ITEA Office
High Tech Campus 69 - 3
5656 AG Eindhoven
The Netherlands

T + 31 88 003 6136
E info@itea3.org
W www.itea3.org

ITEA 3 is a EUREKA strategic ICT cluster programme

Exploitable Results by Third Parties

17030 DayTiMe

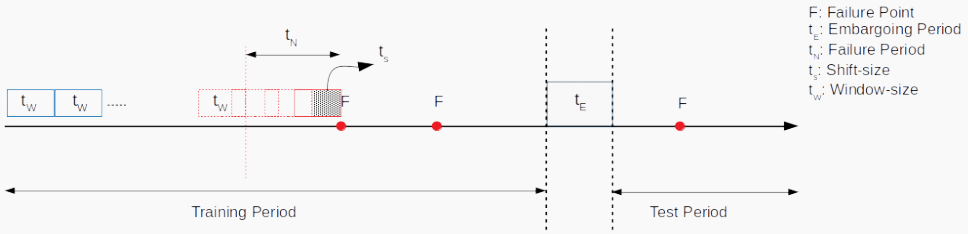
Project details

Project leader:	Ad de Beer
Email:	Ad.de.beer@philips.com
Website:	https://daytime-project.weebly.com/ http://daytimetr.com/

Name: Automatic Process Management for Predictive Maintenance Task		
Input(s):	Main feature(s)	Output(s):
▪ Predicted failure data from ML platforms	Automatically manage the failure task based on the pre-defined workflows and then distribute the task to the proper worker (e.g. technician).	▪ A notification will send by email or SMS
Unique Selling Proposition(s):	▪ Automatically managing the maintenance and failure process of a system is our key business value that reduces the time needed for process management.	
Integration constraint(s):	▪ Format of the failure data ▪ Response latency to request number	
Intended user(s):	▪ Users that need managing, monitoring, and controlling whole processes needed in their systems such as health, law, and insurance domains.	
Provider:	▪ VAS Telekom	
Contact point:	▪ sarper.bagci@hermesiletisim.net	
Condition(s) for reuse:	▪ Commercial licensing	

Name: Log Manager Automation		
Input(s):	Main feature(s)	Output(s):
▪ C# Source Code	▪ Creating log file as sooner as transaction is happened then share to end point of central server.	▪ .txt file extension
Unique Selling Proposition(s):	▪ Automatic raw data and log file generator from working point to central point.	
Integration constraint(s):	▪ TCPIP ▪ .net core utility ▪ .dll services ▪ C# application extension	
Intended user(s):	▪ Technicians who maintain field ▪ ML platform user for AI algorithm input service supplier	
Provider:	▪ Triatech Tıbbi Sistemler San.Tic.A.S.	
Contact point:	▪ mustafa.aydin@triatech.com.tr	
Condition(s) for reuse:	▪ Licencing	

Name: Mail Sender Application		
Input(s):	Main feature(s)	Output(s):
▪ C# Source Code ▪ . net core Std. ▪ Gmail service supplier	▪ After getting predictive maintenance data from special web services which is RindApp application, sending automatically an e-mail to relevant personal to maintain action on field.	▪ Mail information format ▪ E-mail creation
Unique Selling Proposition(s):	▪ Triggered .net web services by the other web services and catching the data then clarifying data set and the usefully data send e-mail to relevant technician.	
Integration constraint(s):	▪ .net core utility ▪ .dll services ▪ C# application extension ▪ .net core web service	
Intended user(s):	▪ User of owner of hardware system in far away from central. ▪ User of digital twin end point	
Provider:	▪ Triatech Tıbbi Sistemler San.Tic.A.S.	
Contact point:	▪ mustafa.aydin@triatech.com.tr	
Condition(s) for reuse:	▪ Licencing	

