the management of shared state is built into the system, TensorFlow enables developers to experiment with novel optimizations and training algorithms. TensorFlow supports a variety of applications, with a focus on training and inference on deep neural networks. Several Google services use TensorFlow in production, we have released it as an open-source project, and it has become widely used for machine learning research. In this paper, we describe the TensorFlow dataflow model and demonstrate the compelling performance that TensorFlow achieves for several real-world applications.

TITLE: Caffe: Convolutional Architecture for Fast Feature Embedding

AUTHOR: Yangqing Jia∗ , Evan Shelhamer∗ , Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, Trevor Darrell SUBMITTED to ACM MULTIMEDIA 2014 OPEN SOURCE SOFTWARE COMPETITION UC Berkeley EECS, Berkeley, CA 94702

ABSTRACT: Caffe provides multimedia scientists and practitioners with a clean and modifiable framework for state-of-the-art deep learning algorithms and a collection of reference models. The framework is a BSD-licensed C++ library with Python and MATLAB bindings for training and deploying generalpurpose convolutional neural networks and other deep models efficiently on commodity architectures. Caffe fits industry and internet-scale media needs by CUDA GPU computation, processing over 40 million images a day on a single K40 or Titan GPU (≈ 2.5 ms per image). By separating model representation from actual implementation, Caffe allows experimentation and seamless switching among platforms for ease of development and deployment from prototyping machines to cloud environments. Caffe is maintained and developed by the Berkeley Vision and Learning Center (BVLC) with the help of an active community of contributors on GitHub. It powers ongoing research projects, large-scale industrial applications, and startup prototypes in vision, speech, and multimedia.

TITLE: Exposé: An ontology for data mining experiments

AUTHOR:V. Joaquin, S. Larisa

ABSTRACT: Research in machine learning and data mining can be speeded up tremendously by moving empirical research results out of people's heads and labs, onto the network and into tools that help us structure and filter the information. This paper presents Exposé, an ontology to describe machine learning experiments in a standardized fashion and support a collaborative approach to the analysis of learning algorithms. Using a common vocabulary, data mining experiments and details of the used algorithms and datasets can be shared between individual re-searchers, software agents, and the community at large. It enables open repositories that collect and organize experiments by many researchers. As can been learned from recent developments in other sciences, such a free exchange and reuse of experiments requires a clear representation. We therefore focus on the design of an ontology to express and share experiment meta-data with the world.

TITLE: Task ontology: Ontology for building conceptual problem-solving models

AUTHOR:I. Mitsuru, S. Kazuhisa, K. Osamu, M. Riichiro

ABSTRACT: We have investigated the property of problem-solving knowledge and tried to design its ontology, that is, Task ontology. The main purpose of this paper is to illustrate a Conceptual Level Programming Environment (named CLEPE) as an implemented system based on Task ontology. CLEPE provides three major advantages as follows. (A) It provides human-friendly primitives in terms of which

users can easily describe their own problem-solving process (descriptiveness, readability). (B) The systems with task ontology can simulate the problem-solving process at an abstract level in terms of conceptual level primitives (conceptual level operationality). (C) It provides ontology author with an environment for building task ontology so that he/she can build a consistent and useful ontology. In this paper, firstly we briefly introduce the concept of task ontology. Secondly, CLEPE and its design principle is described. In CLEPE, one can represent his/her own problem-solving knowledge and realize the conceptual-level execution.

3. SYSTEMANALYSIS

EXISTINGSYSTEM

The domain ontology can be defined as an 'explicit protocol for conceptualization' of the problem. A task ontology is defined as 'extracting and organizing the concepts and relations existing in the problem solving process domain-independent'. In particular, a task ontology is a specification of the concept structure for the task execution process. Thus, the core concept is the subject of processing and the procedure of processing for a problem solving. In general, a person becomes a subject in a task ontology. However, in this paper, agents (programs) become subjects to perform the tasks.

An autonomous machine learning is still in its infancy, and some techniques provide the ability to reduce the unnecessary tasks that are progressively refined to prepare the model and improve its accuracy.

PROPOSEDSYSTEM

In the proposed system, we describe a typical problem solving process for the machine learning as tasks, present their procedure, and propose the modeling method of an autonomous machine learning for using task execution processes. The modeling method of autonomous machine learning based on the task ontology define a structure based grouping method of the UML(Unified Modeling Language) activities and implement a function to automatically generate models based on common elements and structures. The purpose of the proposed autonomous machine learning model is to model autonomous machine learning by reusing existing resources and producing new knowledge through relearning it.

ADVANTAGES OF PROPOSED SYSTEM:

ML schema is a top-level ontology that provides classes, properties, and constraints for machine learning algorithms, datasets and experimentation suggested by the W3C(ML Schema community group).

It can be easily extended and refined, and can be mapped to other domain ontologies developed in the field of machine learning and data mining.

4 SYSTEMREQUIREMENTS

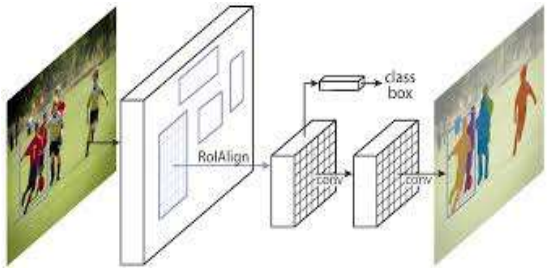
FUNCTIONALREQUIREMENTS

Thesearetherequirementsthattheenduserspecifica llydemandsasbasicfacilitiesthatthesystem shouldoffer.Allthesefunctionalitiesneedtobenece ssarilyincorporatedintothesystemasapart of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements. Functional requirements are specifications that define the specific behavior or functions of a system, software, or product. They describe what the system should do, its features, and how it should perform under certain conditions. In the context of software development, functional requirementsserveasablueprintforthedevelopers, guidingtheminbuildingasystemthatmeets the needs and expectations of the users and stakeholders.

- User
- Admin

SYSTEMDESIGN

SYSTEMARCHITECTURE



UML DIAGRAM'S:

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computersoftware.InitscurrentformUMLiscompr isedoftwomajorcomponents:aMeta-model andanotation.

Inthefuture,someformofmethodorprocessmayals obeaddedto;orassociated with, UML.

TheUnifiedModelingLanguageisastandardlangua geforspecifying,Visualization,Constructing anddocumentingtheartifactsofsoftwaresystem,as wellasforbusinessmodelingandothernon-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software developmentprocess.TheUMLusesmostlygraphic alnotationstoexpressthedesignofsoftware projects.

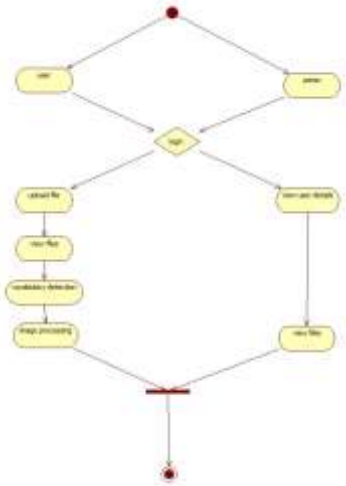
GOALS:

ThePrimary goalsin the design oftheUMLareasfollows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provideextendibilityandspecializationme chanismstoextendthecoreconcepts.
3. Beindependentofparticularprogramming languagesanddevelopmentprocess.
4. Provideaformalbasis forunderstanding themodeling language.
5. Encouragethegrowth of OOtools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integratebest practices.

ACTIVITYDIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions withsupport for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams canbe used to describethebusinessandoperationalstep-by-stepworkflowsofcomponentsinasystem.Anactivit ydiagram shows the overall flow of control.



IMPLEMENTATION

MODULES:

- user
- admin
- Machine learning

MODULEDESCRIPTION

User:

user information and task descriptions for the entire experiment. we describe a typical

problem-solving process for the machine learning as tasks, present their procedure, and propose the modeling method of an autonomous machine learning for using task execution processes. The collection of vocabulary is achieved by extracting words, a textbook or a machine learning library (API) tutorial, and then selecting keywords from the index and title of the textbook. The frequency of coincidence with the key word is calculated and labeled by a category item.

Admin:

The aim of admin is to approve the machine learning users . the entire data must be gathered to admin. The design of autonomous machine learning model is presented based on the task ontology. The tools of autonomous machine learning provide an optimal algorithm for machine learning tasks and functions to determine the hyper-parameter setting through self-analysis.