Name: Failure Prediction Algorithm with Descriptive Log File Analysis		
Input(s):	Main feature(s)	Output(s):
▪ Time series log data ▪ Previous failure data	▪ Our machine learning algorithm is designed as a binary classification model and the model tries to predict lock failures in advance by using a sliding windows feature extraction approach.	▪ To predict whether a failure will occur within N periods (e.g. days, weeks, hours etc.) by analysis of logs of StockArt devices that established in stations provided by TriaTech.
Unique Selling Proposition(s):	▪ The key business value will be is to predict lock failures at the early stages and enable cost saving. If use case owner will know in advance which lock in which station will likely to fail in advance, they can arrange priorities accordingly. Additionally they can optimize maintenance calendar according to their resources. ▪ What makes our classification approach different is that, feature engineering is performed on the log data by using sliding window. The effects of window size and shift size between windows are also tested.  ***Sliding Window Model Creation***	
Integration constraint(s):	▪ Time series log data should be provided ▪ Since this is a supervised machine learning algorithm, previous failure data (medical cabinet lock failures in this example) should be provided	
Intended user(s):	▪ End users that are using any industrial product (medical cabinets in this example)	
Provider:	▪ Havelsan A.Ş	
Contact point:	▪ itarim@havelsan.com.tr	
Condition(s) for reuse:	▪ Commercial licence to be negotiated	

Name: Personalized shaver		
Input(s):	Main feature(s)	Output(s):
▪ Pressure sensors ▪ User test ▪ Knowledge ▪ Digital twin (model)	▪ Based on preferences and measured pressure, an adjusted shaver (advice) will be given to the consumer.	▪ Adjusted pressure settings ▪ Advices via the app
Unique Selling Proposition(s):	▪ Shaving preferences, beard density, hair length and skin properties differ between users. The average settings are not always the best. A personalized shaver will improve satisfaction.	
Integration constraint(s):	▪ Shaving app on mobile phone ▪ Clouds for data exchange	
Intended user(s):	▪ Mail (beard) consumers over the whole world	
Provider:	▪ Philips Consumer Lifestyle	
Contact point:	▪ jan.van.der.weit@philips.com	
Condition(s) for reuse:	▪ N.a.	

Name: Anomalous Logline Detection Tool		
Input(s):	Main feature(s)	Output(s):
▪ log text files ▪ mixed content allowed ▪ no size restriction ▪ application knowledge	▪ Identifying single anomalous log lines ▪ AI/ML application, not rule based ▪ Taking context into account	▪ Anomalous line pointer ▪ In learning phase: feedback-form to teach what's normal

Unique Selling Proposition(s):	▪ Machine problems can be detected in an early phase and solved, preventing failure or misfunction and thus standstill times ▪ Very short training phase, also after software version change
Integration constraint(s):	▪ Service or development engineer, or annotated log files, needed for application-specific machine learning training ▪ Online access to logging computer, or local offline operation ▪ Processing power dependent on log file size and quantity
Intended user(s):	▪ Machine and Instrument application developers ▪ Any service supplier who could exploit routinely logged information files
Provider:	▪ Thunderbyte.AI B.V., Groningen, The Netherlands
Contact point:	▪ rolf.neubert@thunderbyte.ai
Condition(s) for reuse:	▪ Commercial licence to be negotiated

Name: AR Maintenance Tool		
Input(s):	Main feature(s)	Output(s):
▪ PS-Tech AR module based on Unreal Engine 4 technology, 3D CAD models of the device requiring maintenance, structured manual of the maintenance procedure	▪ The 3D models are combined with detailed maintenance guide steps into an AR maintenance tool based on Unreal Engine 4 technology. The 3D models together with the manual enable the creation of an AR app where specific parts of the machinery required during each maintenance step can be highlighted in an AR overlay on top of the actual machine. Next to each highlighted component, additional information like the component name or information about the current component status can be shown. Animations can be added to help a maintenance engineer understand how to handle or work with the highlighted component.	▪ Interactive mobile AR tool for performing guided maintenance on specific machinery