Machine learning:

Machine learning refers to the computer's acquisition of a kind of ability to make predictive judgments and make the best decisions by analyzing and learning a large number of existing data. The representation algorithms include deep learning, artificial neural network, decision tree, enhancement algorithm and so on. The key way for computers to acquire artificial intelligence is machine learning. Nowadays, machine learning plays an important role in various fields of artificial intelligence. Whether in aspects of internet search, biometric identification, auto driving, Mars robot, or in American presidential election, military decision assistants and so on, basically, as long as there is a need for data analysis, machine learning can be used to play a role.

```
BEGIN
DISPLAY "Welcome to the File Management and Processing System"

// Step 1: User or Admin Login
FUNCTION Login()
  INPUT username, password
  IF Authenticate(username, password) == TRUE THEN
    IF Role(username) == "user" THEN
      CALL UserModule(username)
    ELSE IF Role(username) == "admin" THEN
      CALL AdminModule(username)
    END IF
  ELSE
    DISPLAY "Invalid credentials. Please try again."
    TERMINATE
  END IF
END FUNCTION

// Step 2: User Functionalities
FUNCTION UserModule(username)
  DISPLAY "User Dashboard"
  DISPLAY "1. Upload File"
  DISPLAY "2. View Files"
  DISPLAY "3. Vocabulary Detection"
  DISPLAY "4. Image Processing"
  DISPLAY "5. Logout"

  CHOICE ← GET_USER_INPUT("Select an option: ")

  SWITCH(CHOICE)
    CASE 1:
      CALL UploadFile(username)
    CASE 2:
      CALL ViewFiles(username)
    CASE 3:
      CALL VocabularyDetection(username)
    CASE 4:
      CALL ImageProcessing(username)
    CASE 5:
      DISPLAY "User logged out successfully."
      RETURN
    DEFAULT:
      DISPLAY "Invalid selection. Try again."
```

```
    END SWITCH
END FUNCTION


// Step 3: Admin Functionalities
FUNCTION AdminModule(username)
    DISPLAY "Admin Dashboard"
    DISPLAY "1. View User Details"
    DISPLAY "2. View Uploaded Files"
    DISPLAY "3. Logout"

    CHOICE ← GET_USER_INPUT("Select an option: ")

    SWITCH(CHOICE)
        CASE 1:
            CALL ViewUserDetails()
        CASE 2:
            CALL ViewFiles("all")
        CASE 3:
            DISPLAY "Admin logged out successfully."
            RETURN
        DEFAULT:
            DISPLAY "Invalid selection."
    END SWITCH
END FUNCTION

// Step 4: Upload File (User)
FUNCTION UploadFile(username)
    FILE ← INPUT("Select file to upload: ")
    IF FILE IS NOT EMPTY THEN
        STORE FILE IN Database[username]
        DISPLAY "File uploaded successfully."
    ELSE
        DISPLAY "No file selected."
    END IF
END FUNCTION


// Step 5: View Files
FUNCTION ViewFiles(user)
    IF user == "all" THEN
        DISPLAY "Showing all uploaded files..."
        FOR EACH record IN Database:
            DISPLAY record
    ELSE
        DISPLAY "Your uploaded files:"
        FOR EACH file IN Database[user]:
            DISPLAY file
    END IF
END FUNCTION

// Step 6: Vocabulary Detection
FUNCTION VocabularyDetection(username)
    FILE ← SELECT_FROM(Database[username])
    DISPLAY "Performing vocabulary detection..."
    WORDS ← ExtractWords(FILE)
    UNIQUE_WORDS ← CountUnique(WORDS)
    DISPLAY "Detected", UNIQUE_WORDS, "unique words in file."
END FUNCTION

// Step 7: Image Processing
FUNCTION ImageProcessing(username)
    FILE ← SELECT_FROM(Database[username])
    DISPLAY "Running image processing..."
    PROCESSED_IMAGE ← ApplyFilters(FILE)
    DISPLAY "Image processed successfully and stored in database."
END FUNCTION

// Step 8: View User Details (Admin)
FUNCTION ViewUserDetails()
    DISPLAY "Registered Users:"
    FOR EACH user IN UserDatabase:
        DISPLAY user.Name, user.Email, user.Role
END FUNCTION


// Step 9: Program Execution
CALL Login()
```

DISPLAY "Program terminated successfully."

END

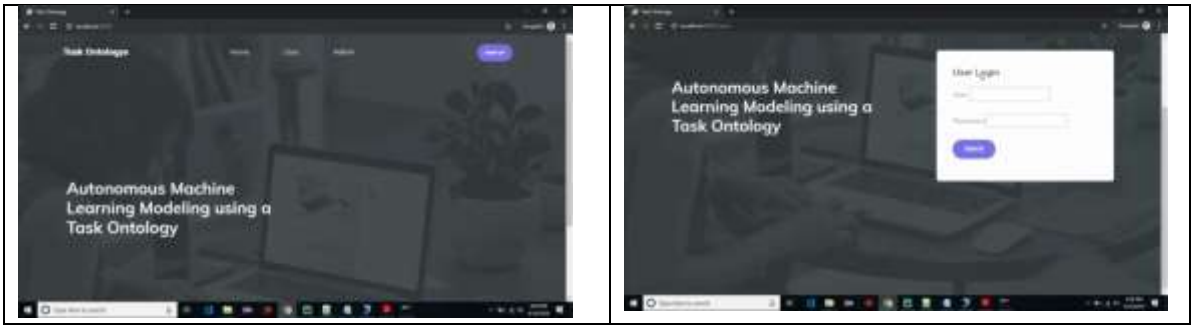
5. RESULTS/DISCUSSION

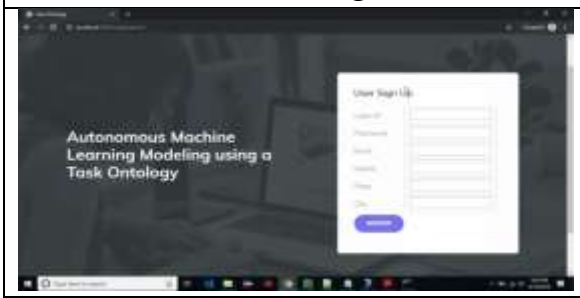
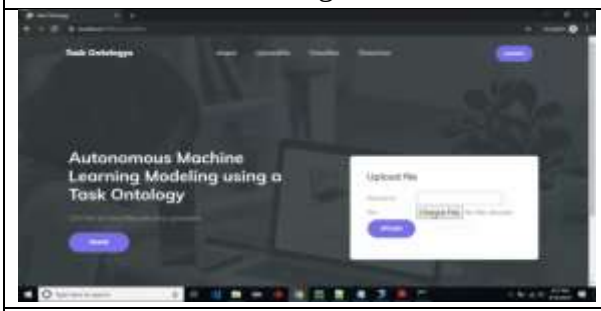
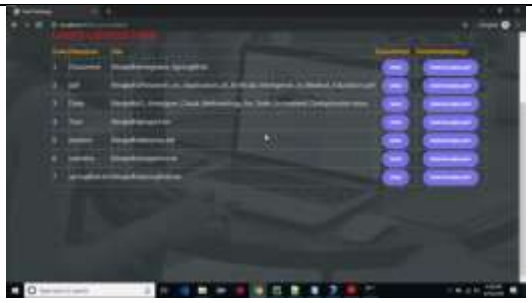
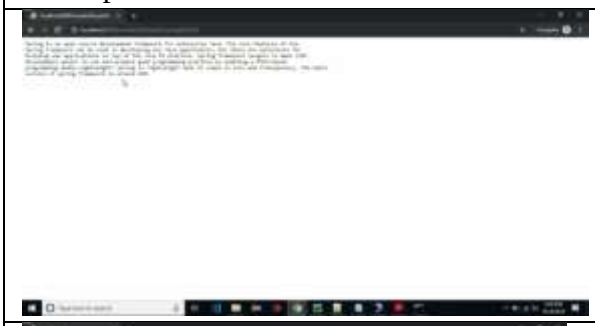
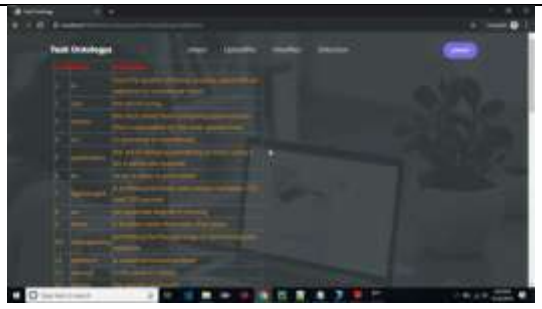
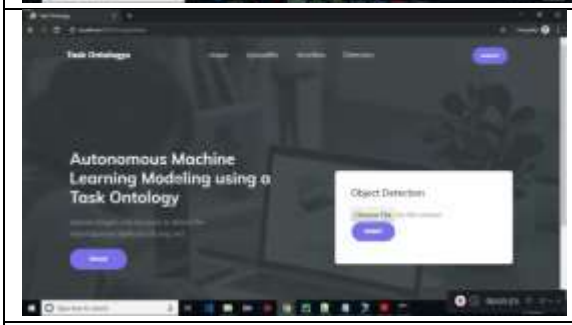
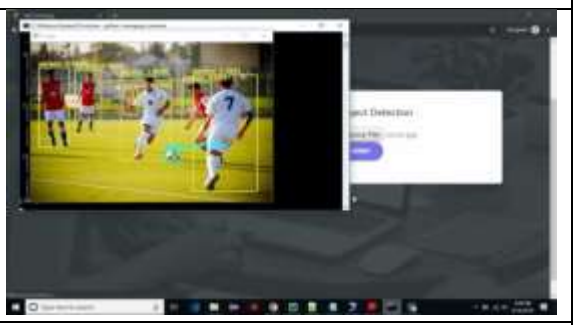
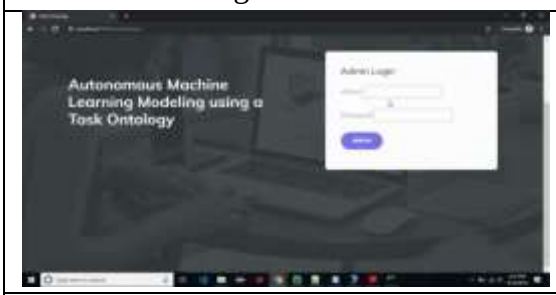
The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