Unique Selling Proposition(s):	▪ AR-guided maintenance using low-cost hardware available to almost any engineer. ▪ Structured step-by-step guidance mapped directly to the machine at hand reduces probability of maintenance errors and the extended (costly) downtime that would result from such errors. ▪ Maintenance engineers are enabled to perform maintenance faster and with less errors, making maintenance more cost-effective
Integration constraint(s):	▪ 3D CAD models of the machine should be available ▪ Discrepancies between CAD models and real-world device must be limited ▪ Structured maintenance guide should be available for the procedure ▪ Engineers require AR-capable mobile device
Intended user(s):	▪ Any industry with complex high-tech machinery in the field that requires regular maintenance by engineers with varying experience levels
Provider:	▪ PS-Tech B.V.
Contact point:	▪ Arjen.brinkman@ps-tech.com
Condition(s) for reuse:	▪ Licensing

Name: Optimized Planned Maintenance Tool for Age-based Maintenance Under Population Heterogeneity		
Input(s):	Main feature(s)	Output(s):
▪ R code	▪ Scheduling the next planned maintenance activity under given data regarding the remaining lifespan of the system and time-to-failure model parameters	▪ .Rdata file ▪ Plots of optimal cost and policy
Unique Selling Proposition(s):	▪ First study that balances exploration and exploitation optimally for a system with fixed lifespan under population heterogeneity	
Integration constraint(s):	▪ .Integration with data base with R ▪ R shiny integration for user interface	
Intended user(s):	▪ Academic researches ▪ Capital goods industry under time-to-failure model uncertainty	
Provider:	▪ Eindhoven University of Technology	
Contact point:	▪ i.dursun@tue.nl	
Condition(s) for reuse:	▪ MIT License	

Name: Multi-Agent Reinforcement Tool		
Input(s):	Main feature(s)	Output(s):
▪ text	▪ Text classification with explainable AI and built-in intelligent feedback loop	▪ Classification
Unique Selling Proposition(s):	▪ Combination of explainable AI and built-in feedback loop. ▪ The multi-agent approach leverages the power of knowledge-sharing between agents/users without necessarily enforcing a single truth	
Integration constraint(s):	▪ Text data has to be provided	
Intended user(s):	▪ Business ▪ Academic research	
Provider:	▪ Datenna B.V.	
Contact point:	▪ info@datenna.com	
Condition(s) for reuse:	▪ to be negotiated	

Name: Malfunction area classification to support the remote diagnostics of medical device		
Input(s):	Main feature(s)	Output(s):
▪ Service work order (SWO) data	▪ The classification algorithm and user interface are developed to automatically determine the suspected malfunction area given (limited) service case details such as subject, problem reported by customer, consumed replaced parts, or case activity notes written by engineers.	▪ Identified malfunction area for the given SWO
Unique Selling Proposition(s):	▪ The classification algorithm is able to identify the correct malfunction area given a free text input with an average of 85% accuracy. It can also handle multi languages	
Integration constraint(s):	▪ Require service work order (description of the problem) as text input ▪ Some specific Python libraries required	
Intended user(s):	▪ Remote service engineer of medical devices	
Provider:	▪ Philips Research	
Contact point:	▪ q.gao@philips.com	
Condition(s) for reuse:	▪ Licensing	