TESTCASES

S.no	Test Case	Excepted Result	Result	Remarks (IF fails)
1	USER REGISTERED	If USER registration successfully.	Pass	If USER is not registered.
2	USER LOGIN	If USER name and password is correct then it will getting valid page.	Pass	If USER name or password is not correct.
4	ADMIN	USER rights will be accepted here.	Pass	If USER are not registered.
5	USER upload files	Choose or select USER files	Pass	If USER is not select or SEND MESSAGES
6	Vocabulary detection	All Vocabulary verification	Pass	If Vocabulary detection is not available.
7	Image processing	Image processing verification	Pass	If Image processing is not available.

6. SCREENSHOTS



Home Page	User login
	
User register	User home page
	
User upload file	User view file
	
	
Detection of images	Image detection
	
Admin login	User details

View files

7. CONCLUSION

In this paper, we extracted important keywords for constructing an ontology from papers and textbooks about machine learning. Moreover we designed a task ontology based on the MEX vocabulary. We also studied workflow for the autonomous machine learning model. The proposed method is applicable for automatic workflow according to designated autonomous level. Therefore, the non-experts are capable of doing complex tasks using the proposed method and can easily implement the machine learning model in a specific application

FUTURE SCOPE

The future scope of Autonomous Machine Learning Modelling Using a Task Ontology is vast and transformative, with implications for various domains such as artificial intelligence, automation, and decision-making systems. As machine learning continues to evolve, the integration of task ontology enables models to self-adapt, understand context, and enhance their learning capabilities with minimal human intervention. This approach is particularly valuable in fields like healthcare, finance, cybersecurity, and robotics, where intelligent decision-making is critical.

In the future, autonomous machine learning systems leveraging task ontology will play a crucial role in reducing model training time, improving generalization across diverse tasks, and minimizing manual effort in data annotation and feature selection. These systems will facilitate more explainable AI by structuring knowledge hierarchically, making model decisions more interpretable. Additionally, as AI regulations become stricter, task ontology-driven models will aid in ensuring compliance and ethical AI deployment.

With advancements in federated learning, cloud computing, and edge AI, autonomous machine learning models will seamlessly integrate across distributed environments, enabling real-time analytics and intelligent automation. In research and industry, these models will empower AI-driven innovations such as self-optimizing business workflows, personalized AI assistants, and autonomous scientific discovery. Ultimately, this technology will drive the next wave of AI-powered automation, making machines more efficient, adaptive, and capable of solving complex real-world challenges with minimal human intervention.

REFERENCES

1. D. P. Pop and A. Altar. (2014). Designing an MVC model for rapid web application development. Procedia Engineering, 69, 1172-1179.
2. M. Aniche, G. Bavota, C. Treude, M. A. Gerosa, and A. van Deursen, “Code smells for Model-View-Controller architectures,” Empir. Softw. Eng., vol. 23, no. 4, pp. 2121–2157, 2018.
3. S. Cass. (2017). The 2017 top programming languages. https://spectrum.ieee.org/computing/software/the-2017-topprogramming-languages. Accessed 18 October 2017.
4. A. Pelme. (2014). Evaluation of a web application architecture. (Dissertation).

Retrieved from http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-107011

5. R. Brown. (2015). Django vs. flask vs. pyramid: Choosing a python web framework. Recuperado el, 31.
6. P. Vogel, T. Klooster, V. Andrikopoulos, and M. Lungu, "A LowEffort Analytics Platform for Visualizing Evolving Flask-Based Python Web Services," in 2017 IEEE Working Conference on Software Visualization (VISSOFT), 2017, pp. 109–113.
7. Anonym. Tornado Python framework. http://www.tornadoweb.org/en/stable/
8. Hellkamp, M. (2017). Bottle: Python Web Framework. Available: https://bottlepy.org/docs/dev/. Accessed 28 October 2017.
9. S. Dasgupta and S. Hooshangi, "Code Quality: Examining the Efficacy of Automated Tools," AMCIS 2017 Proc., Aug. 2017.
10. G. Farah, J. S. Tejada, and D. Correal, "OpenHub: A Scalable Architecture for the Analysis of Software Quality Attributes," in Proceedings of the 11th Working Conference on Mining Software Repositories, 2014, pp. 420–423.
11. Y. Saleh. (2017). How does the django framework in python overcome the frameworks in PHP? https://www.egrovesys.com/blog/how-does-the-django-framework-inpython-overcome-the-frameworks-in-php/ Accessed 24 November 2017.
12. M. P. Malhotra and M. K. Shah. Python Based Software for Calculating Cyclomatic Complexity. (2015). International Journal of Innovative Science, Engineering and Technology, 2(3):546–549, March 2015.
13. M. del Pilar Salas-Zárate, G. Alor-Hernández, R. Valencia-García, L. Rodríguez-Mazahua, A. Rodríguez-González, and J. L. L. Cuadrado (2015). Analyzing best practices on Web development frameworks: The lift approach. Science of Computer Programming, 102, 1-19.
14. N. Gift. (2017). Writing clean, testable, high-quality code in python. https://www.ibm.com/developerworks/aix/library/au-cleancode/ Accessed 19 December 2017.
15. C. Thirumalai, R. R. Shridharshan, and L. R. Reynold, "An assessment of halstead and COCOMO model for effort estimation," in 2017 Innovations in Power and Advanced Computing Technologies (i-PACT), 2017, pp. 1–4.
16. Anonymous. (2017). Radon. https://pypi.python.org/pypi/radon Accessed 30 November 2017.
17. A. Calleja, J. Tapiador and Caballero, J. (2016, September). A look into 30 years of malware development from a software metrics perspective. In International Symposium on Research in Attacks, Intrusions, and Defenses, pp. 325-345. Springer, Cham.
18. T. J. McCabe. (1976). A complexity measure. IEEE Transactions on Software Engineering, SE-2(4):308–320, Dec 1976
19. R. Smith, T. Tang, J. Warren and S. Rixner. (2017, June). An Automated System for Interactively Learning Software Testing. In Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education, pp. 98-103.
20. Anonymous. Pylint. https://www.pylint.org Accessed 12 December 2017. T. J. McCabe. A complexity measure. IEEE Transactions

on Software Engineering, SE-2(4):308–320, Dec 1976.