Name: Yanomaly		
Input(s):	Main feature(s)	Output(s):
▪ Various data sources – flat files, InfluxDB, REST API, OSISoft PI, MQTT	▪ Analytics (in particular anomaly detection) on machine-generated timeseries data ▪ Various visualisations for rapid analysis by non-data scientists ▪ Various browser-based UI screens for training, management and deployment of data sources and models ▪ Scalable architecture for robust deployment of hundreds of AI models	▪ Predictions, anomaly scores, root causes and alarm notifications to various data sinks
Unique Selling Proposition(s):	▪ Robust and scalable AI-based anomaly detection and predictive analytics ▪ Specific algorithms for detecting various issues with different equipments: sensor drift, sensor errors, (impending) failures of electric drives and motor, pumps, valves, degradation of control loop performance, … ▪ Context-dependence of algorithms – algorithms take into account machine operating conditions (product types, startup vs regime) to enhance accuracy	
Integration constraint(s):	Yanomaly is deployed on Kubernetes, a scalable container orchestration platform. It can be deployed on cloud native Kubernetes service providers (such as Google Kubernetes Engine, Amazon Elastic Kubernetes Service or Azure Kubernetes Service) as well as on single node installations (such as k3s). Although single node deployments of Yanomaly sacrifice redundancy and scalability, it is still a valid mode for proof-of-concept, demo or smaller production setups. The installation procedure is tested for deployment to a server running Ubuntu Focal (20.04) and CentOS. Other versions from Ubuntu, starting from Bionic will probably also work. YFM will run on another Linux OS as well, as long as Kubernetes environment is installed. Other installation requirements: ▪ SSH access using public key authentication ▪ Passwordless sudo on the target host (with !requiretty setting) Internet access from the target host. The installation procedure supports installations from behind an HTTP proxy. A list of endpoints to be whitelisted can be provided on request.	
Intended user(s):	▪ Consumption of predicitons / notifications: Process engineers, maintenance engineers, operators, production managers ▪ Building, maintenance of models: data scientists, process engineers	
Provider:	▪ Yazzoom	
Contact point:	▪ David.verstraeten@yazzoom.com	

Condition(s) for reuse:	▪ Software license based on functionality and number of monitored tags

<table>
<tr><th colspan="4">Name: TAZI AutoML Platform</th></tr>
<tr><td>Input(s):</td><td colspan="2">Main feature(s)</td><td>Output(s):</td></tr>
<tr><td>▪ Tabular Data
▪ (DB Table, .csv, .parquet, .json, GCS)</td><td colspan="2">▪ Create Machine Learning models, get business understandable, explainable real time ML outputs.</td><td>▪ ML predictions (.csv, DB table, parquet, json, GCS…)</td></tr>
<tr><td>Unique Selling Proposition(s):</td><td colspan="3">▪ Continuously learning AI algorithms. No code, easy to use AutoML platform. Models learning from human feedback too. Understandable explainable, actionable ML.</td></tr>
<tr><td>Integration constraint(s):</td><td colspan="3">
<table>
<tr><th>▪ Category</th><th>▪ Specification</th><th>▪ Details</th></tr>
<tr><td>▪ OS</td><td>▪ Linux</td><td>▪ CentOS or Red Hat versions 7 or 8</td></tr>
<tr><td>▪ Java</td><td>▪ 1.8</td><td>▪ OpenJDK</td></tr>
<tr><td>▪ Admin access port</td><td>▪ 22</td><td>▪ for ssh access to the server</td></tr>
<tr><td>▪ UI access port</td><td>▪ 80 or 443</td><td>▪ UI & REST API served over http(s)</td></tr>
<tr><td>▪ 3rd party software to be installed on the server</td><td>▪ utilities; access to yum repos needed</td><td>▪ docker, docker-compose, perl, python (3.x), nginx, net-tools package, jq, git, vim, htop, bc</td></tr>
<tr><td rowspan="2">▪ TAZI software packages</td><td>▪ https://git.tazi.ai</td><td>▪ Tazi platform distribution repository; access needed for initial install & updates</td></tr>
<tr><td>▪ https://registry.tazi.ai
▪ https://hub.docker.com</td><td>▪ Private registry for TAZI-built docker images & public registry for 3rd party docker images; access needed for initial install & updates</td></tr>
</table>
</td></tr>
<tr><td>Intended user(s):</td><td colspan="3">▪ Business Experts
▪ Data Scientists</td></tr>
</table>

Provider:	▪ TAZI AI Systems.
Contact point:	▪ nazire@tazi.ai
Condition(s) for reuse:	▪ Commercial Licensing